Divide and Conquer

### Divide and Conquer

- **Divide:** partition A and B into \( \frac{n}{2} \times \frac{n}{2} \) blocks.
- **Conquer:** multiply \( \frac{n}{2} \times \frac{n}{2} \) recursively.
- **Combine:** add appropriate products using 4 matrix additions.

### Matrix Multiplication: Key Idea

**Key idea:** multiply 2-by-2 block matrices with only 7 multiplications.

\[
\begin{align*}
C_{11} &= A_{11}B_{11} + A_{12}B_{22} \\
C_{12} &= A_{11}B_{21} + A_{12}B_{22} \\
C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\
C_{22} &= A_{21}B_{12} + A_{22}B_{22}
\end{align*}
\]

- 7 multiplications.
- 16 + 10 + 8 additions (or subtractions).

### Fast Matrix Multiplication

**Fast matrix multiplication.** (Strassen, 1969)

- **Divide:** partition A and B into \( \frac{n}{2} \times \frac{n}{2} \) blocks.
- **Conquer:** multiply 7 \( \frac{n}{2} \times \frac{n}{2} \) matrices recursively.
- **Combine:** 7 products into 4 terms using 8 matrix additions.

**Analysis:**

- Assume \( n \) is a power of 2.
- \( T(n) = 7T(n/2) + O(n^2) \) arithmetic operations.

\[
T(n) = 7T\left(\frac{n}{2}\right) + O(n^2)
\]
Fast Matrix Multiplication in Practice

Implementation issues.
- Sparsity
- Caching effects
- Numerical stability
- Odd matrix dimensions
- Crossover to classical algorithm around \( n = 128 \).

Common misperception: “Strassen is only a theoretical curiosity.”
- Advanced Computation Group at Apple Computer reports 8x speedup on G4 Velocity Engine when \( n \approx 2,500 \).
- Range of instances where it’s useful is a subject of controversy.

Remark: Can “Strassenize” \( Ax = b \), determinant, eigenvalues, and other matrix ops.