18.335 Fall 2008
Performance Experiments
with Matrix Multiplication

Steven G. Johnson

Hardware: 2.66GHz Intel Core 2 Duo
64-bit mode, double precision, gcc 4.1.2
optimized BLAS dgemm: ATLAS 3.6.0
http://math-atlas.sourceforge.net/

The “obvious” C code:

\[
C = \sum_{k=1}^{m} A_k B_k
\]

for \( i = 1 \) to \( m \)
for \( j = 1 \) to \( p \)

\( C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \)

just three loops, how complicated can it get?

A trivial problem?

\[
C = A \quad B
\]

\( m \times p \times n \times p \)

**flops/time** is not constant!
(small matrices, \( m=n=p \))

(2.66GHz processor? why < 1 gigaflops?)

L1 cache exceeded?
L2 cache exceeded?
L1 cache exceeded for single row?

Not all “noise” is random

All flops are not created equal

- We cannot understand performance without understanding memory efficiency (caches).
- Computers are more complicated than you think.
- Even a trivial algorithm is nontrivial to implement well.

Things to remember