Yigao Li & Tao Yu
ANLY 550
Professor: Justin Thaler
April 25, 2018

Programming Assignment 2 Report

Objective:

Strassen's algorithm improves the speed of matrix multiplication. Naive matrix multiplication applies mathematical definition. In the case of multiplying two n by n matrices, conventional algorithm takes $O(n^3)$ time. Strassen's algorithm uses divide and conquer technique, which effectively reduces the time complexity to $O(n^{log_2 7})$. However, in practical, conventional algorithm can be faster when n is small. Therefore, our goal is to find the optimal n to switch between conventional algorithm and Strassen's.

Analytical Approach:

For an n by n matrix multiplication, the conventional algorithm requires n multiplications and $n-1$ additions for every entry in one matrix. Let A and B be two matrices, and $C = AB$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix}$$

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{pmatrix}$$

and each entry $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$. Therefore, the number of operation required to compute matrix multiplication is a function of dimension,

$$f(n) = n^2(n + n - 1) = 2n^3 - n^2 .$$

Strassen's algorithm only requires 7 matrix products per recurrence, 11 additions and 7 subtractions. The work that needs to be done by Strassen's algorithm(occurs when $n > n_0$) can be represented by the recurrence:

Yigao Li & Tao Yu
ANLY 550
Professor: Justin Thaler
April 25, 2018

$$T(n) = 7T(\tfrac{n}{2}) + (11 + 7)\ \tfrac{n^2}{4}$$

However, Strassen's is not an efficient algorithm for matrices of small dimension. We need to find a cross-over point n_0 such that we switch to basic algorithm when Strassen's recursive procedure comes to $n \le n_0$. Thus, our revised Strassen's algorithm needs time,

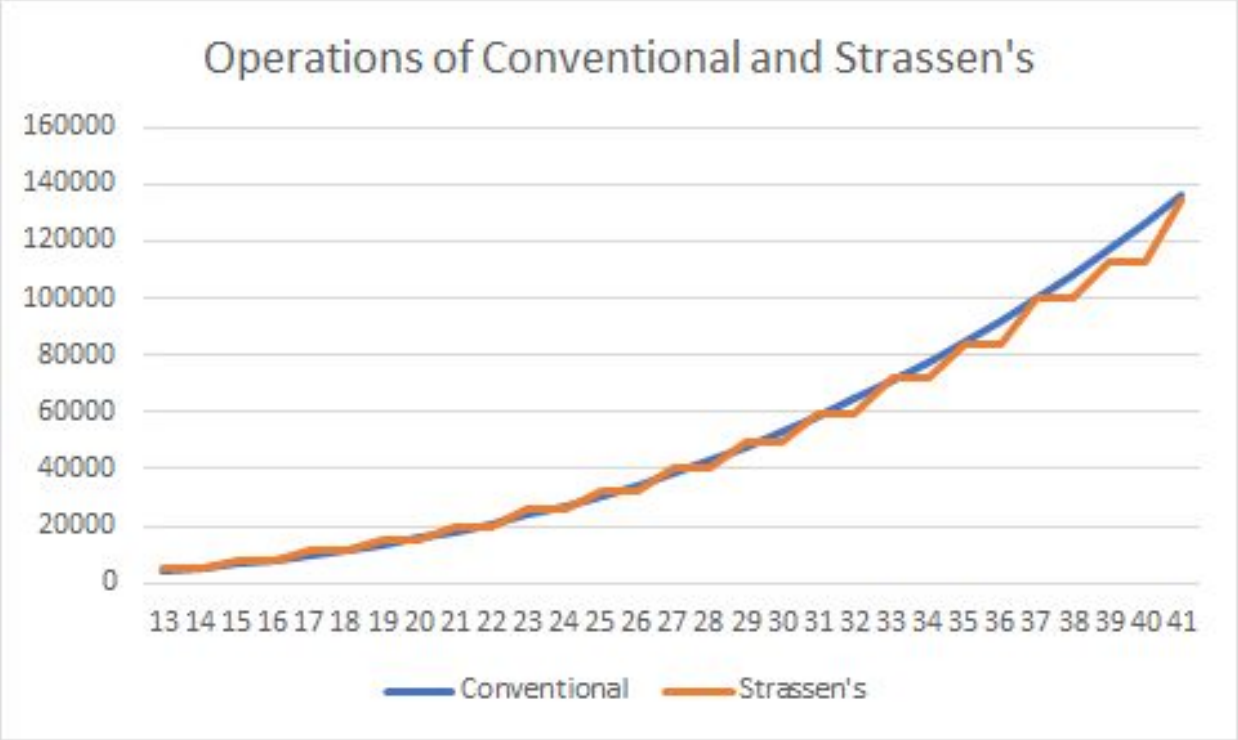
$$T(n) = 7 \bullet min\{T(\tfrac{n}{2}), f(\tfrac{n}{2})\} + 18\tfrac{n^2}{4}$$

In order to figure out the cross-over point from an analytical approach, we make a table of operations used for both algorithms:

Dimension n	$f(n)$	$T(n)$	Choice
1	1	1	Conventional
2	12	25	Conventional
3	45	156	Conventional
4	112	156	Conventional
...	...	...	...
13	4225	5341	Conventional
14	5292	5341	Conventional
15	6525	7872	Conventional
16	7936	7872	Strassen's
17	9537	11097	Conventional
18	11340	11097	Strassen's
...	...	...	...
31	58621	59712	Conventional
32	64512	59712	Strassen's
33	70785	71961	Conventional

Yigao Li & Tao Yu
ANLY 550
Professor: Justin Thaler
April 25, 2018

34	77452	71961	Strassen's
35	84525	83511	Strassen's
36	92016	83511	Strassen's
37	99937	99997	Conventional
38	108300	99997	Strassen's
39	117117	112900	Strassen's
40	126400	112900	Strassen's
41	136161	134505	Strassen's



According to the result above, we set the cross-over point n_0 to be 14 when n is even and 37 when n is odd. In the next section, we will apply this analytical result to improve Strassen's algorithm to experimentally find the cross-over point.

Yigao Li & Tao Yu
ANLY 550
Professor: Justin Thaler
April 25, 2018

Another analytical approach is to set $T(\frac{n}{2})$ as the conventional algorithm's runtime, such as the following:

$$T(n) = 7T(\tfrac{n}{2}) + (11+7)\,\tfrac{n^2}{4} = 7(2(\tfrac{n}{2})^3 - (\tfrac{n}{2})^2) + \tfrac{9}{2}n^2 = \tfrac{7n^3}{2} - \tfrac{11n^2}{2}$$

To find the cross-over point, we set $T(n) = f(n)$ in order to find the intersection.

$$\tfrac{7n^3}{2} - \tfrac{11n^2}{2} = 2n^3 - n^2$$

$$n=15$$

The intersection tells us that the crossover point is 15. So the Strassen becomes faster and more effective when n=15. This approach gives a similar result as our first analytical approach.

Experimental Approach:

Strassen's algorithm breaks matices into 4 submatrices, each with size $\frac{n}{2}$ by $\frac{n}{2}$:

$$A = \begin{pmatrix} X & Y \\ Z & W \end{pmatrix} \quad B = \begin{pmatrix} E & F \\ G & H \end{pmatrix} \quad C = \begin{pmatrix} XE+YG & XF+YH \\ ZE+WG & ZF+WH \end{pmatrix}$$

Strassen's trick is to calculate the following 7 products rather than 8 in C. Let

$$P1 = X(F-H)$$

$$P2 = (X+Y)H$$

$$P3 = (Z+W)E$$

$$P4 = W(G-E)$$

$$P5 = (X+W)(E+H)$$

$$P6 = (Y-W)(G+H)$$

$$P7 = (X-Z)(E+F)$$

Therefore,

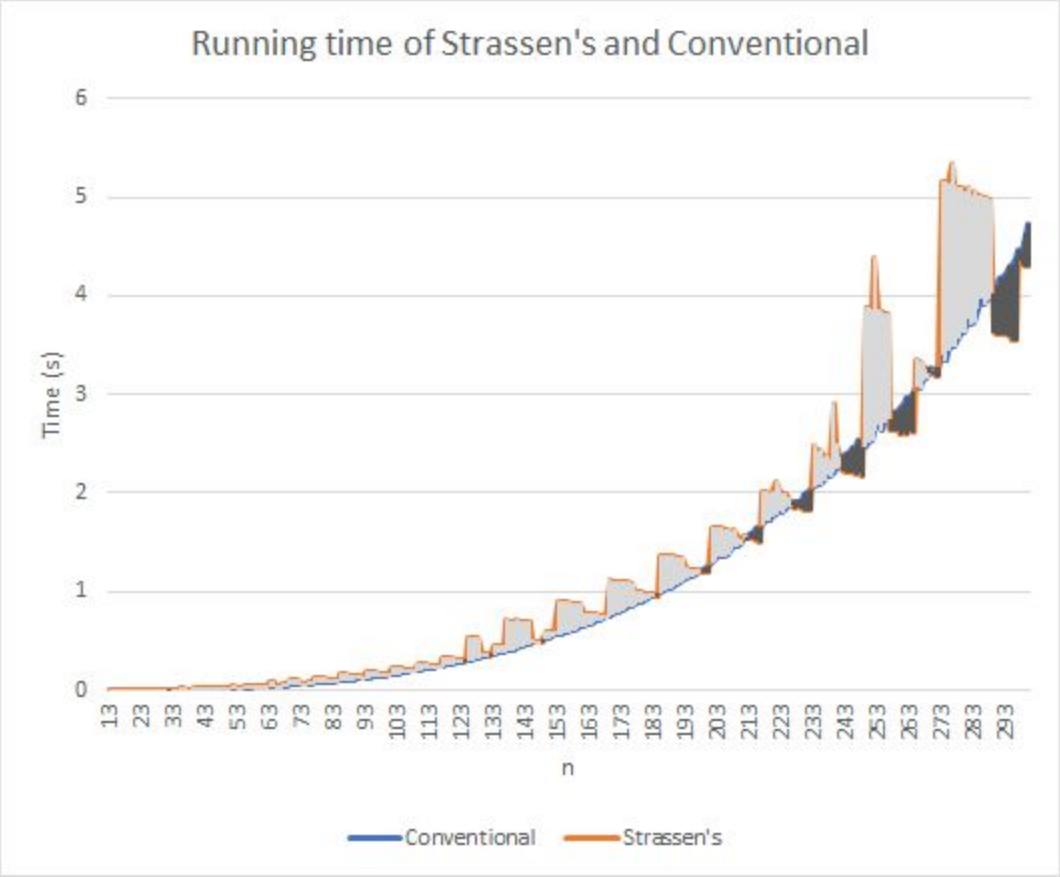
$$C = AB = \begin{pmatrix} P5+P4-P2+P6 & P1+P2 \\ P3+P4 & P5+P1-P3-P7 \end{pmatrix}$$

We implement conventional and Strassen's algorithm to Python 3. Input file contains 2 matrices where each entry is a random number from 0 and ± 1. In order to find the optimal cross-over point, we compare running time of both algorithms for different dimensions. But we cannot find an *n* such that Strassen's algorithm runs faster than the conventional one until

Yigao Li & Tao Yu
ANLY 550
Professor: Justin Thaler
April 25, 2018

300-dimension. The first *n* that Strassen's is the better one happens when $n_0 = 200$. We expect this number to be close to 37, which is not a similar result comparing to what we concluded from analytical approach. There are some reasons that this occurs. First, Strassen's cannot split an odd-dimension matrix into 4 submatrices. To resolve this issue, we pad zeros to make it even, but

Inserting 0s to a nested list (matrix) takes time. Also, computer processor and memory are factors that influence running time. We considered a less complicated type of matrices may save time, but as long as the input are integer matrices, running time did not change significantly.



Conclusion:

The crossover point is $n_0 = 200$ base on the experimental result. It is very different from what we got from the analytical approach. This could be explained by the fact that all arithmetic operations are assumed to cost 1 in the analytical approximation, while in reality the operations

have distinct cost values. Some operations were even considered as free in the analytical approach, while experimentally these operations are not cost-free.