Applied Math 5610 **Homework 1** **Due: January 23, 2019**
Bengt Fornberg

1. Straightforward block partitioned 2×2 matrix multiplication $C = A\,B$ can be written

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

where

$$\begin{cases} C_{11} = A_{11}B_{11} + A_{12}B_{21} \\ C_{12} = A_{11}B_{12} + A_{12}B_{22} \\ C_{21} = A_{21}B_{11} + A_{22}B_{21} \\ C_{22} = A_{21}B_{12} + A_{22}B_{22} \end{cases}$$

In the scheme discovered by Strassen (1969), the computations are rearranged into

$$\begin{cases} P_1 = (A_{11} + A_{22})(B_{11} + B_{22}) \\ P_2 = (A_{21} + A_{22})B_{11} \\ P_3 = A_{11}(B_{12} - B_{22}) \\ P_4 = A_{22}(B_{21} - B_{11}) \\ P_5 = (A_{11} + A_{12})B_{22} \\ P_6 = (A_{21} - A_{11})(B_{11} + B_{12}) \\ P_7 = (A_{12} - A_{22})(B_{21} + B_{22}) \end{cases}$$

$$\begin{cases} C_{11} = P_1 + P_4 - P_5 + P_7 \\ C_{12} = P_3 + P_5 \\ C_{21} = P_2 + P_4 \\ C_{22} = P_1 - P_2 + P_3 + P_6 \end{cases}$$

a. Analytically (by hand or using Mathematica), verify that the Strassen algorithm will give the same result as the traditional algorithm (in particular, check that we never assume that matrix multiplication is commutative).

b. Suppose that we start Strassen's method with two $n \times n$ matrices where $n = 2^m$ and use the algorithm repeatedly until all matrices have become of size 1×1. Show (for example using induction) that the total operation count becomes $7 \cdot 7^m - 6 \cdot 4^m$ operations (i.e. $O(n^{\log_2 7}) \approx O(n^{2.81})$).

c. Verify numerically that Strassen's algorithm indeed works by means of running the following Matlab code for a few cases with random input matrices of size $n = 2^m$, $m = 4, 5, 6, \ldots$ (as far as your computer allows) and compare the result and computer time (for ex. using the profiler or the tic and toc commands) against regular matrix multiplication.

```
function c = strass(a,b)
nmin = 16;
[n,n] = size(a);
if n<= nmin
   c = a*b;
else
   m = n/2; u = 1:m; v = m+1:n;
   p1 = strass(a(u,u)+a(v,v),b(u,u)+b(v,v));
   p2 = strass(a(v,u)+a(v,v),b(u,u));
   p3 = strass(a(u,u),b(u,v)-b(v,v));
   p4 = strass(a(v,v),b(v,u)-b(u,u));
   p5 = strass(a(u,u)+a(u,v),b(v,v));
   p6 = strass(a(v,u)-a(u,u),b(u,u)+b(u,v));
   p7 = strass(a(u,v)-a(v,v),b(v,u)+b(v,v));
   c  = [p1+p4-p5+p7, p3+p5; p2+p4, p1-p2+p3+p6];
end
```

Comment: The asymptotically fastest algorithm known at present for multiplying two general $n \times n$ matrices requires $O(n^{2.373})$ operations. It has been conjectured that it is possible to multiply two general full $n \times n$ matrices in $O(n^2 \log n)$ operations.

2. Solve Problems 2 a-d from Atkinson, pp. 648-649.

3. Give an example of a 2×2 matrix which has a Gershgorin circle without an eigenvalue inside it.

4. Prove the following extension of Gershgorin's theorem:

If (1) λ is on the edge of the Gershgorin set, and
(2) A is irreducible,
then λ lies on the edge of ***every*** Gershgorin circle.

Hint: In the proof of the regular Gershgorin Theorem in Atkinson (pages 589-590), note that on the line

$$|\lambda - a_{kk}||x_k| \le \sum_{j=1;\, j\ne k}^{n} |a_{kj}||x_j| \le r_k \|x\|_\infty \ ,$$

we must have $|\lambda - a_{kk}| = r_k$. This implies that $|x_j| = \|x\|_\infty$ for all j with $a_{kj} \ne 0$. Irreducibility allows us to choose a $j = k_1 \ne k$ that permits the argument to be repeated. Proceeding like that leads to $|\lambda - a_{mm}| = r_m$ for $m = 1, 2, \ldots, n$, i.e. λ is on the edge of every Gershgorin circle.

The hint just above was a very brief outline – expand on it to make the argument both solid and easily comprehensible!