

Matrix Multiplication

- 1a. Find out as much information as you can about the computer you will be using for this exercise. Specifics can include speed and type of CPU, speed and type of RAM, and size of cache memory.
- b. Write a program or download one of sample codes from the web page

`http://fractal.math.unr/~ejolson/466/`

that computes $C \leftarrow AB$. Hint: The Matlab code will be the easiest to work with for this exercise.

- c. Let A and B be two $n \times n$ matrices with entries A_{ij} and B_{ij} distributed randomly according to the uniform distribution on the interval $[0, 1]$. Let T_n be the number of seconds it takes to compute $C \leftarrow AB$. Find T_n for your computer and program for values of $n = 127, 128, 256, 259, 511, 512, 763, 768, 1024, 1031, 1280, 1281$. If your computer is particularly fast, you may also want to test $n = 1536, 1537, 1792, 1797, 2048, 2051$.
- d. Looking at the loop structure in the Code Example 2a it is clear that the number of multiplications used to multiply two $n \times n$ matrices together is exactly n^3 . How many additions are used?
- e. If T_n is proportional to the total number of multiplications then

$$T_n \sim Cn^3 = O(n^3).$$

It then follows that the points $(n, T_n^{1/3})$ should lie in a straight line. Plot these points for the values of n given above in part c.

- f. Do the points $(n, T_n^{1/3})$ lie in a straight line? If not, to give some plausible reasons why? Consider how precise the timing clock is, cache speed and size compared to main memory, and cache-line alignment and associativity for matrix sizes $n = 2^k$.
- g. [Extra Credit and for CS/Math 666] Repeat steps b through f using a different programming language. Try different compiler optimization switches to find the best combination. What is the version of the compiler and optimization switches used?