STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2019
LECTURE 9

1. BACK SUBSTITUTION AND TRIDIAGONAL SOLVE

- backsolve or back substitution refers to a simple, intuitive way of solving linear systems of the form $R\mathbf{x} = \mathbf{b}$ or $L\mathbf{x} = \mathbf{b}$ where R is upper-triangular and L is lower-triangular
- take $R\mathbf{x} = \mathbf{b}$ for illustration

$$\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

- start at the bottom and work our way up

$$\begin{aligned} b_n &= r_{nn}x_n \\ b_{n-1} &= r_{n-1,n}x_n + r_{n-1,n-1}x_{n-1} \\ &\vdots \\ b_1 &= r_{11}x_1 + r_{12}x_2 + \cdots + r_{1n}x_n \end{aligned}$$

- we get

$$\begin{aligned} x_n &= \frac{b_n}{r_{nn}} \\ x_{n-1} &= \frac{b_{n-1} - r_{n-1,n}(b_n/r_n)}{r_{n-1,n-1}} \\ &\vdots \end{aligned}$$

- this requires that $r_{kk} \neq 0$ for all $k = 1, \dots, n$, which is guaranteed if R is nonsingular
- for example we could use QR factorization
- given $A \in \mathbb{C}^{n\times n}$ nonsingular and $\mathbf{b} \in \mathbb{C}^n$
  - step 1: find QR factorization $A = QR$
  - step 2: form $\mathbf{b} = Q^*\mathbf{b}$
  - step 3: backsolve $R\mathbf{x} = \mathbf{y}$ to get $\mathbf{x}$
- it is easy to solve $A\mathbf{x} = \mathbf{b}$ if
  - A is unitary or orthogonal (includes permutation matrices)
  - A is upper- or lower-triangular (includes diagonal matrices)
  - $A\mathbf{x} = \mathbf{b}$ with such A can be solved with $O(n^2)$ flops
  - if A represents a special orthogonal matrix like the discrete Fourier or wavelet transforms, then $A\mathbf{x} = \mathbf{b}$ can in fact be solved in $O(n \log n)$ flops using algorithms like fast Fourier or fast wavelet transforms
- if A is not one of these forms, we factorize A into a product of matrices of these forms
- this may be viewed as the basic impetus for matrix factorizations like LU, Cholesky, QR, SVD, EVD
- actually to the above list, we could also add

Date: October 23, 2019, version 1.0. Comments, bug reports: `lekheng@galton.uchicago.edu`.

- A is bidiagonal/tridiagonal (or banded, i.e., $a_{ij} = 0$ if $|i-j| > b$ for some *bandwidth* $b \ll n$)
- A is Toeplitz or Hankel, i.e., $a_{ij} = a_{i-j}$ or $a_{ij} = a_{i+j}$ — constant on the diagonals or the opposite diagonals
- A is semiseparable
- $A\mathbf{x} = \mathbf{b}$ with bidiagonal or tridiagonal A can be solved in $O(n)$ flops
- $A\mathbf{x} = \mathbf{b}$ with Toeplitz or Hankel A can be solved in $O(n^2 \log n)$ flops
- these are often called structured matrices

- for example, a tridiagonal system

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}$$

may be solved by first computing

$$c_i' = \begin{cases} \dfrac{c_i}{b_i} & i = 1, \\ \dfrac{c_i}{b_i - a_i c_{i-1}'} & i = 2, 3, \dots, n-1, \end{cases}$$

and

$$d_i' = \begin{cases} \dfrac{d_i}{b_i} & i = 1, \\ \dfrac{d_i - a_i d_{i-1}'}{b_i - a_i c_{i-1}'} & i = 2, 3, \dots, n, \end{cases}$$

followed by back substitution

$$\begin{aligned} x_n &= d_n', \\ x_i &= d_i' - c_i' x_{i+1}, \qquad i = n-1, n-2, \dots, 1 \end{aligned}$$

- in this course we will just restrict ourselves to unitary and triangular factors
- but we will discuss a general principle for solving linear systems and least squares problems based on rank-retaining factorizations that works with any structured matrices

2. Rank-retaining factorizations

- let $A \in \mathbb{C}^{m \times n}$ with $\operatorname{rank}(A) = r$, a *rank-retaining factorization* is a factorization of A into

$$A = GH$$

where $G \in \mathbb{C}^{m \times r}$ and $H \in \mathbb{C}^{r \times n}$ and

$$\operatorname{rank}(G) = \operatorname{rank}(H) = r$$

  - example: condensed SVD $A = U\Sigma V^*$, $U \in \mathbb{C}^{m \times r}$, $\Sigma \in \mathbb{C}^{r \times r}$, $V \in \mathbb{C}^{n \times r}$ where we could pick $G = U\Sigma$ and $H = V^*$ or $G = U$ and $H = \Sigma V^*$
  - example: condensed QR $A\Pi = QR$, $Q \in \mathbb{C}^{m \times r}$, $R \in \mathbb{C}^{r \times n}$, where we could pick $G = Q$ and $H = R\Pi^{\mathsf{T}}$
  - example: condensed LU $\Pi_1 A \Pi_2 = LU$, $L \in \mathbb{C}^{m \times r}$, $U \in \mathbb{C}^{r \times n}$, where we could pick $G = \Pi_1^{\mathsf{T}} L$ and $H = U\Pi_2^{\mathsf{T}}$
- easy facts: if $A = GH$ is rank-retaining, then
  (i) $G^*G \in \mathbb{C}^{r \times r}$ is nonsingular

(ii) $HH^* \in \mathbb{C}^{r\times r}$ is nonsingular
(iii) $\operatorname{im}(A) = \operatorname{im}(G)$
(iv) $\ker(A^*) = \ker(G^*)$
(v) $\ker(A) = \ker(H)$
(vi) $\operatorname{im}(A^*) = \operatorname{im}(H^*)$

- prove these as exercises

3. General principle for linear systems and least squares

- we will discuss a general principle for solving linear systems and least squares problems via matrix factorization
- given $A \in \mathbb{C}^{m\times n}$ and $\mathbf{b} \in \mathbb{C}^m$, two of the most common problems are
  - if $A\mathbf{x} = \mathbf{b}$ is consistent and A is full column rank, we want the unique solution
  - if $A\mathbf{x} = \mathbf{b}$ is inconsistent and A is full column rank, we want the unique least squares solution
- the trouble is that when A is rank deficient, i.e., not full rank, then the solution is not unique and so we want the minimum length solution instead
  - if $A\mathbf{x} = \mathbf{b}$ is consistent and A is rank deficient, we want the minimum length solution
$$\min\{\|\mathbf{x}\|_2 : A\mathbf{x} = \mathbf{b}\} \tag{3.1}$$
  - if $A\mathbf{x} = \mathbf{b}$ is inconsistent and A is rank deficient, we want the minimum length least squares solution
$$\min\{\|\mathbf{x}\|_2 : \mathbf{x} \in \operatorname{argmin}\|\mathbf{b} - A\mathbf{x}\|_2\} \tag{3.2}$$
- if we can solve the min length versions then we can solve the full column rank versions, so let's focus on the min length version

4. Min length linear systems via rank-retaining factorization

- we start from the consistent case: $\mathbf{b} \in \operatorname{im}(A)$ and so $\mathbf{b} = A\mathbf{x}$ for some $\mathbf{x} \in \mathbb{C}^n$
  - recall the Fredholm alternative that we proved in the homework:
$$\mathbb{C}^n = \operatorname{im}(A^*) \oplus \ker(A)$$
  - $\mathbf{x} \in \mathbb{C}^n$ can be written uniquely as
$$\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1, \quad \mathbf{x}_0 \in \ker(A),\ \mathbf{x}_1 \in \operatorname{im}(A^*),\ \mathbf{x}_0^*\mathbf{x}_1 = 0$$
  - since
$$\mathbf{b} = A\mathbf{x} = A\mathbf{x}_0 + A\mathbf{x}_1 = A\mathbf{x}_1$$
    $\mathbf{x}_1$ is also a solution to the linear system
  - by Pythagoras theorem
$$\|\mathbf{x}\|_2^2 = \|\mathbf{x}_0\|_2^2 + \|\mathbf{x}_1\|_2^2 \geq \|\mathbf{x}_1\|_2^2$$
  - so for a minimum length solution we set $\mathbf{x}_0 = \mathbf{0}$, i.e., the minimum length solution is given by $\mathbf{x} = \mathbf{x}_1$
- now we will see how to find $\mathbf{x}_1$ using a rank-retaining factorization
$$A = GH \tag{4.1}$$
  - since $\mathbf{x}_1 \in \operatorname{im}(A^*) = \operatorname{im}(H^*)$ by easy fact (vi), so for some $\mathbf{v} \in \mathbb{C}^r$,
$$\mathbf{x}_1 = H^*\mathbf{v} \tag{4.2}$$
  - by easy fact (iii), $\mathbf{b} \in \operatorname{im}(A) = \operatorname{im}(G)$ and so for some $\mathbf{s} \in \mathbb{C}^r$,
$$\mathbf{b} = G\mathbf{s} \tag{4.3}$$

  - so upon substituting (4.1), (4.2), (4.3), $A\mathbf{x}_1 = \mathbf{b}$ becomes
$$GHH^*\mathbf{v} = G\mathbf{s}$$
  - now multiply by G^* to get
$$(G^*G)HH^*\mathbf{v} = (G^*G)\mathbf{s}$$
  - by easy fact (i), G^*G is nonsingular and so
$$HH^*\mathbf{v} = \mathbf{s}$$
  - by easy fact (ii), HH^* is nonsingular and so
$$\mathbf{v} = (HH^*)^{-1}\mathbf{s}$$
  - plugging back into (4.2), we get
$$\mathbf{x}_1 = H^*(HH^*)^{-1}\mathbf{s} \tag{4.4}$$
- this gives an algorithm for solving the minimum length linear system (3.1)
  - step 1: compute rank retaining factorization $A = GH$
  - step 2: solve $G\mathbf{s} = \mathbf{b}$ for $\mathbf{s} \in \mathbb{C}^r$
  - step 3: solve $HH^*\mathbf{z} = \mathbf{s}$ for $\mathbf{z} \in \mathbb{C}^r$
  - step 4: compute $\mathbf{x}_1 = H^*\mathbf{z}$
- this works because
$$A\mathbf{x}_1 = GH\mathbf{x}_1 = GHH^*\mathbf{z} = G(HH^*)(HH^*)^{-1}\mathbf{s} = G\mathbf{s} = \mathbf{b}$$
- note that the system in steps 2 and 3 involve a full-rank G and a nonsingular HH^* — both have unique solutions
- example: if $A\Pi = QR$ is the condensed QR, then with $G = Q$ and $H = R\Pi^{\mathsf{T}}$
  - step 2: $Q\mathbf{s} = \mathbf{b}$ is easy to obtain via
$$Q^*Q\mathbf{s} = Q^*\mathbf{b}$$
    and so $\mathbf{s} = Q^*\mathbf{b}$
  - step 3: $R\Pi^{\mathsf{T}}\Pi R^*\mathbf{z} = \mathbf{s}$ is also easy to obtain via two backsolves
$$\begin{cases} R\mathbf{y} = \mathbf{s} \\ R^*\mathbf{z} = \mathbf{y} \end{cases}$$
- example: if $A = U\Sigma V^*$ is the condensed SVD, then with $G = U$ and $H = \Sigma V^*$
  - step 2: $U\mathbf{s} = \mathbf{b}$ is easy to obtain via
$$U^*U\mathbf{s} = U^*\mathbf{b}$$
    and so $\mathbf{s} = U^*\mathbf{b}$
  - step 3: $\Sigma V^* V\Sigma\mathbf{z} = \mathbf{s}$ is just
$$\Sigma^2\mathbf{z} = \mathbf{s}$$
    or
$$\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_r^2 \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_r \end{bmatrix} = \begin{bmatrix} s_1 \\ \vdots \\ s_r \end{bmatrix}$$
    and so for $k = 1, \dots, r$,
$$z_k = s_k/\sigma_k^2$$
- note that (4.4) is in terms of $\mathbf{s}$, if we want an analytic expression, it should involve only quantities we know, i.e., $\mathbf{b}, G, H$

- to express $\mathbf{s}$ in terms of quantities we know, we just multiply (4.3) by G^* to get
$$G^*G\mathbf{s} = G^*\mathbf{b}$$
and using fact (i) to get
$$\mathbf{s} = (G^*G)^{-1}G^*\mathbf{b}$$
- with this and (4.4), we get an analytic expression for the minimum length solution
$$\mathbf{x}_1 = H^*(HH^*)^{-1}(G^*G)^{-1}G^*\mathbf{b} \tag{4.5}$$

5. MIN LENGTH LEAST SQUARES VIA RANK-RETAINING FACTORIZATION

- we now consider the inconsistent case: $\mathbf{b} \notin \operatorname{im}(A)$
  - this time we use the other part of the Fredholm alternative:
$$\mathbb{C}^m = \ker(A^*) \oplus \operatorname{im}(A)$$
  - any $\mathbf{b} \in \mathbb{C}^m$ can be written uniquely as
$$\mathbf{b} = \mathbf{b}_0 + \mathbf{b}_1, \quad \mathbf{b}_0 \in \ker(A^*),\ \mathbf{b}_1 \in \operatorname{im}(A),\ \mathbf{b}_0^*\mathbf{b}_1 = 0$$
  - since $\mathbf{b}_1 - A\mathbf{x} \in \operatorname{im}(A)$, it must also be orthogonal to $\mathbf{b}_0$ and by Pythagoras
$$\|\mathbf{b} - A\mathbf{x}\|_2^2 = \|\mathbf{b}_0 + \mathbf{b}_1 - A\mathbf{x}\|_2^2 = \|\mathbf{b}_0\|_2^2 + \|\mathbf{b}_1 - A\mathbf{x}\|_2^2 \geq \|\mathbf{b}_0\|_2^2$$
  - so for a least squares solution, we must have
$$\|\mathbf{b}_1 - A\mathbf{x}\|_2^2 = 0$$
i.e.,
$$A\mathbf{x} = \mathbf{b}_1 \tag{5.1}$$
  - this is always consistent since $\mathbf{b}_1 \in \operatorname{im}(A)$ and we proceed as in the consistent case to get from (4.5),
$$\mathbf{x}_1 = H^*(HH^*)^{-1}(G^*G)^{-1}G^*\mathbf{b}_1 \tag{5.2}$$
  - but by easy fact (iv), $\ker(A^*) = \ker(G^*)$ and so
$$G^*\mathbf{b} = G^*(\mathbf{b}_0 + \mathbf{b}_1) = G^*\mathbf{b}_0 + G^*\mathbf{b}_1 = G^*\mathbf{b}_1 \tag{5.3}$$
  - in other words, the $\mathbf{b}_1$ in (5.2) may be replaced by $\mathbf{b}$ and we get
$$\mathbf{x}_1 = H^*(HH^*)^{-1}(G^*G)^{-1}G^*\mathbf{b} \tag{5.4}$$
- note that there is no difference in the expression (4.5) for minimum length linear system and the expression (5.4) for minimum length least squares
- following the previous section, we can write down an algorithm using (5.4) to get the minimum length solution to a least squares problem (3.2) (exercise)
- a consequence of (5.4) is that given a rank-retaining factorization $A = GH$, the Moore–Penrose pseudoinverse of A is given by
$$A^\dagger = H^*(HH^*)^{-1}(G^*G)^{-1}G^* \tag{5.5}$$
- example: if $A = U\Sigma V^*$ is the condensed SVD, then $A^\dagger = V\Sigma^{-1}U^*$ since (5.5) with $G = U$ and $H = \Sigma V^*$ yields
$$A^\dagger = V\Sigma(\Sigma V^*V\Sigma)^{-1}(U^*U)^{-1}U^* = V\Sigma\Sigma^{-2}U^* = V\Sigma^{-1}U^*$$
- example: if $A\Pi = QR$ is the condensed QR, then $A^\dagger = \Pi R^*(RR^*)^{-1}Q^*$ since (5.5) with $G = Q$ and $H = R\Pi^\mathsf{T}$ yields
$$A^\dagger = \Pi R^*(R\Pi^\mathsf{T}\Pi R^*)^{-1}(Q^*Q)^{-1}Q^* = \Pi R^*(RR^*)^{-1}Q^*$$

6. OTHER USES OF QR

- the QR decomposition for a square matrix may be used to determine the magnitude of determinant
$$|\det(A)| = |\det(QR)| = |\det(Q)||\det(R)| = |\det(R)| = \prod_{k=1}^{n} |r_{kk}|$$
- we used two facts: determinant of unitary matrix must have absolute value 1, determinant of triangular (upper or lower) matrix is just product of diagonal elements
- the rank-retaining QR decomposition may be used to determine orthonormal bases for the fundamental subspaces
$$A\Pi = [Q_1, Q_2] \begin{bmatrix} R_1 & S \\ 0 & 0 \end{bmatrix}$$
- the columns of Q_1 form an orthonormal basis for $\operatorname{im}(A)$ (follows from Gram–Schmidt) and the columns of Q_2 form an orthonormal basis for $\ker(A^*)$
- if we need orthonormal bases for $\operatorname{im}(A^*)$ and $\ker(A)$, we find the rank-retaining QR factorization of A^*
- this is a cheaper way than SVD to obtain orthonormal bases for the fundamental subsapces

7. FULL RANK LEAST SQUARES PROBLEM

- the general method for a rank-retaining factorization works for matrices of any rank but there are better alternatives to solve least squares problem when the coefficient matrix A has full column rank
- here we seek to minimize $\|A\mathbf{x} - \mathbf{b}\|_2$ where $A \in \mathbb{C}^{m\times n}$ has $\operatorname{rank}(A) = n \le m$ and $\mathbf{b} \in \mathbb{C}^m$
- such problems *always* have unique solution $\mathbf{x}^*$ (why?)
- so there is no question of finding a min length solution — since there's only one solution in this case, we don't get to choose
- we consider three methods:
  (1) QR factorization
  (2) normal equations
  (3) augmented system
- mathematically they all give the same solution (i.e., in exact arithmetic) but they have different numerical properties
- so one has to know all three since each is good/bad under different circumstances

8. FULL RANK LEAST SQUARES VIA QR

- the first approach is to take advantage of the fact that the 2-norm is invariant under orthogonal transformations, and seek an orthogonal matrix Q such that the transformed problem
$$\min \|A\mathbf{x} - \mathbf{b}\|_2 = \min \|Q^*(A\mathbf{x} - \mathbf{b})\|_2$$
is "easy" to solve
- we could use the QR factorization of A
$$A = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

- then $Q_1^* A = R$ and
$$\begin{aligned}\min\|A\mathbf{x}-\mathbf{b}\|_2 &= \min\|Q^*(A\mathbf{x}-\mathbf{b})\|_2\\ &= \min\|(Q^*A)\mathbf{x}-Q^*\mathbf{b}\|_2\\ &= \min\left\|\begin{bmatrix}R\\0\end{bmatrix}\mathbf{x}-Q^*\mathbf{b}\right\|_2\end{aligned}$$
- if we partition
$$Q^*\mathbf{b}=\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}$$
then
$$\min\|A\mathbf{x}-\mathbf{b}\|_2^2=\min\left\|\begin{bmatrix}R\\0\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\right\|_2^2=\min\|R\mathbf{x}-\mathbf{c}\|_2^2+\|\mathbf{d}\|_2^2$$
- therefore the minimum is achieved by the vector $\mathbf{x}$ such that $R\mathbf{x}=\mathbf{c}$ and therefore
$$\min_{\mathbf{x}\in\mathbb{C}^n}\|A\mathbf{x}-\mathbf{b}\|_2=\|\mathbf{d}\|_2$$

9. FULL RANK LEAST SQUARES VIA NORMAL EQUATIONS

- the second approach is to define
$$\varphi(\mathbf{x})=\frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2$$
which is a differentiable function of $\mathbf{x}$
- we can minimize $\varphi(\mathbf{x})$ by noting that $\nabla\varphi(\mathbf{x})=A^*(A\mathbf{x}-\mathbf{b})$ which means that $\nabla\varphi(\mathbf{x})=\mathbf{0}$ if and only if
$$A^*A\mathbf{x}=A^*\mathbf{b} \tag{9.1}$$
- this system of equations is called the *normal equations*, and were used by Gauss to solve least squares problems
- we saw at least two other ways to derive (9.1) in the homeworks
- it is generally a bad idea to solve the normal equations to get the least squares solution, although this is not always the case
- for example, if $n \ll m$ then A^*A is $n\times n$, which is a much smaller system to solve than solving $\min\|A\mathbf{x}-\mathbf{b}\|_2^2$ via finding QR of A, and if $\kappa(A^*A)$ is not too large, we can indeed solve (9.1)
- for A of full column rank, the matrix A^*A is positive definite and one should apply Cholesky factorization (to be discussed later) to the matrix A^*A in order to solve (9.1)
- which is the better method?
- this is not a simple question to answer
- the normal equations produce an $\mathbf{x}^*$ whose relative error depends on $\kappa_2(A^{\mathsf{T}}A)=\kappa_2(A)^2$, whereas the QR factorization produces an $\mathbf{x}^*$ whose relative error depends on $\kappa_2(A)+\rho_{\mathrm{LS}}(\mathbf{x}^*)\kappa_2(A)^2$ where
$$\rho_{\mathrm{LS}}(\mathbf{x}):=\frac{\|\mathbf{b}-A\mathbf{x}\|_2}{\|A\|_2\|\mathbf{x}\|_2}$$
is called the *relative residual* at $\mathbf{x}$
- so the QR factorization method in the previous section is appealing if $\rho_{\mathrm{LS}}(\mathbf{x}^*)$ is small, i.e., $\mathbf{b}$ is very close to $\mathrm{im}(A)$, the span of the columns of A — which is more often than not the case (e.g. in linear regression) as the most common reason for wanting to solve $\min\|A\mathbf{x}-\mathbf{b}\|_2$ is when we expect $A\mathbf{x}^*\approx\mathbf{b}$
- the normal equations involve much less arithmetic when $n \ll m$ and the $n\times n$ matrix A^*A requires less storage

10. Full rank least squares via augmented system

- we can cast the normal equation in another form
- let $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ be the residual
- now by the normal equations
$$A^*\mathbf{r} = A^*\mathbf{b} - A^*A\mathbf{x} = \mathbf{0}$$
- and so we obtain the system
$$\begin{aligned} \mathbf{r} + A\mathbf{x} &= \mathbf{b} \\ A^*\mathbf{r} &= \mathbf{0} \end{aligned}$$
- in matrix form, we get
$$\begin{bmatrix} I & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$
- this is often a large system since the coefficient matrix has dimension $(m+n)\times(m+n)$, but it preserves the sparsity of A

11. QR factorization versus normal equations

- assuming a dense A, the following table compares the relative merits of normal equations (NE) method, QR method, and the SVD method discussed a few lectures ago

accuracy:	NE	<	QR	<	SVD
speed:	NE	>	QR	>	SVD

11.1. Conditioning of least squares.

Theorem 1 (Wedin). *Let $A, \widehat{A} \in \mathbb{R}^{m\times n}$ where $\operatorname{rank}(A) = \operatorname{rank}(\widehat{A}) = n \le m$. Suppose $\mathbf{x}$ and $\widehat{\mathbf{x}} \in \mathbb{R}^n$ are solutions to the respective least squares problems*
$$\min\|A\mathbf{x} - \mathbf{b}\|_2 \quad \text{and} \quad \min\|\widehat{A}\widehat{\mathbf{x}} - \widehat{\mathbf{b}}\|_2,$$
and let $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ and $\widehat{\mathbf{r}} = \widehat{\mathbf{b}} - \widehat{A}\widehat{\mathbf{x}}$ be the respective residuals. If $\epsilon > 0$ is such that
$$\frac{\|A - \widehat{A}\|_2}{\|A\|_2} \le \epsilon, \quad \frac{\|\mathbf{b} - \widehat{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \le \epsilon, \quad \kappa_2(A)\epsilon < 1,$$
then
$$\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \le \frac{\kappa_2(A)\epsilon}{1 - \kappa_2(A)\epsilon}\left(2 + (\kappa_2(A) + 1)\frac{\|\mathbf{r}\|_2}{\|A\|_2\|\mathbf{x}\|_2}\right), \tag{11.1}$$
and
$$\frac{\|\mathbf{r} - \widehat{\mathbf{r}}\|_2}{\|\mathbf{r}\|_2} \le 1 + 2\kappa_2(A)\epsilon.$$

- recall that for singular or rectangular matrices, $\kappa_2(A) = \|A\|_2\|A^\dagger\|_2$
- note that if $\mathbf{r} = \mathbf{0}$, i.e., the least squares problem becomes a linear system, (11.1) reduces to the bound we obtained in Homework **2**, Problem **6**(e)
- in other words, for a linear system, the term involving $\kappa_2(A)^2$ vanishes
- a simplification of (11.1) is to assume that $\widehat{\mathbf{b}} = \mathbf{b}$ and get
$$\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \le \frac{\kappa_2(A)\epsilon}{1 - \kappa_2(A)\epsilon}\left(1 + \kappa_2(A)\frac{\|\mathbf{r}\|_2}{\|A\|_2\|\mathbf{x}\|_2}\right)$$
- if we expand the right hand side, we get
$$\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \le \kappa_2(A)\left(1 + \kappa_2(A)\frac{\|\mathbf{r}\|_2}{\|A\|_2\|\mathbf{x}\|_2}\right)\epsilon + O(\epsilon^2) \tag{11.2}$$
- the coefficient of ϵ above is sometimes called the *least squares condition number*

11.2. **Accuracy.**

- the QR method, if properly implemented (say, using Householder or Givens algorithm that we will discuss next time), is backward stable in the following sense: when we use the method to solve
$$\min\|A\mathbf{x} - \mathbf{b}\|_2,$$
we get the *exact* solution to a perturbed problem
$$\min\|\widehat{A}\widehat{\mathbf{x}} - \widehat{\mathbf{b}}\|_2,$$
that is near to our original problem in the sense that
$$\frac{\|A - \widehat{A}\|_2}{\|A\|_2} \le \epsilon, \quad \frac{\|\mathbf{b} - \widehat{\mathbf{b}}\|_2}{\|\mathbf{b}\|_2} \le \epsilon$$
for some small ϵ
- in practice, the value of ϵ depends on m, n and the unit roundoff u of the computer/program[1] you use and is typically very small, roughly $mn\mathsf{u}/(1 - mn\mathsf{u})$
- this, combined with Theorem 1 allows us to get a bound on the relative error (as long as $\kappa_2(A) < 1/\epsilon$)
- if we use (11.2), we see that the relative error is bounded by $(\kappa_2(A) + \rho_{\text{LS}}(\mathbf{x}^*)\kappa_2(A)^2)\epsilon$
- so if $\rho_{\text{LS}}(\mathbf{x}^*)$ is small, then QR is good for accuracy
- the normal equations method, given that it relies on solving $A^{\mathsf{T}}A\mathbf{x} = A^{\mathsf{T}}\mathbf{b}$, cannot avoid the condition number $\kappa_2(A^{\mathsf{T}}A) = \kappa_2(A)^2$ no matter which version of Homework **1**, Problem **4** we use
- the relative error in this case is therefore always bounded by $\kappa_2(A)^2\epsilon$, never just $\kappa_2(A)\epsilon$
- as long as A is well-conditioned, it is alright to use the normal equations method, especially if you want to save on computational cost, the QR method is generally preferable
- for very ill-conditioned problem, one would have to use the SVD method discussed a few lectures ago but this is the most expensive

11.3. **Computational costs.**

- assuming that our matrix $A \in \mathbb{R}^{m\times n}$ is dense (most or all entries nonzero), then the exact flop counts for the two methods described earlier for computing the least squares solution $\mathbf{x}$ are:
  - QR factorization ($A = Q\left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right]$) + orthogonal transformation ($\mathbf{c} = Q^{\mathsf{T}}\mathbf{b}$) + backsolve ($R\mathbf{x} = \mathbf{c}$):
  $$2n^2\left(m - \frac{n}{3}\right) \tag{11.3}$$
  - normal equations ($C = A^{\mathsf{T}}A$, $\mathbf{c} = A^{\mathsf{T}}\mathbf{b}$) + Cholesky factorization ($C = R^{\mathsf{T}}R$) + two backsolves ($R^{\mathsf{T}}\mathbf{y} = \mathbf{c}$, $R\mathbf{x} = \mathbf{y}$):
  $$n^2\left(m + \frac{n}{3}\right) \tag{11.4}$$
- so both methods have similar computation cost if $m \approx n$ but the normal equations method is up to twice as fast for $m \gg n$
- the flop count in (11.3) assumes that we do Householder QR (discussed later) since the matrix is dense
- the flop count in (11.4) assumes that we do Cholesky factorization (discussed later)

[1]The unit roundoff $\mathsf{u} = \varepsilon_{\text{machine}}/2$ and is around 10^{-16} for double precision, 10^{-19} for extended precision, 10^{-35} for quadruple precision.

11.4. **Roundoff errors.**

- another issue with the normal equations is the loss of information when we roundoff
- for example, if
$$A = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \end{bmatrix}, \qquad A^{\mathsf{T}}A = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1 \end{bmatrix},$$
and ϵ is so small that your computer rounds off $1+\epsilon^2$ to 1, then you end up with a rank-deficient matrix
$$\mathrm{fl}(A^{\mathsf{T}}A) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$
- note that for the QR method, we work directly with A and do not need to form $A^{\mathsf{T}}A$ so we don't face this problem
- statisticians often use the normal equations because in many statistical problems, the measurement errors in A are much larger than the roundoff errors and so the latter type of errors are relatively insignificant