MA6252 Homework 2

Liu Zhaoqiang A0109983

In this homework, I tried direct implementation and Strassen algorithm for matrix-matrix multiplication and compared their computation time.

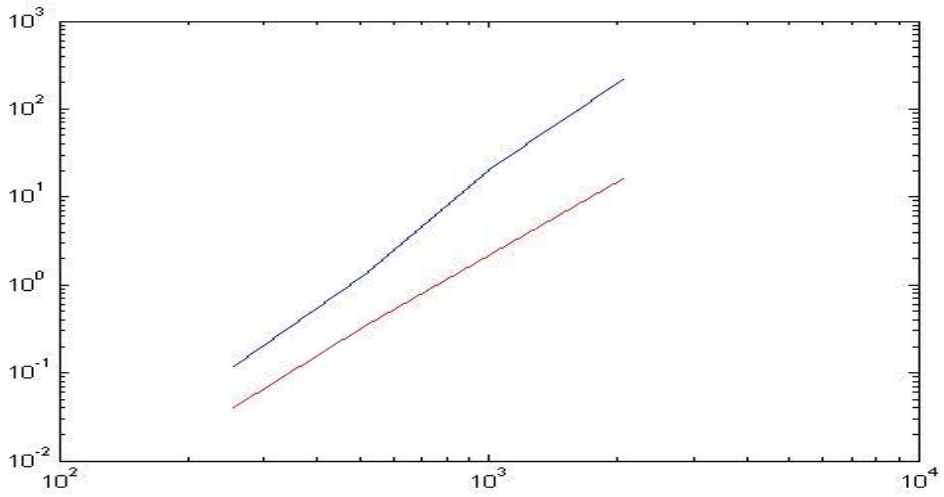
For Strassen algorithm, my code mainly consider 2^n x 2^n matrices' multiplication. If the matrices A, B are not of type 2^n x 2^n, we fill the missing rows and columns with zeros. Thus the computation time is the same with 2^{n+1} x 2^{n+1} matrices' multiplication. For example, if the matrices A, B are of the type 5000*5000, they are treated as 8192*8192 matrices in my code of Strassen algorithm.

My system information is as follows:
Linux Deepin 12.12(a system similar with Ubuntu and made in China), 64bit,
7.6 GiB, Intel Core i7-3630QM CPU @ 2.40GHz × 8.
My compiler is g++.

The computation times are as follows:
Table 1: No optimization options

Dimension of Matrix(n*n)	Computation Time for Direct Implementation(t1/s)	Computation Time for Strassen algorithm(t2/s)
256*256	0.12	0.04
512*512	1.28	0.34
1024*1024	22.12	2.33
2048*2048	218.11	16.42

Loglog graph:



Where the red line represents the Computation Times for Strassen algorithm and the blue

line represents the Computation Time for Direct Implementation.

Using polyfit function in matlab, we have:

log(t1) = 3.6595*log(n) -22.4451; log(t2) = 2.8820*log(n) -19.1413.

(n represents the dimension of matrix, t1 is the computation time for direct implementation, t2 is the computation time for Strassen algorithm)

Table 2: Applying optimization options –O3 or –Ofast

Dimension	Direct, -O3(s)	Strassen, -O3(s)	Direct, -Ofast(s)	Strassen, -Ofast(s)
1000*1000	7.5	0.6	8.27	0.59
1500*1500	31.49	4.54	29.13	4.57
2000*2000	72.67	4.55	75.95	4.84
3000*3000	293.1	32.19	286.81	31.57
4000*4000	811.68	31.78	686.48	31.53
5000*5000	1776.19	221.44	1596.35	221

Table 3: Computation time for matrix-matrix multiplication using Matlab(My Matlab version is: Matlab 7.11.0(R2010b))

Dimension	Time(s)	Dimension	Time(s)
1000*1000	0.2400	3000*3000	4.7700
1024*1024	0.2200	4000*4000	11.4200
1500*1500	0.6200	4096*4096	11.9000
2000*2000	1.5600	5000*5000	22.1200
2048*2048	1.5900	8192*8192	112.5100

According to the above tables and the graph, we can summarize following findings:

(1) The Strassen Algorithm is faster than the direct implementation, and is much faster when the dimension of matrix is large;

(2) According to the linear fitting of the loglog graph, we can see the time complexity of strassen algorithm is about $\log_2 7$=2.8074 (but there should be some bad results in the computation of direct implementation)

(3) Comparing Table 2 and Table 1, we can conclude that we can use optimization options to decrease much computation time;

(4) I downloaded CBlas and GotoBlas2, but because I am not familiar with makefile and compiling options, I could not run the test programs and use the dgemm function in them. However, according to Table 3, we can see that the computation time for matrix-matrix multiplication in Matlab is much less than that of my Strassen algorithm with optimization options when the dimension of matrix is large. Matlab also uses dgemm from BLAS level 3 for matrix-matrix multiplication, and it is even optimized by Intel MKL when applying matrix-matrix multiplication. Thus, we can conclude that using efficient libraries can improve computation efficiency greatly.