UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences

Examination in MAT-INF4130 — Numerical linear algebra

Day of examination: Friday, December 15, 2017

Examination hours: 14:30 – 18:30

This problem set consists of 5 pages.

Appendices: None.

Permitted aids: None

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Problem 1 Matrix multiplication

Let $\mathbf{A}$ and $\mathbf{B}$ be $n \times n$ real matrices. The operation count (additions/multiplications) required to compute the product $\mathbf{AB}$ is $2n^3$. You do not have to show this.

1a

Consider the $2n \times 2n$ block matrix

$$\begin{bmatrix} \mathbf{W} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix},$$

where all matrices $\mathbf{A}, \ldots, \mathbf{Z}$ are in $\mathbb{R}^{n \times n}$. How many operations does it take to compute $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ by the obvious algorithm?

Possible solution: We have that

$$\mathbf{W} = \mathbf{AE} + \mathbf{BG}.$$

So it takes $4n^3 + n^2$ operations to compute this. We must compute 4 such matrices, hence the total operation cost is $16n^3 + 4n^2$.

(Continued on page 2.)

1b

An alternative method to compute $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ is to use Strassen's formulas:

$$\mathbf{P}_1 = (\mathbf{A}+\mathbf{D})(\mathbf{E}+\mathbf{H}), \quad \mathbf{P}_5 = (\mathbf{A}+\mathbf{B})\mathbf{H},$$
$$\mathbf{P}_2 = (\mathbf{C}+\mathbf{D})\mathbf{E}, \quad \mathbf{P}_6 = (\mathbf{C}-\mathbf{A})(\mathbf{E}+\mathbf{F}),$$
$$\mathbf{P}_3 = \mathbf{A}(\mathbf{F}-\mathbf{H}), \quad \mathbf{P}_7 = (\mathbf{B}-\mathbf{D})(\mathbf{G}+\mathbf{H}),$$
$$\mathbf{P}_4 = \mathbf{D}(\mathbf{G}-\mathbf{E})$$
$$\mathbf{W} = \mathbf{P}_1+\mathbf{P}_4-\mathbf{P}_5+\mathbf{P}_7, \quad \mathbf{Y} = \mathbf{P}_2+\mathbf{P}_4,$$
$$\mathbf{X} = \mathbf{P}_3+\mathbf{P}_5, \quad \mathbf{Z} = \mathbf{P}_1+\mathbf{P}_3-\mathbf{P}_2+\mathbf{P}_6.$$

You do not have to verify these formulas. What is the operation count for this method?

Possible solution: $\mathbf{P}_1$, $\mathbf{P}_6$ and $\mathbf{P}_7$ each need $2n^2+2n^3$ operations. $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ and $\mathbf{P}_5$ each need n^2+2n^3 operations. Hence forming the $\mathbf{P}$'s needs $3(2n^2+2n^3)+4(n^2+2n^3) = 10n^2+14n^3$ operations. To find the final result demands $8n^2$ operations. Thus the total cost is $14n^3+18n^2$.

1c

Describe a recursive algorithm, based on Strassen's formulas, which given two matrices $\mathbf{A}$ and $\mathbf{B}$ of size $m \times m$, with $m = 2^k$ for some $k \geq 0$, calculates the product $\mathbf{AB}$.

Possible solution: Here is a program that seems to work:

```
function Z=strassen(A,B)
[m,~]=size(A);
if m==1
  Z=A*B; return;
end
one=1:m/2; two=m/2+1:m;
P1=strassen(A(one,one)+A(two,two),B(one,one)+B(two,two));
P2=strassen(A(two,one)+A(two,two),B(one,one));
P3=strassen(A(one,one),B(one,two)-B(two,two));
P4=strassen(A(two,two),B(two,one)-B(one,one));
P5=strassen(A(one,one)+A(one,two),B(two,two));
P6=strassen(A(two,one)-A(one,one),B(one,one)+B(one,two));
P7=strassen(A(one,two)-A(two,two),B(two,one)+B(two,two));
Z=[P1+P4-P5+P7,P3+P5; P2+P4,P1+P3-P2+P6];
```

1d

Show that the operation count of the recursive algorithm is $\mathcal{O}\left(m^{\log_2(7)}\right)$. Note that $\log_2(7) \approx 2.8 < 3$, so this is less costly than straightforward matrix multiplication.

(Continued on page 3.)

Possible solution: Let s_k be the cost of Strassen multiplication of two matrices of size 2^k. Then, up to leading order $s_{k+1} = 7s_k$. This means that $s_k = \gamma 7^k$ for some constant γ. We have $m = 2^k$ and $7 = 2^{\log_2(7)}$, so $s_k = 2^{k \log_2(7)} = m^{\log_2(7)}$.

Problem 2 SVD

Let $\mathbf{A} \in \mathbb{C}^{m \times n}$, with $m \geq n$, be a matrix on the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} \\ \mathbf{C} \end{bmatrix}$$

where $\mathbf{B}$ is a non-singular $n \times n$ matrix and $\mathbf{C}$ is in $\mathbb{C}^{(m-n) \times n}$. Let $\mathbf{A}^\dagger$ denote the pseudoinverse of $\mathbf{A}$. Show that $\left\|\mathbf{A}^\dagger\right\|_2 \leq \left\|\mathbf{B}^{-1}\right\|_2$. (**Hint:** use the singular value factorization of $\mathbf{A}$, i.e., $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ with $\mathbf{U}$ and $\mathbf{V}$ unitary.)

Possible solution: Observe first that $\mathbf{A}^*\mathbf{A} = \mathbf{B}^*\mathbf{B} + \mathbf{C}^*\mathbf{C}$, so that the smallest singular value of $\mathbf{A}$; λ_A, is larger or equal to the smallest singular value of $\mathbf{B}$; λ_B. We have that $\left\|\mathbf{B}^{-1}\right\| = 1/\lambda_B$.
We have that $\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*$, where $\mathbf{V}$ and $\mathbf{U}$ are unitary. $\mathbf{\Sigma}$ is diagonal with the singular values of $\mathbf{A}$ on the diagonal. For $\mathbf{x} \in \mathbb{C}^m$,

$$\begin{aligned}
\left\|\mathbf{A}^\dagger\mathbf{x}\right\|_2 &= \left\|\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*\mathbf{x}\right\|_2 \\
&= \left\|\mathbf{\Sigma}^{-1}\mathbf{U}^*\mathbf{x}\right\|_2 && (\mathbf{V} \text{ is unitary}) \\
&\leq \left\|\mathbf{\Sigma}^{-1}\right\|_2 \left\|\mathbf{U}^*\mathbf{x}\right\|_2 && (\|\cdot\|_2 \text{ subordinate to vector norm}) \\
&= \left\|\mathbf{\Sigma}^{-1}\right\|_2 \left\|\mathbf{x}\right\|_2 && (\mathbf{U}^* \text{ unitary}) \\
&= \frac{1}{\lambda_A} \left\|\mathbf{x}\right\|_2 \\
&\leq \frac{1}{\lambda_B} \left\|\mathbf{x}\right\|_2 \\
&= \left\|\mathbf{B}^{-1}\right\|_2 \left\|\mathbf{x}\right\|_2 .
\end{aligned}$$

Problem 3 An iterative method

Assume that $\mathbf{A} \in \mathbb{C}^{n \times n}$ is non-singular and nondefective (the eigenvectors of $\mathbf{A}$ form a basis for $\mathbb{C}^n$). We wish to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$. Assume that we have a list of the eigenvalues, in no particular order, $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$. We have that $m \leq n$, since some of the eigenvalues may have multiplicity larger than one. Given $\mathbf{x}_0 \in \mathbb{C}^n$, and $k \geq 0$, we define the sequence $\{\mathbf{x}_k\}_{k=0}^{m-1}$ by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{\lambda_{k+1}}\mathbf{r}_k, \text{ where } \mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k.$$

(Continued on page 4.)

3a

Let the coefficients c_{ik} be defined by

$$\mathbf{r}_k = \sum_{i=1}^{n} c_{ik}\mathbf{u}_i,$$

where $\{(\sigma_i, \mathbf{u}_i)\}_{i=1}^{n}$ are the eigenpairs of $\mathbf{A}$. Show that

$$c_{i,k+1} = \begin{cases} 0 & \text{if } \sigma_i = \lambda_{k+1}, \\ c_{i,k}\left(1 - \frac{\sigma_i}{\lambda_{k+1}}\right) & \text{otherwise.} \end{cases}$$

Possible solution: Observe that $\mathbf{A}\mathbf{u}_j = \sigma_j\mathbf{u}_j$, where $\sigma_j \in \{\lambda_1, \dots, \lambda_m\}$. We have that

$$\begin{aligned} \mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_{k+1} &= \mathbf{b} - \mathbf{A}\left(\mathbf{x}_k + \frac{1}{\lambda_{k+1}}\mathbf{r}_k\right) \\ &= \mathbf{r}_k - \frac{1}{\lambda_{k+1}}A\mathbf{r}_k \\ &= \sum_i c_{ik}\left(1 - \frac{\sigma_i}{\lambda_{k+1}}\right)\mathbf{u}_i. \end{aligned}$$

Hence

$$c_{i,k+1} = c_{i,k}\left(1 - \frac{\sigma_i}{\lambda_{k+1}}\right).$$

3b

Show that for some $l \le m$, we have that $\mathbf{x}_l = \mathbf{x}_{l+1} = \cdots = \mathbf{x}_m = \mathbf{x}$, where $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Possible solution: After at most m iterations we will have $c_{i,k} = 0$ for all i, and thus $\mathbf{x}_k = \mathbf{x}$.

3c

Consider this iteration for the $n \times n$ matrix $\mathbf{T} = \text{tridiag}(c, d, c)$, where d and c are positive real numbers and $d > 2c$. The eigenvalues of $\mathbf{T}$ are

$$\lambda_j = d + 2c\cos\left(\frac{j\pi}{n+1}\right), \qquad j = 1, \dots, n.$$

What is the operation count for solving $\mathbf{T}\mathbf{x} = \mathbf{b}$ using the iterative algorithm above?

Possible solution: Each iterative step consists in finding $\mathbf{x} + (\mathbf{b} - \mathbf{T}\mathbf{x})/\lambda$. The matrix multiplication $\mathbf{T}\mathbf{x}$ requires $3n$ flops, we have to add $\mathbf{b}$ (n flops), divide by λ (n flops), and add $\mathbf{x}$ (n flops). Altogether $3n + n + n + n = 6n$ flops. The iteration reaches a fixpoint in at most n steps, so the total count is $6n^2$.

(Continued on page 5.)

3d

Let now $\mathbf{B}$ be an $n \times n$ matrix which is zero on the "tridiagonal", i.e., $b_{ij} = 0$ if $|i - j| \leq 1$. Set $\mathbf{A} = \mathbf{T} + \mathbf{B}$, where $\mathbf{T}$ is the tridiagonal matrix above. We wish to solve $\mathbf{Ax} = \mathbf{b}$ by the iterative scheme

$$\mathbf{Tx}_{k+1} = \mathbf{b} - \mathbf{Bx}_k.$$

Show that this iteration will converge if

$$\min\left\{\max_i \sum_{j=1}^{n} |b_{ij}|, \max_j \sum_{i=1}^{n} |b_{ij}|\right\} < d - 2c.$$

(**Hint**: use Gershgorin's circle theorem for the eigenvalues of $\mathbf{B}$).

Possible solution: The iterative scheme can be written

$$\mathbf{x}_{k+1} = -\mathbf{T}^{-1}\mathbf{Bx}_k + \mathbf{T}^{-1}\mathbf{b} =: \mathbf{G}x_k + \mathbf{c}.$$

The iteration will converge if $\rho(\mathbf{G}) < 1$. We have that $\rho(\mathbf{G}) \leq \rho(\mathbf{T}^{-1})\rho(\mathbf{B})$. The eigenvalues of $\mathbf{T}$ satisfy

$$\lambda_j \geq d - 2c, \qquad \frac{1}{\lambda_j} \leq \frac{1}{d - 2c}.$$

Hence $\rho(\mathbf{T}^{-1}) \leq 1/(d - 2c) < \infty$. Regarding the eigenvalues of $\mathbf{B}$, by Gershgorin's theorem,

$$|\mu_j| \leq \min\left\{\max_i \sum_{\substack{j \\ |i-j|>1}} |b_{ij}|, \max_j \sum_{\substack{i \\ |i-j|>1}} |b_{ij}|\right\} =: C_{\mathbf{B}}.$$

The algorithm will converge if

$$C_{\mathbf{B}} < d - 2c.$$

THE END