MATH 102 – BONUS HOMEWORK ASSIGNMENT

Due Friday, December 1st, 2017 before the lecture.
Handwritten submissions only.

Completing the following bonus exercise is completely voluntary. The points of the bonus exercise can contribute to your score, but they are not part of the maximal point number.

Bonus Exercise (1+2+2+2+2+2+1)**.**
On a typical computer, performing multiplication and division of numbers requires much more computing time than performing addition and subtraction of numbers. For that reason, given a computational task for a computer, we want to let the computer solve that task using as few numerical multiplications and divisions as possible, even if that comes at the price of additional numerical additions and subtractions as a trade-off.

In this problem we take a look at matrix–matrix multiplication as an example. Let $A \in \mathbb{R}^{2n \times 2n}$ and $B \in \mathbb{R}^{2n \times 2n}$ be square matrices of the same size with their dimensions being a multiple of two ($2n$). The task for the computer is to compute their product $C = A \cdot B$, which is again a $2n \times 2n$ matrix. It is easily seen that the standard method for matrix–matrix multiplication requires $8n^3$ numerical multiplications: we have to compute $4n^2$ entries of the resulting matrix, and each of these entries is computed by adding up $2n$ numerical products.

Surprisingly, one can do better! A famous method by Volker Strassen from 1969 computes the product $C = AB$ with fewer numerical multiplications. First, we write the matrices A and B in block form with blocks of size $n \times n$:

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right), \quad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right).$$

We then compute the following seven $n \times n$ matrices:

$$\begin{gathered} M_1 = (A_{11} + A_{22})(B_{11} + B_{22}), \\ M_2 = (A_{21} + A_{22}) B_{11}, \quad M_3 = A_{11}(B_{12} - B_{22}), \\ M_4 = A_{22}(B_{21} - B_{11}), \quad M_5 = (A_{11} + A_{12}) B_{22}, \\ M_6 = (A_{21} - A_{11})(B_{11} + B_{12}), \quad M_7 = (A_{12} - A_{22})(B_{21} + B_{22}). \end{gathered}$$

One can show that the product C can be written in terms of $n \times n$ matrix blocks as follows:

$$(*) \qquad C = \left(\begin{array}{c|c} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ \hline M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{array} \right).$$

This manner of computing the matrix C is known as the *Strassen method* for matrix–matrix multiplication. In this bonus task, we explore the algorithmic theory of the Strassen method:

(a) First, explain why the *standard method* for computing the product $AB \in \mathbb{R}^{2n \times 2n}$ uses $8n^3$ numerical multiplications.
(b) Show that the identity $(*)$ is actually true, using the definitions of the seven matrices $M_1, \dots, M_7$.
(c) How many numerical multiplications do you need for the Strassen method $(*)$ in special case $n = 1$, i.e., when multiplying matrices of size 2×2?
(d) How many numerical multiplications do you need for the Strassen method $(*)$ in special case $n = 2$, i.e., when multiplying matrices of size 4×4, assuming that the seven 2×2 matrix–matrix products are computed by the standard method?
(e) How many numerical multiplications do you need for the Strassen method $(*)$ in special case $n = 2$, this time assuming that the seven 2×2 matrix–matrix products are computed in turn by the Strassen method for 2×2 matrices?

(f) Finally, suppose that the new method of computation is applied recursively to compute the seven $n \times n$ matrix–matrix multiplications in the definition of $M_1, \ldots, M_7$. Show that the computation of the $2n \times 2n$ matrix $C = AB$ then costs $7^{\log_2(2n)}$ numerical multiplications.

(g) Show that

$$7^{\log_2(2n)} = 7n^{\log_2(7)} \approx 7n^{2.80735}.$$

The net result is that the multiplication of $2n \times 2n$ matrices can be implemented on a computer using only numerical multiplications. There are more advanced algorithms that lower the exponent of n even further: the best known bound has an exponent of 2.3728639, and it is an unproven conjecture (related to combinatorial group theory) that an exponent of only 2 can be achieved.