# Chapter1

Guorui Zhu

2024 年 9 月 11 日

## 1

### 1.1

Let $A$ be an orthogonal matrix. Show that $\det(A) = \pm 1$. Show that if $B$ also is orthogonal and $\det(A) = -det(B)$, then $A + B$ is singular.

proof:

### 1.2

The rank of a matrix is the dimension of the space spanned by its columns. Show that $A$ has rank one if and only if $A = ab^\top$ for some column vectors $a$ and $b$.

proof:

### 1.3

Show that if a matrix is orthogonal and trian- gular, then it is diagonal. What are its diagonal elements.

proof:

### 1.4

A matrix is strictly upper triangular if it is upper triangular with zero diagonal elements. Show that if $A$ is strictly upper triangular and $n$-by-$n$, then $A^n = 0$.

proof:

### 1.5

Let $\|\cdot\|$ be a vector norm on $\mathbb{R}^m$ and assume that $C \in \mathbb{R}^{m\times n}$. Show that if $\text{rank}(C) = n$, then $\|x\|_C := \|Cx\|$ is a vector norm.

proof:

### 1.6

Show that if $0 \neq s \in \mathbb{R}^n$ and $E \in \mathbb{R}^{n \times n}$, then

$$\left\| E(I - \frac{ss^\top}{s^\top s}) \right\|_F^2 = \|E\|_F^2 - \frac{\|Es\|_2^2}{s^\top s}.$$

proof:

### 1.7

Verify that $\left\|xy^H\right\|_F = \left\|xy^H\right\|_2 = \|x\|_2 \|y\|_2$ for any $x, y \in \mathbb{C}^n$.

proof:

### 1.8

One can identify the degree $d$ polynomials $p(x) = \sum_{i=0}^{d} a_i x^i$ with $\mathbb{R}^{d+1}$ via the vector of coefficients. Let $x$ be fixed. Let $S_x$ be the set of polynomials with an infinite relative condition number with respect to evaluating them at $x$ (i.e., they are zero at $x$). In a few words, describe $S_x$ geometrically as a subset of $\mathbb{R}^{d+1}$. Let $S_x(\kappa)$ be the set of polynomials whose relative condition number is $\kappa$ or greater. Describe $S_x(\kappa)$ geometrically in a few words. Describe how $S_x(\kappa)$ changes geometrically as $\kappa \to \infty$.

proof:

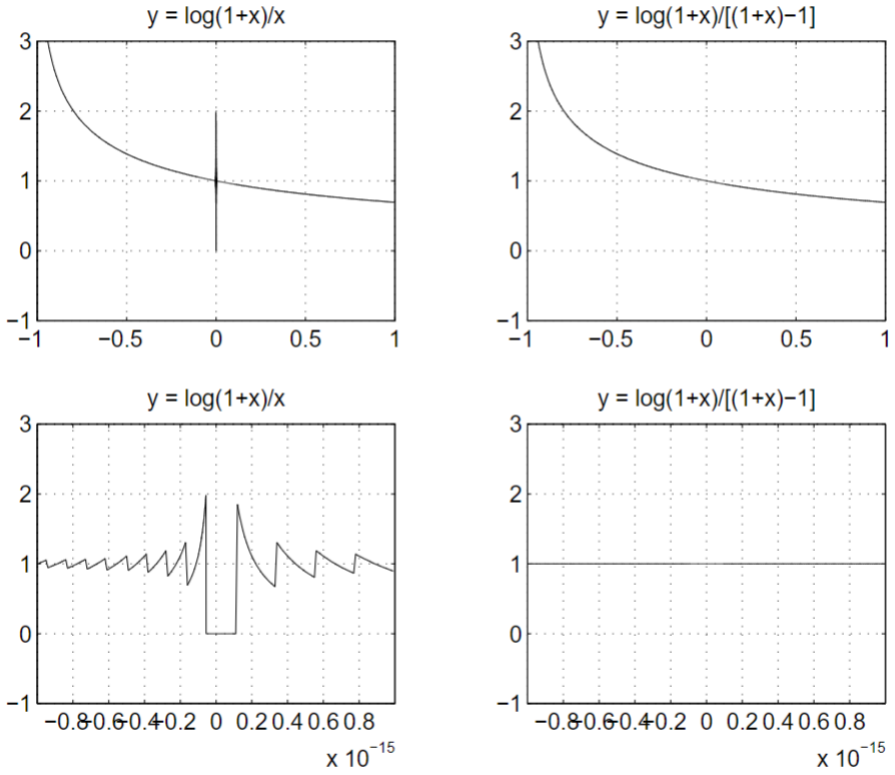
### 1.9

Consider the figure below. It plots the function $y = \frac{\log(1+x)}{x}$ computed in two different ways. Mathematically, $y$ is a smooth function of $x$ near $x = 0$, equaling 1 at 0. But if we compute $y$ using this formula, we get the plots in section 1.9 on the left (shown in the ranges $x \in [-1, 1]$ on the top left and $x \in [-10^{-15}, 10^{-15}]$ on the bottom left). This formula is clearly unstable near $x = 0$. On the other hand, if we use the algorithm

$d = 1 + x$
if $d = 1$ then
    $y = 1$
else
    $y = \dfrac{\log(d)}{d - 1}$
end if

we get the two plots on the right, which are correct near $x = 0$. Explain this phenomenon, proving that the second algorithm must compute an accurate answer in floating point arithmetic. Assume that the log function returns an accurate answer for any argument. (This is true of any reasonable implemen- tation of logarithm.) Assume IEEE floating point arithmetic if that makes your argument easier. (Both algorithms can malfunction on a Cray machine.)

proof:




## 1.10

Show that, barring overflow or underflow, $\mathrm{fl}(\sum_{i=1}^{d} x_i y_i) = \sum_{i=1}^{d} x_i y_i(1+\delta_i)$, where $|\delta_i| \leq d\epsilon$. Use this to prove the following fact. Let $A^{m\times n}$ and $B^{n\times p}$ be matrices, and compute their product in the usual way. Barring overflow or underflow show that $|\mathrm{fl}(A\cdot B) - A\cdot B| \leq n\cdot\epsilon\cdot|A|\cdot|B|$. Here the absolute value of a matrix $|A|$ means the matrix with entries $(|A|)_{ij} = |a_{ij}|$, and the inequality is meant componentwise. The result of this question will be used in section 2.4.2, where we analyze the roundoff errors in Gaussian elimination.

proof:

## 1.11

Let $L$ be a lower triangular matrix and solve $Lx = b$ by forward substitution. Show that barring overflow or underflow, the computed solution $\hat{x}$ satisfies $(L+\delta L)\hat{x} = b$, where $|\delta l_{ij}| \leq n\epsilon|l_{ij}|$, where $\epsilon$ is the machine precision. This means that forward substitution is backward stable. Argue that backward substitution for solving upper triangular systems satisfies the same bound. The result of this question will be used in section 2.4.2, where we analyze the roundoff errors in Gaussian elimination.

proof:

### 1.12

In order to analyze the effects of rounding errors, we have used the following model (see equation (1.1)):

$$\mathrm{fl}(a \odot b) = (a \odot b)(1 + \delta)$$

where $\odot$ is one of the four basic operations $+, -, *$, and $/$, and $|\delta| \leq \epsilon$. To show that our analyses also work for complex data, we need to prove an analogous formula for the four basic complex operations. Now $\delta$ will be a tiny complex number bounded in absolute value by a small multiple of $\epsilon$ . Prove that this is true for complex addition, subtraction, multiplication, and division. Your algorithm for complex division should successfully compute $a/a \approx 1$, where $|a|$ is either very large (larger than the square root of the overflow threshold) or very small (smaller than the square root of the underflow threshold). Is it true that both the real and imaginary parts of the complex product are always computed to high relative accuracy?

proof:

### 1.13

Prove lemma 1.3.: Let $\mathcal{B} = \mathbb{R}^n$ (or $\mathbb{C}^n$) and $\langle \cdot\,, \cdot \rangle$ be an inner product. Then there is an $n$-by-$n$ s.p.d. (h.p.d.) matrix $A$ such that $\langle x, y\rangle = y^T Ax\,(y^* Ax)$. Conversely, if $A$ is s.p.d (h.p.d.), then $y^T Ax\,(y^* Ax)$ is an inner product.

proof:

### 1.14

Prove lemma 1.5: $\forall x \in \mathbb{R}^n$,

$$\begin{aligned} \|x_2\| &\leq \|x\|_1 \leq \sqrt{n}\|x_2\|_2, \\ \|x_2\|_\infty &\leq \|x_2\|_2 \leq \sqrt{n}\|x_2\|_\infty, \\ \|x_2\|_\infty &\leq \|x_2\|_1 \leq n\|x_2\|_\infty \end{aligned}$$

.

proof:

### 1.15

Prove lemma 1.6.: An operator norm is a matrix norm.

proof:

### 1.16

Prove all parts except 7 of Lemma 1.7. Hint for part 8: Use the fact that if $X$ and $Y$ are both $n$-by-$n$, then $XY$ and $YX$ have the same eigenvalues. Hint for part 9: Use the fact that a matrix is normal if and only if it has a complete set of orthonormal eigenvectors.

proof:

### 1.17

We mentioned that on a Cray machine the expression $\arccos\left(x/\sqrt{x^2+y^2}\right)$ caused an error, because roundoff caused $(x/\sqrt{x^2+y^2})$ to exceed 1. Show that this is impossible using IEEE arithmetic, barring overflow or underflow. Hint: You will need to use more than the simple model $fl(a \odot b) = (a \odot b)(1+\delta)$ with $|\delta|$ small. Think about evaluating $\sqrt{x^2}$, and show that, barring overflow or underflow, $fl(\sqrt{x^2}) = x$ exactly; in numerical experiments done by A. Liu, this failed about 5% of the time on a Cray YMP. You might try some numerical experiments and explain them. Extra credit: Prove the same result using correctly rounded decimal arithmetic. (The proof is different.) This question is due to W. Kahan, who was inspired by a bug in a Cray program of J. Sethian.

proof:

### 1.18

Suppose $a$ and $b$ are normalized IEEE double precision floating point numbers, and consider the following algorithm, running with IEEE arithmetic:

$$\begin{aligned}
&\text{IF}(|a| < |b|), \text{swap} a \text{and} b \\
&s_1 = a + b \\
&s_2 = (a - s_1) + b
\end{aligned}$$

Prove the following facts:

1. Barring overflow or underflow, the only roundoff error committed in running the algorithm is computing $s_1 = fl(a+b)$. In other words, both subtractions $s_1 - a$ and $(s_1 - a) - b$ are computed exactly.

2. $s_1 + s_2 = a + b$,exactly. This means that $s_2$ is actually the roundoff error committed when rounding the exact value of $a+b$ to get $s_1$.

Thus, this program in effect simulates quadruple precision arithmetic, representing the true sum $a+b$ as the higher-order bits ($s_1$) and the lower-order bits ($s_2$). Using this and similar tricks in a systematic way, it is possible to effi- ciently simulate all four basic floating point operations in arbitrary precision arithmetic, using only the underlying floating point instructions and no "bit-fiddling" [202]. 128-bit arithmetic is implemented this way on the IBM RS6000 and Cray (but much less efficiently on the Cray, which does not have IEEE arithmetic).

proof:

# 2

## 2.1

Using your favorite World Wide Web browser, go toNETLIB (http://www.netlib.org), and answer the following questions

1. You need a Fortran subroutine to compute the eigenvalues and eigenvec-tors of real symmetric matrices in double precision. Find one using theAttribute/Value database search on the NETLIB repository. Report thename and URL of the subroutine as well as how you found it.
2. Using the Performance Database Server, find out the current world speedrecord for solving 100-by-100 dense linear systems using Gaussian elimination. What is the speed in Mflops, and which machine attained it? Dothe same for 1000-by-1000 dense linear systems and "big as you want" dense linear systems. Using the same database, find out how fast your work station can solve 100-by-100 dense linear systems. Hint: Look atthe LINPACK benchmark.

proof:

## 2.2

Consider solving $AX = B$ for $X$, where $A$ is $n$-by-$n$, and $X$ and $B$ are $n$-by-$m$. There are two obvious algorithms. The first algorithm factorizes $A = PLU$ using Gaussian elimination and then solves for each column of $X$ by forward and back substitution. The second algorithm computes $A^{-1}$ using Gaussian elimination and then multiplies $X = A^{-1}B$. Count the number of flops required by each algorithm, and show that the first one requires fewer flops.

proof:

## 2.3

Let $\|\cdot\|$ be the two-norm. Given a nonsingular matrix $A$ and a vector $b$, show that for sufficiently small $\|\delta A\|$ , there are nonzero $\delta A$ and $\delta b$ such that inequality (2.2) is an equality. This justifies calling $\kappa(A) = \|A^{-1}\|\dot{\|}A\|$ the condition number of $A$. Hint: Use the ideas in the proof of Theorem 2.1.

proof:

## 2.4

Show that bounds (2.7) and (2.8) are attainable.(fig. 1)

proof:

## 2.5

Prove Theorem 2.3. Given the residual $r = A\hat{x} - b$, use Theorem 2.3 to show that bound (2.9) is no larger than bound (2.7). This explains why LAPACK computes a bound based on (2.9), as described in section 2.4.4.

## 2.6

Prove Lemma 2.2.

Let $P$,$P_1$,and $P_2$ be $n$-by-$n$ permutation matrices and $X$ be an $n$-by-$n$ matrix. Then

1. $PX$ is the same as $X$ with its rows permuted. $XP$ is the same as $X$ with its columns permuted.
2. $p^{-1} = P^\top$.
3. $\det P = \pm 1$.
4. $P_1 \cdot P_2$ is also a permutation matrix.

proof:

## 2.7

If $A$ is a nonsingular symmetric matrix and has the factorization $A = LDM^T$ , where $L$ and $M$ are unit lower triangular matrices and $D$ is a diagonal matrix, show that $L = M$.

proof:

## 2.8

Consider the following two ways of solving a 2-by-2 linear system of equations:

$$Ax = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = b$$

Algorithm 1. Gaussian elimination with partial pivoting (GEPP).

Algorithm 2. Cramer's rule.

Show by means of a numerical example that Cramer's rule is not backward stable. Hint: Choose the matrix nearly singular and $[b_1 b_2]^\top \approx [a_{12} a_{22}]^\top$. What does backward stability imply about the size of the residual? Your numerical example can be done by hand on paper (for example, with four-decimal-digit floating point), on a computer, or a hand calculator.

proof:

## 2.9

Let $B$ be an $n$-by-$n$ upper bidiagonal matrix, i.e., nonzero only on the main diagonal and first superdiagonal. Derive an algorithm for computing $\kappa_\infty(B) := \|B\|_\infty \|B^{-1}\|_\infty$ exactly (ignoring roundoff ). In other words, you should not use an iterative algorithm such as Hager's estimator. Your algorithm should be as cheap as possible; it should be possible to do using no more than $2n - 2$ additions, $n$ multiplications, $n$ divisions, $4n - 2$ absolute values, and $2n - 2$ comparisons. (Anything close to this is acceptable.)

proof:

### 2.10

Let $A$ be $n$-by- $n$. Show that $\left\|A^\top A\right\|_2 = \|A\|_2^2$ and $\kappa_2\left(A^\top A\right) = \kappa_2(A)^2$.

proof:

### 2.11

Let $A$ be symmetric and positive definite. Show that $|a_{ij}| < \left(a_{ii}a_{jj}\right)^{1/2}.(i \neq j)$.

proof:

### 2.12

Show that if

$$Y = \begin{pmatrix} I_n & Z \\ 0 & I_n \end{pmatrix},$$

then $\kappa_{F(Y)} = \|Y\|_F \left\|Y^{-1}\right\|_F = 2n + \|Z\|_F^2$.

proof:

### 2.13

In this question we will ask how to solve $By = c$ given a fast way to solve $Ax = b$, where $A - B$ is "small" in some sense.

1. Prove the Sherman-Morrison formula: Let $A$ be nonsingular, $u$ and $v$ be column vectors, and $A + uv^\top$ be nonsingular. Then $\left(A + uv^\top\right)^{-1} = A^{-1} - \left(A^{-1}uv^\top A^{-1}\right) / \left(1 + v^\top A^{-1}u\right)$.

   More generally, prove the Sherman-Morrison-Woodbury formula: Let $U$ and $V$ be $n$ by- $k$ rectangular matrices, were $k \leq n$ and $A$ is $n$-by- $n$. Then $T = I + V^\top A^{-1}U$ is nonsingular if and only if $A + UV^\top$ is nonsingular, in which case $\left(A + UV^\top\right)^{-1} = A^{-1} - A^{-1}UT^{-1}V^\top A^{-1}$.

2. If you have a fast algorithm to solve $Ax = b$, show how to build a fast solver for $By = c$, where $B = A + uv^\top$.

3. Suppose that $\|A - B\|$ is "small" and you have a fast algorithm for solving $Ax = b$. Describe an iterative scheme for solving $By = c$. How fast do you expect your algorithm to converge? Hint: Use iterative refinement.

proof:

### 2.14 Programming,ommitted

### 2.15 Programming,ommitted

### 2.16

Show how to reorganize the Cholesky algorithm (Algorithm 2.11) to do most of its operations using Level 3 BLAS. Mimic Algorithm 2.10.

proof:

## 2.17

Let

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where $A_{11}$ is $k$-by-$k$ and nonsingular. Then $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ is called the Schur complement of $A_{11}$ in $A$, or just Schur complement for short.

1. Show that after $k$ steps of Gaussian elimination without pivoting, $A_{22}$ has been overwritten by $S$.
2. Suppose $A = A^T$, $A_{11}$ is positive definite and $A_{22}$ is negative definite ($-A_{22}$ is positive definite). Show that $A$ is nonsingular, that Gaussian elimination without pivoting will work in exact arithmetic, but (by means of a 2-by-2 example) that Gaussian elimination without pivoting may be numerically unstable.

proof:

## 2.18

Matrix $A$ is called column diagonally dominant, or diagonally dominant for short, if

$$|a_{ii}| > \sum_{j \neq i} |a_{ji}| .$$

1. Show that $A$ is nonsingular. Hint: Use Gershgorin's theorem
2. Show that Gaussian elimination with partial pivoting does not actually permute any rows, i.e., that is identical to GEWP. Hint: Show that after one step of Gaussian elimination, the trailing submatrix, the Schur complement of $a_{11}$ in $a$ , is still diagonally dominant.

proof:

## 2.19

Given an $n$-by-$n$ nonsingular matrix $A$, how do you efficiently solve the following problems, using Gaussian elimination with partial pivoting?

1. Solve the linear system $A^k x = b$, where $k$ is a positive integer.
2. Compute $\alpha = c^\top A^{-1} b$.
3. Solve the matrix equation $AX = B$, where $B$ is $n$-by-$m$.

proof:

### 2.20

Prove that Strassen's algorithm (Algorithm 2.8) correctly multiplies $n$-by-$n$ matrices, where $n$ is a power of 2.

proof:

## 3

### 3.1

Show that the two variations of Algorithm 3.1, CGS and MGS, are mathematically equivalent by showing that the two formulas for rji yield the same results in exact arithmetic.

proof:

### 3.2

This question will illustrate the difference in numerical stability among three algorithms for computing the QR factorization of a matrix:

- Householder QR (Algorithm 3.2),
- CGS (Algorithm 3.1),
- MGS (Algorithm 3.1)

Obtain the Matlab program QRStability.m from HOMEPAGE/Matlab/QRStability.m. This program generates random matrices with user-specified dimensions $m$ and $n$ and condition number *cnd*, computes their QR decomposition using the three algorithms, and measures the accuracy of the results. It does this with the residual $\frac{\|A-Q\cdot R\|}{\|A\|}$, which should be around machine epsilon $\epsilon$ for a stable algorithm, and the orthogonality of Q $\left\|Q^\top \cdot Q - I\right\|$, which should also be around $\epsilon$. Run this program for small matrix dimensions (such as $m = 6$ and $n = 4$), modest numbers of random matrices (samples= 20), and condition numbers ranging from $cnd = 1$ up to $cnd = 1015$. Describe what you see. Which algorithms are more stable than others? See if you can describe how large $\left\|Q^\top \cdot Q - I\right\|$ can be as a function of choice of algorithm, *cnd* and $\epsilon$.

proof:

### 3.3

Let $A$ be $m$-by-$n$, $m \geq n$, and have full rank.

- Show that $\begin{pmatrix} I & A \\ A^\top & 0 \end{pmatrix} \cdot \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ has a solution where $x$ minimizes $\|Ax - b\|_2$. One reason for this formulation is that we can apply iterative refinement to this linear system if we want a more accurate answer (see section 2.5).
- What is the condition number of the coefficient matrix, in terms of the singular values of $A$? Hint: Use the SVD of $A$.

- Give an explicit expression for the inverse of the cofficient matrix, as a block 2-by-2 matrix. Hint: Use 2-by-2 block Gaussian elimination. Where have we previously seen the $(2,1)$ block entry?
- Show how to use the QR decomposition of $A$ to implement an iterative refinement algorithm to improve the accuracy of $x$.

proof:

### 3.4

Weighted least squares: If some components of $Ax-b$ are more important than others, we can weight them with a scale factor $d_i$ and solve the weighted least squares problem $\min \|D(Ax-b)\|_2$ instead, where $D$ has diagonal entries $d_i$. More generally, recall that if $C$ is symmetric positive definite, then $\|x\|_C \equiv \left(x^T C x\right)^{1/2}$ is a norm, and we can consider minimizing $\|Ax-b\|_C$. Derive the normal equations for this problem, as well as the formulation corresponding to the previous question.

proof:

### 3.5

Let $A \in \mathbb{R}^{n\times n}$ be positive definite. Two vectors $u_1$ and $u_2$ are called $A$-orthogonal if $u_1^T A u_2 = 0$. If $U \in \mathbb{R}^{n\times r}$ and $U^T A U = I$, then the columns of $U$ are said to be $A$-orthonormal. Show that every subspace has an $A$-orthonormal basis.

proof:

### 3.6

Let $A$ have the form $A = \begin{pmatrix} R \\ S \end{pmatrix}$, where $R$ is $n$-by-$n$ and upper triangular, and $S$ is $m$-by-$n$ and dense. Describe an algorithm using Householder transformations for reducing $A$ to upper triangular form. Your algorithm should not "fill in" the zeros in $R$ and thus require fewer operations than would Algorithm 3.2 applied to $A$.

proof:

### 3.7

If $A = R + uv^T$, where $R$ is an upper triangular matrix, and $u$ and $v$ are column vectors, describe an efficient algorithm to compute the QR decomposition of $A$. Hint: Using Givens rotations, your algorithm should take $O\left(n^2\right)$ operations. In contrast, Algorithm 3.2 would take $O\left(n^3\right)$ operations.

proof:

### 3.8

Let $x \in \mathbb{R}^n$ and let $P$ be a Householder matrix such that $Px = \pm\|x\|_2 e_1$. Let $G_{1,2}, \ldots, G_{n-1,n}$ be Givens rotations, and let $Q = G_{1,2} \cdots G_{n-1,n}$. Suppose $Qx = \pm\|x\|_2 e_1$. Must $P$ equal $Q$ ? (You need to give a proof or a counterexample.)

proof:

### 3.9

Let $A$ be $m$-by-$n$, with SVD $A = U\Sigma V^T$ . Compute the SVDs of the following matrices in terms of $U, \Sigma$, and $V$:

1. $(A^T A)^{-1}$,
2. $(A^T A)^{-1} A^T$,
3. $A(A^T A)^{-1}$,
4. $A(A^T A)^{-1} A^T$.

proof:

### 3.10

Let $A_k$ be a best rank-$k$ approximation of the matrix $A$, as defined in Part 9 of Theorem 3.3. Let $\sigma_i$ be the $i$-th singular value of $A$. Show that $A_k$ is unique if $\sigma_k > \sigma_{k+1}$.

proof:

### 3.11

Let $A$ be $m$-by-$n$. Show that $X = A^\dagger$ (the Moore-Penrose pseudoinverse) minimizes $\|AX - I\|_F$ over all $n$-by-$m$ matrices $X$. What is the value of this minimum?

proof:

### 3.12

Let $A, B$, and $C$ be matrices with dimensions such that the product $A^T C B^T$ is well defined. Let $\mathcal{X}$ be the set of matrices $X$ minimizing $\|AXB - C\|_F$ , and let $X_0$ be the unique member of $\mathcal{X}$ minimizing $\|X\|_F$ . Show that $X_0 = A^\dagger C B^\dagger \|$. Hint: Use the SVDs of $A$ and $B$.

proof:

### 3.13

Show that the Moore-Penrose pseudoinverse of A satisfies the following identities:

1. $AA^+A = A$,
2. $A^+AA^+ = A^+$,

3. $A^+A = (A^+A)^\top$,

4. $AA^+ = (AA^+)^\top$.

proof:

### 3.14

Prove part4 of Theorem 3.3 : Let $H = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$, where $A$ is square and $A = U\Sigma V^T$ is its SVD. Let $\Sigma = \text{diag}\,(\sigma_1, \ldots, \sigma_n)\,, U = [u_1, \ldots, u_n]$, and $V = [v_1, \ldots, v_n]$. Prove that the $2n$ eigenvalues of $H$ are $\pm\sigma_i$, with corresponding unit eigenvectors $\frac{1}{\sqrt{2}} \begin{bmatrix} v_i \\ \pm u_i \end{bmatrix}$. Extend to the case of rectangular $A$.

proof:

### 3.15

Let $A$ be $m$-by-$n$, $m < n$, and of full rank. Then $\min \|Ax - b\|_2$ is called an underdetermined least squares problem. Show that the solution is an $(n-m)$-dimensional set. Show how to compute the unique minimum norm solution using appropriately modified normal equations, QR decomposition, and SVD.

proof:

### 3.16

Prove Lemma 3.1.

Let $P$ be an exact Householder (or Givens) transformation, and $\tilde{P}$ be its floating point approximation. Then

$$\text{fl}(\tilde{P}A) = P(A+E), \quad \|E\|_2 = \text{O}(\epsilon)\|A\|_2,$$

and

$$\text{fl}(A\tilde{P}) = (A+F)P, \quad \|F\|_2 = \text{O}(\epsilon)\|A\|_2.$$

Sketch of Proof: Apply the usual formula $\text{fl}(a \odot b) = (a \odot b)(1+\epsilon)$ to the formulas for computing and applying $\tilde{P}$. See Question 3.16.

proof:

### 3.17

In section 2.6.3, we showed how to reorganize Gaussian elimination to perform Level 2 BLAS and Level 3 BLAS as each step in order to exploit the higher speed of these operations. In this problem, we will show how to apply a sequence of Householder transformations using Level 2 and Level 3 BLAS.

1. Let $u_1, \ldots, u_b$ be a sequence of vectors of dimension $n$, where $\|u_i\|_2 = 1$ and the first $i-1$ components of $u_i$ are zero. Let $P = P_b \cdot P_{b-1} \cdots P_1$, where $P_i = I - 2u_i u_i^\top$ is a Householder transformation. Show that there is a $b$-by-$b$ lower triangular matrix $T$ such that $P = I - UTU^\top$ , where $U = [u_1, \ldots, u_b]$. In particular, provide an algorithm for computing the entries of $T$ . This identity shows that we can replace multiplication by $b$ Householder transformations $P_1$ through $P_b$ by three matrix multiplications by $U$, $T$ , and $U^\top$(plus the cost of computing $T$).

2. Let House($x$) be a function of the vector $x$ which returns a unit vector $u$ such that $(I - 2uu^\top)x = \|x\|_2 e_1$; we showed how to implement House($x$) in section 3.4. Then Algorithm 3.2 for computing the QR decomposition of the $m$-by-$n$ matrix $A$ may be written as

$$\begin{aligned}
&\text{for } i = 1 : m \\
&u_i = \text{House}(A(i : m, i)) \\
&P_i = I - 2u_i u_i^\top \\
&A(i : m, i : n) = P_i A(i : m, i : n) \\
&\text{endfor}
\end{aligned}$$

   Show how to implement this in terms of the Level 2 BLAS in an efficient way (in particular, matrix-vector multiplications and rank-1 updates). What is the floating point operation count? (Just the high-order terms in n and m are enough.) It is sufficient to write a short program in the same notation as above (although trying it in Matlab and comparing with Matlab's own QR factorization are a good way to make sure that you are right!).

3. Using the results of step (1), show how to implement QR decomposition in terms of Level 3 BLAS. What is the operation count? This technique is used to accelerate the QR decomposition, just as we accelerated Gaussian elimination in section 2.6. It is used in the LAPACK routine *sgeqrf*

   proof:

### 3.18

It is often of interest to solve constrained least squares problems, where the solution $x$ must satisfy a linear or nonlinear constraint in addition to minimizing $\|Ax - b\|_2$. We consider one such problem here. Suppose that we want to choose $x$ to minimize in addition to minimizing $\|Ax - b\|_2$. We consider one such problem subject to the linear constraint $Cx = d$. Suppose also that $A$ is $m$-by-$n$, $C$ is $p$-by-$n$, and $C$ has full rank. We also assume that $p \leq n$ (so $Cx = d$ is guaranteed to be consistent) and $n \leq m + p$ (so the system is not underdetermined). Show that there is a unique solution under the assumption that $\begin{pmatrix} A \\ C \end{pmatrix}$ has full column rank. Show how to compute $x$ using two QR decompositions and some matrix-vector multiplications and solving some triangular systems of equations. Hint: Look at LAPACK routine sgglse and its description in the LAPACK manual [10] (NETLIB/lapack/lug/lapack lug.html)

proof:

### 3.19

Write a program (in Matlab or any other language) to update a geodetic database using least squares, as described in Example 3.3. Take as input a set of "landmarks," their approximate coordinates $(x_i, y_i)$, and a set of new angle measurements $\theta_j$ and distance measurements $L_{ij}$ . The output should be corrections $(\delta x_i, \delta y_i)$ for each landmark, an error bound for the corrections, and a picture (triangulation) of the old and new landmarks.

### 3.20

Prove Theorem 3.4.

Suppose that $A$ is $m$-by-$n$ with $m \geq n$ and has full rank. Suppose that $x$ minimizes $\|Ax - b\|_2$. Let $r = b - Ax$ be the residual. Let $\tilde{x}$ minimize $\|(A + \delta A)\tilde{x} - (b + \delta b)\|_2$. Assume

$$\epsilon := \max\left(\frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta b\|_2}{\|b\|_2}\right) < \frac{1}{\kappa_2(A)} = \frac{\sigma_{\min}(A)}{\sigma_{\min}(A)}.$$

Then

$$\frac{\|\tilde{x} - x\|_2}{\|x\|_2} \leq \epsilon \cdot \left\{\frac{2 \cdot \kappa_2(A)}{\cos\theta} + \tan\theta \cdot \kappa_2^2(A)\right\} + O(\epsilon^2) \equiv \epsilon \cdot \kappa_{LS} + O(\epsilon^2).$$

where $\sin\theta = \frac{\|r\|_2}{\|b\|_2}$. In other words, $\theta$ is the angle between the vectors $b$ and $Ax$ and measures whether the residual norm $\|r\|_2$ is large(near$\|b\|$) or small(near 0). $\kappa_{LS}$ is the condition number for the least squares problem.

proof:

### 3.21

Redo Example 3.1, using a rank-deficient least squares technique from section 3.5.1. Does this improve the accuracy of the high-degree approximating polynomials?

proof:

## 4

### 4.1

Let $A$ be defined as in equation (4.1). Show that $\det(A) = \prod_{i=1}^{b} \det(A_{ii})$ and then that $\det(A - \lambda I) = \prod_{i=1}^{b} \det(A_{ii} - \lambda I)$. Conclude that the set of eigenvalues of $A$ is the union of the sets of eigenvalues of $A_{11}$ through $A_{bb}$.

proof:

### 4.2

Suppose that $A$ is normal; i.e.,$AA^\dagger = A^\dagger A$. Show that if $A$ is also triangular, it must be diagonal. Use this to show that an $n$-by-$n$ matrix is normal if and only if it has $n$ orthonormal eigenvectors. Hint: Show that $A$ is normal if and only if its Schur form is normal.

proof:

### 4.3

Let $\lambda$ and $\mu$ be distinct eigenvalues of $A$, let $x$ be a right eigenvector for $\lambda$, and let $y$ be a left eigenvecctor for $\mu$. Show that $x$ and $y$ are orthogonal.

proof:

### 4.4

Suppose $A$ has distinct eigenvalues. Let

$$f(z) = \sum_{i=-\infty}^{+\infty} a_i z^i$$

be a function which is defined at the eigenvalues of $A$. Let

$$Q^\dagger A Q = T$$

be the Schur form of $A$(so $Q$ is unitary and $T$ upper triangular).

1. Show that $f(A) = Qf(T)Q^\dagger$. Thus to compute $f(A)$ it suffices to be able to compute $f(T)$. In the rest of the problem you will derive asimple recurrence formula for $f(T)$.
2. Show that $(f(T))_{ii} = f(T_{ii})$ so that the diagonal of $f(T)$ can be computedfrom the diagonal of $T$.
3. Show that $Tf(T) = f(T)T$.
4. From the last result, show that theith superdiagonal of $f(T)$ can be computed from the $(i-1)$-st and earlier subdiagonals. Thus, startingat the diagonal of $f(T)$, we can compute the first superdiagonal, second super diagonal, and so on.

proof:

### 4.5

Let $A$ be a square matrix. Apply either Question 4.4 to the Schur form of $A$ or equation (4.6) to the Jordan form of $A$ to conclude that the eigenvalues of $f(A)$ are $f(\lambda_i)$, where the $\lambda_i$ are the eigenvalues of $A$. This result is called the spectral mapping theorem. This question is used in the proof of Theorem 6.5 and section 6.5.6.

proof:

### 4.6

In this problem we will show how to solve the Sylvester or Lyapunov equation $AX - XB = C$, where $X$ and $C$ are $m$-by-$n$, $A$ is $m$-by-$m$, and $B$ is $n$-by-$n$. This is a system of $mn$ linear equations for the entries of $X$.

1. Given the Schur decompositions of $A$ and $B$, show how $AX - XB = C$ can be transformed into a similar system $A'Y - YB' = C'$, where $A'$ and $B'$ are upper triangular.

2. Show how to solve for the entries of $Y$ one at a time by a process analogous to back substitution. What condition on the eigenvalues of $A$ and $B$ guarantees that the system of equations is nonsingular?

3. Show how to transform $Y$ to get the solution $X$.

proof:

### 4.7

Suppose that $T = \begin{pmatrix} A & C \\ 0 & B \end{pmatrix}$ is in Schur form. We want to find a matrix $S$ so that $S^{-1}TS = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$. It turns out we can choose $S$ of the form $\begin{pmatrix} I & R \\ 0 & I \end{pmatrix}$. Show how to solve for $R$.

proof:

### 4.8

Let $A$ be $m$-by-$n$ and $B$ be $n$-by-$m$. Show that the matrices

$$\begin{pmatrix} AB & 0 \\ B & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0 \\ B & BA \end{pmatrix}$$

are similar. Conclude that the nonzero eigenvalues of $AB$ are the same as those of $BA$.

proof:

### 4.9

Let $A$ be $n$-by-$n$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. Show that

$$\sum_{i=1}^{n} |\lambda_i|^2 = \min_{\det(S)=0} \left\| S^{-1}AS \right\|_F^2.$$

proof:

### 4.10

Let $A$ be $n$-by-$n$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$.

1. Show that $A$ can be written $A = H + S$, where $H = H^\dagger$ is Hermitian and $S = -S^\dagger$ is skew-Hermitian. Give explicit formulas for $H$ and $S$ in terms of $A$.

2. Show that $\sum_{i=1}^{n} |\mathcal{R}\lambda_i|^2 \leq \|H\|_F^2$.

3. Show that $\sum_{i=1}^{n} |\mathcal{I}\lambda_i|^2 \leq \|S\|_F^2$.

4. Show that $A$ is normal iff $\sum_{i=1}^{n} |\lambda_i|^2 = \|A\|_F^2$.

proof:

### 4.11

Let $\lambda$ be a simple eigenvalue, and let $x$ and $y$ be right and left eigenvectors. We define the spectral projection $P$ corresponding to $\lambda$ as $P = \frac{xy^\dagger}{y^\dagger x}$. Prove that $P$ has the following properties.

1. $P$ is uniquely defined, even though we could use any nonzero scalar multiples of $x$ and $y$ in its definition.
2. $P^2 = P$. (Any matrix satisfying $P^2 = P$ is called a projection matrix.)
3. $AP = PA = \lambda P$ . (These properties motivate the name spectral projection, since $P$ "contains" the left and right invariant subspaces of $\lambda$.)
4. $\|P\|_2$ is the condition number of $\lambda$.

proof:

### 4.12

Let

$$A = \begin{pmatrix} c & c \\ 0 & b \end{pmatrix}.$$

Show that the condition numbers of the eigenvalues of $A$ are both equal to

$$(1 + (\frac{c}{a-b})^2)1/2.$$

Thus, the condition number is large if the difference $a-b$ between the eigenvalues is small compared to $c$, the offdiagonal part of the matrix.

proof:

### 4.13

Let $A$ be a matrix, $x$ be a unit vector, $\mu$ be a scalar, and $r = Ax - \mu x$. Show that there is a matrix $E$ with $\|E\|_F = \|r\|_2$ such that $A + E$ has eigenvalue $\mu$ and eigenvector $x$.

proof:

### 4.14 Programming,omitted

### 4.15 Programming,omitted

## 5

### 5.1

Show that $A = B + iC$ is Hermitian if and only if

$$M = \begin{pmatrix} B & -C \\ C & B \end{pmatrix}$$

is symmetric. Express the eigenvalues and eigenvectors of $M$ in terms of those of $A$.

proof:

### 5.2

Prove Corollary 5.1, using Weyl's theorem (Theorem 5.1) and part 4 of Theorem 3.3.

proof:

### 5.3

Consider Figure 5.1. Consider the corresponding contour plot for an arbitrary 3-by-3 matrix $A$ with eigenvalues $\alpha_3 \leq \alpha_2 \leq \alpha_1$. Let $C_1$ and $C_2$ be the two great circles along which $\rho(u, A) = \alpha_2$. At what angle do they intersect?

proof:

### 5.4

Use the Courant-Fischer minimax theorem (Theorem 5.2) to prove the Cauchy interlace theorem:

1. Suppose that $A = \begin{pmatrix} H & b \\ b^\top & u \end{pmatrix}$ is an $n$-by-$n$ symmetric matrix and $H$ is $(n-1)$-by-$(n-1)$. Let $\alpha_n \leq \cdots \leq \alpha_1$ be the eigenvalues of $A$ and $\theta_{n-1} \leq \cdots \leq \theta_1$ be the eigenvalues of H. Show that these two sets of eigenvalues interlace:

$$\alpha_n \leq \theta_{n-1} \leq \cdots \leq \theta_i \leq \alpha_i \leq \theta_{i-1} \leq \alpha_{i-1} \leq \cdots \leq \theta_1 \leq \alpha_1.$$

2. Let $A = \begin{pmatrix} H & b \\ b^\top & u \end{pmatrix}$ be $n$-by-$n$ and $H$ be $m$-by-$n$, with eigenvalues $\theta_m \leq \cdots \leq \theta_1$. Show that the eigenvalues of $A$ and $H$ interlace in the sense that $\alpha_j + (n-m) \leq \theta_j \leq \alpha_j$ (or equivalently $\alpha_j \leq \theta_{j-(n-m)} \leq \alpha_{j-(n-m)}$).

proof:

### 5.5

Let $A = A^\top$ with eigenvalues $\alpha_1 \leq \cdots \leq \alpha_n$. Let $H = H^\top$ with eigenvalues $\theta_1 \leq \cdots \leq \theta_n$. Let $A + H$ have eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. Use the Courant-Fischer minimax theorem (Theorem 5.2) to show that $\alpha_j + \theta_n \leq \lambda_j \leq \alpha_j + \theta_1$. If $H$ is positive definite, conclude that $\lambda_j > \alpha_j$ . In other words, adding a symmetric positive definite matrix $H$ to another symmetric matrix $A$ can only increase its eigenvalues.

This result will be used in the proof of Theorem 7.1.

proof:

### 5.6

Let $A = \begin{pmatrix} A_1 & A_2 \end{pmatrix}$ be $n$-by-$n$, where $A_1$ is $n$-by-$m$ and $A_2$ is $n$-by-$n-m$. Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the singular values of $A\tau_1 \geq \cdots \geq \tau_m$ be the singular values of $A_1$. Use the Cauchy interlace theorem from Question 5.4 and part 4 of Theorem 3.3 to prove that $\sigma_j \geq \tau_j \geq \sigma_{j+n-m}$.

proof:

### 5.7

Let $q$ be a unit vector and $d$ be any vector orthogonal to $q$. Show that $\left\|(q+d)q^\top - I\right\|_2 = \|q+d\|_2$. (This result is used in the proof of Theorem 5.4.)

proof:

### 5.8

Formulate and prove a theorem for singular vectors analogous to Theorem 5.4.

proof:

### 5.9

Prove bound (5.6) from Theorem 5.5.

proof:

### 5.10

Prove bound (5.7) from Theorem 5.5.

proof:

### 5.11

Suppose $\theta = \theta_1 + \theta_2$, where all three angles lie between 0 and $\frac{\pi}{2}$. Prove that

$$\frac{\sin 2\theta}{2} \leq \frac{\sin 2\theta_1 + \sin 2\theta_2}{2}.$$

This result is used in the proof of Theorem 5.7.

proof:

### 5.12

Prove Corollary 5.2. Hint: Use part 4 of Theorem 3.3.

proof:

### 5.13

Let $A$ be a symmetric matrix. Consider running shifted QR iteration (Algorithm 4.5) with a Rayleigh quotient shift ($\sigma_i = a_{nn}$) at every iteration, yielding a sequence $\sigma_1, \sigma_2, \ldots$ of shifts. Also run Rayleigh quotient iteration (Algorithm 5.1), starting with $x_0 = [0, \ldots, 0, 1]^\top$ , yielding a sequence of Rayleigh quotients $\rho_1, \rho_2, \ldots$. Show that these sequences are identical: $\sigma_i = \rho_i$ for all $i$. This justifies the claim in section 5.3.2 that shifted QR iteration enjoys local cubic convergence.

proof:

### 5.14

Prove Lemma 5.1.

proof:

### 5.15

Prove that if

$$t(n) = 2t(\frac{n}{2}) + cn^3 + O(n^2),$$

then

$$t(n) \approx c\frac{4}{3}n^3.$$

This justifies the complexity analysis of the divide-and-conquer algorithm (Algorithm 5.2).

proof:

### 5.16

Let $A = D + \rho uu^\top$ , where $D = \mathrm{diag}(d_1, \ldots, d_n)$ and $u = [u_1, \ldots, u_n]^\top$ . Show that if $d_i = d_{i+1}$ or $u_i = 0$, then $d_i$ is an eigenvalue of $A$. If $u_i = 0$, show that the eigenvector corresponding to $d_i$ is $e_i$, the $i$-th column of the identity matrix. Derive a similarly simple expression when $d_i = d_{i+1}$. This shows how to handle deflation in the divide-and-conquer algorithm, Algorithm 5.2.

proof:

### 5.17

Let $\psi$ and $\psi'$ be given scalars. Show how to compute scalars $c$ and $\hat{c}$ in the function definition

$$h(\lambda) = \hat{c} + \frac{c}{d - \lambda}$$

so that at $\lambda = \xi$, $h(\xi) = \psi$, and $h'(\xi) = \psi'$. This result is needed to derive the secular equation solver in section 5.3.3.

proof:

### 5.18

Use the SVD to show that if $A$ is an $m$-by-$n$ real matrix with $m \geq n$, then there exists an $m$-by-$n$ matrix $Q$ with orthonormal columns ($Q\top Q = I$) and an $n$-by-$n$ positive semidefinite matrix $P$ such that $A = QP$. This decomposition is called the polar decomposition of $A$, because it is analogous to the polar form of a complex number $z = e^{i \arg(z)} \cdot |z|$. Show that if $A$ is nonsingular, then the polar decomposition is unique.

proof:

### 5.19

Prove Lemma 5.5.

proof:

### 5.20

Prove Lemma 5.7.

proof:

### 5.21

Prove Theorem 5.13. Also, reduce the exponent $4n - 2$ in Theorem 5.13 to $2n - 1$. Hint: In Lemma 5.7, multiply $D_1$ and divide $D_2$ by an appropriately chosen constant.

proof:

### 5.22

Prove that Algorithm 5.13 computes the SVD of $G$, assuming that $G^\top G$ converges to a diagonal matrix.

proof:

### 5.23

Let $A$ be an $n$-by-$n$ symmetric positive definite matrix with Cholesky decomposition $A = LL^T$, and let $\hat{L}$ be the Cholesky factor computed in floating point arithmetic. In this question we will bound the relative error in the (squared) singular values of $\hat{L}$ as approximations of the eigenvalues of $A$. Show that $A$ can be written as $A = DAD$, where $D = \mathrm{diag}(a_{1/2}^1, \ldots, a_{n/n}^n)$ and $a_{ii} = 1$ for all $i$. Write $L = DX$. Show that $\kappa^2(X) = \kappa(A)$. Using bound (2.16) for the backward error $\delta A$ of Cholesky $A + \delta A = \hat{L}\hat{L}^T$, show that one can write $\hat{L}^T\hat{L} = Y^TY$ where $\|Y^TY - I\|_2 \leq O(\epsilon)\kappa(A)$. Use Theorem 5.6 to conclude that the eigenvalues of $\hat{L}^T\hat{L}$ and of $L^TL$ differ relatively by at most $O(\epsilon)\kappa(A)$. Then show that this is also true of the eigenvalues of $\hat{L}^T\hat{L}$ and $L^TL$. This means that the squares of the singular values of $\hat{L}$ differ relatively from the eigenvalues of $A$ by at most $O(\epsilon)\kappa(A) = O(\epsilon)\kappa^2(L)$.

proof:

### 5.24

This question justifies the stopping criterion for one-sided Jacobi's method for the SVD (Algorithm 5.13). Let $A = G^T G$, where $G$ and $A$ are $n$-by-$n$. Suppose that $|a_{jk}| \leq \epsilon\sqrt{a_{jj}a_{kk}}$ for all $j = k$. Let $\sigma_n \leq \cdots \leq \sigma_1$ be the singular values of $G$, and $\alpha_2^2 \leq \cdots \leq \alpha_1^2$ be the sorted diagonal entries of $A$. Prove that $|\sigma_i - \alpha_i| \leq n\epsilon|\alpha_i|$ so that the $\alpha_i$ equal the singular values to high relative accuracy. Hint: Use Corollary 5.2.

proof:

### 5.25

In Question 4.15, you "noticed" that running QR for $m$ steps on a symmetric matrix, "flipping" the rows and columns, running for another $m$ steps, and flipping again got you back to the original matrix. (Flipping $X$ means replacing $X$ by $JXJ$, where $J$ is the identity matrix with its row in reverse order.) In this exercise we will prove this for symmetric positive definite matrices $T$ using an approach different from Corollary 5.4.

Consider $LR$ iteration (Algorithm 5.9) with a zero shift, applied to the symmetric positive definite matrix $T$ (which is not necessarily tridiagonal): Let $T = B_0^T B_0$ be the Cholesky decomposition, $T_1 = B_0 B_0^T = B_1^T B_1$, and more generally $T_i = B_{i-1} B_{i-1}^T = B_i^T B_i$. Let $\hat{T}_i$ denote the matrix obtained from $T_0$ after $i$ steps of unshifted QR iteration; i.e., if $\hat{T}_i = Q_i R_i$ is the QR decomposition, then $\hat{T}_{i+1} = R_i Q_i$. In Lemma 5.6 we showed that $\hat{T}_i = T_2$; i.e., one step of QR is the same as two steps of LR.

1. Show that $T_i = (B_{i-1}B_{i-2}\cdots B_0)^{-T} T_0 (B_{i-1}B_{i-2}\cdots B_0)^T$.
2. Show that $T_i = (B_{i-1}B_{i-2}\cdots B_0) T_0 (B_{i-1}B_{i-2}\cdots B_0)^{-1}$.
3. Show that $\hat{T}_0 = (B_i B_{i-1}\cdots B_0)^T (B_i B_{i-1}\cdots B_0)$ is the Cholesky decomposition of $\hat{T}_0$.
4. Show that $\hat{T}_0^i = (Q_0 \cdots Q_{i-2}Q_{i-1}) \cdot (R_{i-1}R_{i-2}\cdots R_0)$ is the QR decomposition of $\hat{T}_0^i$.
5. Show that $\hat{T}_0^{2i} = (R_{2i-1}R_{2i-2}\cdots R_0)^T (R_{2i-1}R_{2i-2}\cdots R_0)$ is the Cholesky decomposition of $\hat{T}_0^{2i}$.
6. Show the result after $m$ steps of QR, flipping $m$ steps of QR, and flipping, is the same as the original matrix. Hint: Use the fact that the Cholesky factorization is unique.

proof:

### 5.26

Suppose that $x$ is an $n$-vector. Define the matrix $C$ by $c_{ij} = |x_i| + |x_j| - |x_i - x_j|$. Show that $C(x)$ is positive semidefinite.

proof:

### 5.27

Let

$$A = \begin{pmatrix} I & B^H \\ B & I \end{pmatrix}$$

with $\|B\|_2 < 1$. Show that

$$\|A\|_2 \|A^{-1}\|_2 = \frac{1 + \|B\|_2}{1 - \|B\|_2}.$$

proof:

### 5.28

A square matrix $A$ is said to be *skew Hermitian* if $A^* = -A$. Prove that

1. the eigenvalues of a skew Hermitian are purely imaginary.
2. $I - A$ is nonsingular.
3. $C = (I - A)^{-1}(I + A)$ is unitary. $C$ is called the Cayley transform of $A$.

proof:

For this example, we can describe the structure of the actual $\delta A$ as follows: $|\delta a_{ij}| \leq \epsilon |a_{ij}|$, where $\epsilon$ is a tiny number. We write this more succinctly as

$$|\delta A| \leq \epsilon |A| \tag{2.6}$$

(see section 1.1 for notation). We also say that $\delta A$ is a *small componentwise relative perturbation in A*. Since $\delta A$ can often be made to satisfy bound (2.6) in practice, along with $|\delta b| \leq \epsilon |b|$ (see section 2.5.1), we will derive perturbation theory using these bounds on $\delta A$ and $\delta b$.

We begin with equation (2.1):

$$\delta x = A^{-1}(-\delta A \hat{x} + \delta b).$$

Now take absolute values, and repeatedly use the triangle inequality to get

$$\begin{aligned} |\delta x| &= |A^{-1}(-\delta A \hat{x} + \delta b)| \\ &\leq |A^{-1}|(|\delta A| \cdot |\hat{x}| + |\delta b|) \\ &\leq |A^{-1}|(\epsilon |A| \cdot |\hat{x}| + \epsilon |b|) \\ &= \epsilon(|A^{-1}|(|A| \cdot |\hat{x}| + |b|)). \end{aligned}$$

Now using any vector norm (like the infinity-, one-, or Frobenius norms), where $\| \, |z| \, \| = \|z\|$, we get the bound

$$\|\delta x\| \leq \epsilon \| |A^{-1}|(|A| \cdot |\hat{x}| + |b|)\|. \tag{2.7}$$

Assuming for the moment that $\delta b = 0$, we can weaken this bound to

$$\|\delta x\| \leq \epsilon \| |A^{-1}| \cdot |A| \| \cdot \|\hat{x}\|$$

or

$$\frac{\|\delta x\|}{\|x\|} \leq \epsilon \| \, |A^{-1}| \cdot |A| \, \|. \tag{2.8}$$

This leads us to define $\kappa_{CR}(A) \equiv \| \, |A^{-1}| \cdot |A| \, \|$ as the *componentwise relative condition number of A*, or just *relative condition number* for short. It is sometimes also called the Bauer condition number [26] or Skeel condition number [223, 224, 225]. For a proof that bounds (2.7) and (2.8) are attainable, see Question 2.4.

图 1: question 2.6.