Block Matrix Computations and the Singular Value Decomposition

A Tale of Two Ideas

Charles F. Van Loan

Department of Computer Science

Cornell University

Supported in part by the NSF contract CCR-9901988.

Block Matrices

A *block matrix* is a matrix with matrix entries, e.g.,

$$A = \left[\begin{array}{cc|cc} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \hline \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{array}\right] = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \qquad A_{ij} \in \mathbb{R}^{3\times 2}$$

Operations are pretty much "business as usual", e.g.

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]^T \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{c|c} A_{11}^T & A_{21}^T \\ \hline A_{12}^T & A_{22}^T \end{array}\right] \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{c|c} A_{11}^T A_{11} + A_{21}^T A_{21} & etc \\ \hline etc & etc \end{array}\right]$$

E.g., Strassen Multiplication

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$\begin{aligned}
P_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) \\
P_2 &= (A_{21} + A_{22})B_{11} \\
P_3 &= A_{11}(B_{12} - B_{22}) \\
P_4 &= A_{22}(B_{21} - B_{11}) \\
P_5 &= (A_{11} + A_{12})B_{22} \\
P_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\
P_7 &= (A_{12} - A_{22})(B_{21} + B_{22}) \\
C_{11} &= P_1 + P_4 - P_5 + P_7 \\
C_{12} &= P_3 + P_5 \\
C_{21} &= P_2 + P_4 \\
C_{22} &= P_1 + P_3 - P_2 + P_6
\end{aligned}$$

Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{m \times n}$ then there exist orthogonal $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ so

$$U^T A V \;=\; \Sigma \;=\; \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \;=\; \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values.

Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{m\times n}$ then there exist orthogonal $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ so

$$U^T AV = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values.

Fact 1. The columns of $V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$ and $U = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}$ are the right and left singular vectors and they are related:

$$\begin{aligned} Av_j &= \sigma_j u_j \\ A^T u_j &= \sigma_j v_j \end{aligned} \qquad j = 1{:}n$$

Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{m\times n}$ then there exist orthogonal $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ so

$$U^T A V \;=\; \Sigma \;=\; \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \;=\; \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values.

Fact 2. The SVD of A is related to the eigen-decompositions of $A^T A$ and AA^T:

$$V^T(A^T A)V \;=\; \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$$

$$U^T(AA^T)U \;=\; \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2, \underbrace{0, \ldots, 0}_{m-n})$$

Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{m \times n}$ then there exist orthogonal $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ so

$$U^T A V \;=\; \Sigma \;=\; \text{diag}(\sigma_1, \ldots, \sigma_n) \;=\; \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values.

Fact 3. The smallest singular value is the distance from A to the set of rank deficient matrices:

$$\sigma_{min} \;=\; \min_{\text{rank}(B) < n} \; \| \, A - B \, \|_F$$

Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{m \times n}$ then there exist orthogonal $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ so

$$U^T A V \;=\; \Sigma \;=\; \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \;=\; \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the singular values.

Fact 4. The matrix $\sigma_1 u_1 v_1^T$ is the closest rank-1 matrix to A, i.e., it solves the problem:

$$\sigma_{min} \;=\; \min_{\mathrm{rank}(B) = 1} \; \| \, A - B \, \|_F$$

The High-Level Message..

- It is important to be able to think at the block level because of problem structure.
- It is important to be able to develop block matrix algorithms
- There is a progression...

“Simple” Linear Algebra

$\downarrow$

Block Linear Algebra

$\downarrow$

Multilinear Algebra

Reasoning at the Block Level

Uncontrollability

The system

$$\dot{x} \;=\; Ax \;+\; Bu \qquad A \in \mathbb{R}^{n\times n},\; B \in \mathbb{R}^{n\times p},\; n > p$$

is uncontrollable if

$$G = \int_0^t e^{A(t-\tau)} B B^T e^{A\tau} d\tau$$

is singular.

Uncontrollability

The system

$$\dot{x} \;=\; Ax \;+\; Bu \qquad A \in \mathrm{I\!R}^{n\times n},\; B \in \mathrm{I\!R}^{n\times p},\; n > p$$

is uncontrollable if

$$G = \int_0^t e^{A(t-\tau)} B B^T e^{A\tau} d\tau$$

is singular.

$$\tilde{A} \;=\; \begin{bmatrix} A & BB^T \\ 0 & A^T \end{bmatrix} \qquad \longrightarrow \qquad e^{\tilde{A}t} \;=\; \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} \qquad \longrightarrow \qquad G = F_{11}^T F_{12}$$

Nearness to Uncontrollability

The system

$$\dot{x} \;=\; Ax \;+\; Bu \qquad A \in \mathbb{R}^{n \times n},\; B \in \mathbb{R}^{n \times p},\; n > p$$

is *nearly* uncontrollable if the minimum singular value of

$$G = \int_0^t e^{A(t-\tau)} BB^T e^{A\tau} d\tau$$

is *small.*

$$\tilde{A} \;=\; \begin{bmatrix} A & BB^T \\ 0 & A \end{bmatrix} \qquad \longrightarrow \qquad e^{\tilde{A}t} \;=\; \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} \qquad \longrightarrow \qquad G = F_{11}^T F_{12}$$

Developing Algorithms at the Block Level

Block Matrix Factorizations: A Challenge

By a *block* algorithm we mean an algorithm that is *rich* in matrix-matrix multiplication.

Is there a block algorithm for Gaussian elimination? I.e., is there a way to rearrange the $O(n^3)$ operations so that the implementation spends at most $O(n^2)$ time *not* doing matrix-matrix multiplication?

Why?

Re-use ideology: when you touch data, you want to use it a lot.

Not all linear algebra operations are equal in this regard.

Level	Example	Data	Work
1	$\alpha = y^T z$	$O(n)$	$O(n)$
	$y = y + \alpha z$	$O(n)$	$O(n)$
2	$y = y + As$	$O(n^2)$	$O(n^2)$
	$A = A + yz^T$	$O(n^2)$	$O(n^2)$
3	$A = A + BC$	$O(n^2)$	$O(n^3)$

Here, α is a scalar, y, z vectors, and A, B, C matrices

Scalar LU

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$\left.\begin{aligned} a_{11} &= u_{11} \\ a_{12} &= u_{12} \\ a_{13} &= u_{13} \\ a_{21} &= \ell_{21}u_{11} \\ a_{31} &= \ell_{31}u_{11} \\ a_{22} &= \ell_{21}u_{12} + u_{22} \\ a_{23} &= \ell_{21}u_{13} + u_{23} \\ a_{32} &= \ell_{31}u_{12} + \ell_{32}u_{22} \\ a_{33} &= \ell_{31}u_{13} + \ell_{32}u_{23} + u_{33} \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} u_{11} &= a_{11} \\ u_{12} &= a_{12} \\ u_{13} &= a_{13} \\ \ell_{21} &= a_{21}/u_{11} \\ \ell_{31} &= a_{31}/u_{11} \\ u_{22} &= a_{22} - \ell_{21}u_{12} \\ u_{23} &= a_{23} - \ell_{21}u_{13} \\ \ell_{32} &= (a_{32} - \ell_{31}u_{12})/u_{22} \\ u_{33} &= a_{33} - \ell_{31}u_{13} - \ell_{32}u_{23} \end{aligned}\right.$$

Recursive Description

If $\alpha \in \mathbb{R}$, $v, w \in \mathbb{R}^{n-1}$, and $B \in \mathbb{R}^{(n-1)\times(n-1)}$ then

$$A = \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ v/\alpha & \tilde{L} \end{bmatrix} \begin{bmatrix} \alpha & w^T \\ 0 & \tilde{U} \end{bmatrix}$$

is the LU factorization of A if

$$\tilde{L}\tilde{U} = \tilde{A}$$

is the LU factorization of

$$\tilde{A} = B - vw^T/\alpha.$$

Rich in level-2 operations.

Block LU: Recursive Description

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$

$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad p \qquad n-p$$

$A_{11} = L_{11}U_{11}$	Get L_{11}, U_{11} via LU of A_{11}
$A_{21} = L_{21}U_{11}$	Solve triangular systems for L_{21}
$A_{12} = L_{11}U_{12}$	Solve triangular systems for U_{12}
$A_{22} = L_{21}U_{12} + L_{22}U_{22}$	Form $\tilde{A} = A_{22} - L_{21}U_{12}$
	$\downarrow \quad \downarrow \quad \downarrow$
	Get L_{22}, U_{22} via LU of $\tilde{A}$

Block LU: Recursive Description

$$
\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}
\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}
=
\begin{array}{c} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \\ \quad p \qquad n-p \end{array}
\begin{array}{c} p \\ n-p \\ \ \end{array}
$$

$A_{11} = L_{11}U_{11}$	Get L_{11}, U_{11} via LU of A_{11}	$O(p^3)$
$A_{21} = L_{21}U_{11}$	Solve triangular systems for L_{21}	$O(np^2)$
$A_{12} = L_{11}U_{12}$	Solve triangular systems for U_{12}	$O(np^2)$
$A_{22} = L_{21}U_{12} + L_{22}U_{22}$	Form $\tilde{A} = A_{22} - L_{21}U_{12}$	$O(n^2p)$
	$\downarrow \quad \downarrow \quad \downarrow$	
Recur $\rightarrow$ $\quad$ $\rightarrow$	Get L_{22}, U_{22} via LU of $\tilde{A}$	

Rich in level-3 operations!!!

Consider: $\tilde{A} = A_{22} - L_{21}U_{12}$

$p = 3$:

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{bmatrix} - \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{bmatrix}$$

p must be "big enough" so that the advantage of data re-use is realized.

Block Algorithms for (Some) Matrix Factorizations

- LU with pivoting:

$$PA = LU \qquad P \text{ permutation, } L \text{ lower triangular, } U \text{ upper triangular}$$

- Cholesky factorization for symmetric positive definite A:

$$A = GG^T \qquad G \text{ lower triangular}$$

- The QR factorization for rectangular matrices:

$$A = QR \qquad Q \in \mathbb{R}^{m\times m} \text{ orthogonal } R \in \mathbb{R}^{m\times n} \text{ upper triangular}$$

Block Matrix Factorization Algorithms via Aggregation

Developing a Block QR Factorization

The standard algorithm computes Q as a product of Householder reflections,

$$Q^T A \;=\; H_n \cdots H_1 A \;=\; R$$

After 2 steps...

$$H_2 H_1 A \;=\; \left[\begin{array}{ccccc} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{array}\right].$$

The H matrices look like this:

$$H = I - 2vv^T \qquad\qquad v \text{ a unit 2-norm column vector}$$

Developing a Block QR Factorization

The standard algorithm computes Q as a product of Householder reflections,

$$Q^T A \;=\; H_n \cdots H_1 A \;=\; R$$

After 2 steps...

$$H_2 H_1 A \;=\; \left[\begin{array}{ccccc} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \end{array}\right].$$

The H matrices look like this:

$$H = I - 2vv^T \qquad\qquad v \text{ a unit 2-norm column vector}$$

Developing a Block QR Factorization

The standard algorithm computes Q as a product of Householder reflections,

$$Q^T A \;=\; H_n \cdots H_1 A \;=\; R$$

After 3 steps...

$$H_3 H_2 H_1 A \;=\; \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

The H matrices look like this:

$$H = I - 2vv^T \qquad\qquad v \text{ a unit 2-norm column vector}$$

Aggregation

$$A = \left[\begin{array}{c|c} A_1 & A_2 \end{array} \right] \qquad p << n$$

$$\quad p \quad n-p$$

- Generate the first p Householders based on A_1:

$$H_p \cdots H_1 A_1 = R_{11} \qquad \text{(upper triangular)}$$

- Aggregate $H_1, \ldots, H_p$:

$$H_p \cdots H_1 = I - 2WY^T \qquad W, Y \in \mathbb{R}^{m \times p}$$

- Apply to rest of matrix:

$$(H_p \cdots H_1)A = \left[\begin{array}{c|c} (H_p \cdots H_1)A_1 & (I - 2WY^T)A_2 \end{array} \right]$$

The WY Representation

- Aggregation:

$$(I - 2WY^T)(I - 2vv^T) \;=\; I - 2W_+Y_+^T$$

where

$$\begin{aligned} W_+ &= \left[\, W \;\middle|\; (I - 2WY^T)v \,\right] \\ Y_+ &= \left[\, Y \;\middle|\; v \,\right] \end{aligned}$$

- Application

$$A \;\leftarrow (I - 2WY^T)A = A - (2W)(Y^TA)$$

The Curse of Similarity Transforms

A Block Householder Tridiagonalization?

A symmetric. Compute orthogonal Q such that

$$Q^T A Q = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

The standard algorithm computes Q as a product of Householder reflections,

$$Q = H_1 \cdots H_{n-2}$$

A Block Householder Tridiagonalization?

H_1 introduces zeros in first column

$$H_1^T A = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix}.$$

Scrambles rows 2 through 6.

A Block Householder Tridiagonalization?

Must also post-multiply...

$$H_1^T A H_1 \;=\; \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix}.$$

H_1 scrambles columns 2 through 6.

A Block Householder Tridiagonalization?

H_2 introduces zeros into second column

$$H_1^T A H_1 = \begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times \\ 0 & \boxtimes & \times & \times & \times & \times \\ 0 & \boxtimes & \times & \times & \times & \times \\ 0 & \boxtimes & \times & \times & \times & \times \\ 0 & \boxtimes & \times & \times & \times & \times \end{bmatrix}.$$

Note that because of H_1's impact, H_2 depends on all of A's entries.

The H_i can be aggregated, but A must be completely updated along the way destroys the advantage.

Jacobi Methods for Symmetric Eigenproblem

These methods do not initially reduce A to "condensed" form.

They repeatedly reduce the sum-of-squares of the off-diagonal elements.

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & 0 \\ \times & \times & \times & \times \\ \times & 0 & \times & \times \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{bmatrix}^T A \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{bmatrix}$$

The (2,4) subproblem:

$$\begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{22} & a_{24} \\ a_{42} & a_{44} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

$c^2 + s^2 = 1$ and the off-diagonal sum of squares is reduced by $2a_{24}^2$.

Jacobi Methods for Symmetric Eigenproblem

Cycle through subproblems:

$$(1,2)\ ,\ (1,3)\ ,\ (1,4)\ ,\ (2,3)\ ,\ (2,4)\ ,\ (3,4)$$

(1,2):

$$A = \begin{bmatrix} \times & 0 & \times & \times \\ 0 & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \leftarrow \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T A \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(1,3):

$$A = \begin{bmatrix} \times & \times & 0 & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \leftarrow \begin{bmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T A \begin{bmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

etc.

Parallel Jacobi

Parallel Ordering:

$$\{(1,2)\ ,\ (3,4)\}\ ,\ \{(1,4)\ ,\ (2,3)\}\ ,\{\ (1,3)\ ,\ (2,4)\}$$

(1,2):

$$A\ =\begin{bmatrix} \times & 0 & \times & \times \\ 0 & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \leftarrow \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T A \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3,4):

$$A\ =\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & 0 \\ \times & \times & 0 & \times \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}^T A \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}$$

Block Jacobi

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & 0 \\ \times & \times & \times & \times \\ \times & 0 & \times & \times \end{bmatrix} \leftarrow \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & Q_{11} & 0 & Q_{12} \\ 0 & 0 & I & 0 \\ 0 & Q_{21} & 0 & Q_{22} \end{bmatrix}^T A \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & Q_{11} & 0 & Q_{12} \\ 0 & 0 & I & 0 \\ 0 & Q_{21} & 0 & Q_{22} \end{bmatrix}$$

The (2,4) subproblem:

$$\begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^T \begin{bmatrix} A_{22} & A_{24} \\ A_{42} & A_{44} \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

Convergence analysis requires an understanding of 2-by-2 block matrices that are orthogonal...

The CS Decomposition

The blocks of an orthogonal matrix have related SVDs:

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} C & S \\ -S & C \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}^T$$

$$\begin{aligned} C &= \operatorname{diag}(\cos(\theta_1), \ldots, \cos(\theta_m)) \\ S &= \operatorname{diag}(\sin(\theta_1), \ldots, \sin(\theta_m)) \end{aligned}$$

$$\begin{aligned} U_1^T Q_{11} V_1 &= C \\ U_1^T Q_{12} V_2 &= S \\ U_2^T Q_{21} V_1 &= -S \\ U_2^T Q_{22} V_2 &= C \end{aligned}$$

The Curse of Sparsity

How to Extract Level-3 Performance?

Suppose we want to solve a large sparse $Ax = b$ problem.

Iterative methods for this problem can often be dramatically accelerated if you can find a matrix M with two properties:

- It is easy/efficient to solve linear systems of the form $Mz = r$.
- M approximates the "essence" of A.

M is called a *preconditioner* and the original iteration is modified to effectively solve the equivalent linear system $(M^{-1}A)x = (M^{-1}b)$.

Idea: Kronecker Product Preconditioners

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \left[\begin{array}{ccc|ccc} b_{11}c_{11} & b_{11}c_{12} & b_{11}c_{13} & b_{12}c_{11} & b_{12}c_{12} & b_{12}c_{13} \\ b_{11}c_{21} & b_{11}c_{22} & b_{11}c_{23} & b_{12}c_{21} & b_{12}c_{22} & b_{12}c_{23} \\ b_{11}c_{31} & b_{11}c_{32} & b_{11}c_{33} & b_{12}c_{31} & b_{12}c_{32} & b_{12}c_{33} \\ \hline b_{21}c_{11} & b_{21}c_{12} & b_{21}c_{13} & b_{22}c_{11} & b_{22}c_{12} & b_{22}c_{13} \\ b_{21}c_{21} & b_{21}c_{22} & b_{21}c_{23} & b_{22}c_{21} & b_{22}c_{22} & b_{22}c_{23} \\ b_{21}c_{31} & b_{21}c_{32} & b_{21}c_{33} & b_{22}c_{31} & b_{22}c_{32} & b_{22}c_{33} \end{array}\right]$$

"Replicated Block Structure"

Some Properties and a Big Fact

Properties:

$$
\begin{aligned}
(B \otimes C)^T &= B^T \otimes C^T \\
(B \otimes C)^{-1} &= B^{-1} \otimes C^{-1} \\
(B \otimes C)(D \otimes F) &= BD \otimes CF \\
B \otimes (C \otimes D) &= (B \otimes C) \otimes D
\end{aligned}
$$

Big Fact:

If B and C are m-by-m, then the m^2-by-m^2 linear system $(B \otimes C)z = r$ can be solved in $O(m^3)$ time rather than $O(m^6)$ time.

Capturing Essence with a Kronecker Product

Given

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \qquad A_{ij} \in \mathbb{R}^{m\times m}$$

choose B and C to minimize

$$\| A - B \otimes C \|_F = \left\| \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} - \begin{bmatrix} b_{11}C & b_{12}C & b_{13}C \\ b_{21}C & b_{22}C & b_{23}C \\ b_{31}C & b_{32}C & b_{33}C \end{bmatrix} \right\|_F$$

The exact solution can be obtained via the SVD..

Solution of the min $\| A - B \otimes C \|$ Problem

- Makes the blocks of A into vectors and arrange block-column major order:

$$\tilde{A} = \left[\text{col}(A_{11}) \,|\, \text{col}(A_{21}) \,|\, \text{col}(A_{31}) \,|\, \text{col}(A_{12}) \,|\, \cdots \,|\, \text{col}(A_{33}) \right]$$

- Compute the largest singular value σ_{max} and the corresponding singular vectors u_{max} and v_{max}.

- $B_{opt} = \sqrt{\sigma_{max}} \cdot \text{reshape}(v_{max}, 3, 3))$.

- $C_{opt} = \sqrt{\sigma_{max}} \cdot \text{reshape}(u_{max}, m, m))$.

Conclusions

- It is important to be able to think at the block level because of problem structure.
- It is important to be able to develop block matrix algorithms
- There is a progression...

"Simple" Linear Algebra

$$\downarrow$$

Block Linear Algebra

$$\downarrow$$

Multilinear Algebra (Thursday)