MATH 578 Numerical Analysis, Fall 2020
Student Notes

Kai Yang
kai.yang2@mail.mcgill.ca
McGill University
Montréal, Quebec Canada H3A 1A2

October 17, 2020

Abstract

This is just my notes going through the highlights from the lecture notes for MATH 578 Numerical Analysis, Fall 2020 (Tsogtgerel, 2020). As a computational statistician muggle taking this course for optimization and machine learning, this notes might not always be a good summary, I must say... As an overview, the first chapter covers up some basics of analysis functions – specifically, Lagrange interpolation (Taylor series expansion is a special case) and minimax polynomials; the second chapter covers up linear system part.

I Function Evaluation

1 Basic Computer Arithmetic $\forall a \in \mathbb{Z}$, a *base–β representation* exists for some $\beta \in \mathbb{N} \setminus \{1\}$:

$$a = \pm \sum_{k=0}^{\infty} a_k \beta^k$$

where $0 \leq a_k \leq \beta - 1$ is defined as the k–th digit of a in base β. And grade-school column sum/difference first carries out *Cauchy sum* or *difference*, which takes sum/difference for each digits; then it recursively perform carrying for addition for borrowing for subtraction. Let

$$n := \max\{k | a_k \neq 0\},\ m := \max\{k | b_k \neq 0\}$$

So a, b will be $n+1$ and $m+1$ digit number. The bit complexity for addition/subtraction will then be $O(n+m+1)$. Column multiplication carries out similarly. However, multiplication can also be done row-wisely: the *Cauchy product*

$$ab = \left(\sum_{i=0}^{\infty} a_i \beta^i\right) \cdot b = \sum_{i=0}^{n} a_i \cdot \beta^i b$$

where $\beta^j b$ is simply shifting digits, and multiplication by a_i can be carried out as column addition. The bit complexity for column multiplication would then be $O(nm+1)$.

As for division algorithm, assume that the quotient is expressed as:

$$q = q_0 + q_{-1}\beta^{-1} + q_{-2}\beta^{-2} + \cdots$$

And let a, b here be positive and normalized. The *partial reminder* refers to the *normalized* reminder obtained in the division process. Two division algorithms for a/b were introduced here: i). *restoring division*: keeping performing subtraction see if the partial reminder goes below 0, and if it goes below 0, "restore" by adding the divisor back to it to prevent negative digits; ii). *non-restoring division*: the idea of non-restoring division is to use generalized digit, e.g. $\{-1, 1\}$ for binary computing, to allow negative sign in a digit, and a conversion back to standard digit will be indeed required *in the end*. To generalize non-restoring division to any radix β, note that the partial reminders are given by:

$$r_{j+1} = \beta r_j - q_{-j} b$$

the above two division processes determine q_{-j} both by subtracting b from βr_j, the difference is for restoring division, $0 \le q_{-j} < \beta$ gives partial reminder $0 \le r_{j+1} < b$; for non-restoring division, $-\beta < q_j < \beta$ gives partial reminder $-b \le r_{j+1} < b$.

However, both of above division algorithms are not efficient – especially not for bignums. WLOG, let a, b be integers here, the idea of *long division* is to determine the quotient by observing the first digit of the divisor and perform restoring division. In comparison, *SRT division* is non-restoring division with normalized divisor and reminder. *Error propagation* describes the idea of computation will alternate (mostly increase) the error of approximation numbers, such as floating point numbers. Usually error propagation is captured upper-boundedly by *conditional number*, e.g. conditional number of summation is

$$\kappa_+(x) = \frac{|x_1| + |x_2| + \cdots + |x_n|}{|x_1 + x_2 + \cdots + x_n|}$$

Furthermore, the following axiom is used for a wide-range of numerical error analysis for floating point numbers: For each $\star \in \{+, -, \times, /\}$, there exists a binary operation $\circledast : \tilde{\mathbb{R}} \times \tilde{\mathbb{R}} \mapsto \tilde{\mathbb{R}}$ s.t.

$$\left|x \star y - x \circledast y\right| \le \varepsilon \left|x \star y\right|, \; x, y \in \tilde{\mathbb{R}}$$

dividing by zero is excluded. Normally, ε is referred as "machine precision."

2 Evaluation of Power Series A function $f : (a,b) \mapsto \mathbb{R}$ is called *analytic* at $c \in (a,b)$ if it can be developable into a power series around c; and called analytic at (a,b) if analytic at c, $\forall c \in (a,b)$. For such class of analytic functions, a way to evaluate them is through Taylor series, backed by a generalized version of mean value theorem proposed by Lagrange: Let f be a $n+1$ times differentiable function in (c,x), with $f^{(n)}$ continuous in $[c,x)$. Then $\exists \xi \in (c,x)$ s.t.

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x-c)^k + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-c)^{n+1}$$

See Lagrange interpolation section coming later for proof. This theorem gives an expression of the error as a result of approximating using n–th order Taylor series. Moreover, the following series are listed with their relative condition numbers:

$$\begin{aligned}
\frac{1}{1-x} &= \sum_{k=0}^{\infty} x^k,\ \forall |x| < 1 \\
\kappa(x) &= \left|\frac{(1-x)^{-2}}{(1-x)^{-1}/x}\right| = \left|\frac{x}{1-x}\right| \\
e^x &= 1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} + \cdots,\ \forall x \in \mathbb{R} \\
\kappa(x) &= \left|\frac{(e^x)'}{e^x/x}\right| = |x| \\
\log(1+x) &= \sum_{k=1}^{\infty} \frac{(-1)^{n-1} x^n}{k},\ -1 < x \leq 1 \\
\kappa(x) &= \left|\frac{(1+x)^{-1}}{\log(1+x)/x}\right| = \frac{x}{(1+x)} \cdot \left|\frac{1}{\log(1+x)}\right| \\
\sin x &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!},\ \forall x \in \mathbb{R} \\
\kappa(x) &= \left|\frac{\cos x}{\sin x/x}\right| = |x \cot x| \\
\cos x &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!},\ \forall x \in \mathbb{R} \\
\kappa(x) &= \left|\frac{-\sin x}{\cos x/x}\right| = |x \tan x| \\
\arctan x &= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{2n+1},\ |x| \leq 1
\end{aligned}$$

$$\arcsin x = x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \cdots,\ -1 \le x < 1$$

And recall that the relative condition numbers is defined by:

$$\kappa := \lim_{\varepsilon \downarrow 0} \sup_{\|\delta x\| \le \varepsilon} \frac{\|\delta f\| / \|f(x)\|}{\|\delta x\| / \|x\|}$$

3 Acceleration of Convergence Two methods of acceleration of convergence are discussed here:

i). *Euler transform*: *Hausdorff moment characterization* says that

$$\begin{aligned} m_k = \int_0^1 x^k d\mu \text{ for some } \sigma-\text{additive Borel probability measure } \mu \\ \Leftrightarrow m_0 = 1,\ m \text{ is completely monotone; i.e. } (-1)^n \Delta^n m_k \ge 0,\ \forall n, k \end{aligned}$$

The formula

$$\frac{\pi}{4} = \sum_{k=0}^{n} (-1)^k \frac{1}{2k+1}$$

can be accelerated by repeatedly taking average of two consecutive terms, called *Euler transform*. Applying Hausdorff moment characterization, error analysis for this can be done by noticing that

$$a_k = \frac{1}{k} = \int_0^1 t^k d\mu$$

and the rest follows from power series.

ii). *Aitken's Δ^2–process*: used to evaluate a noisy geometric series. For a series defined by

$$\begin{aligned} a_k &= Cq^k + O\left(\delta^k\right), \text{ for some } 0 < \delta < q < 1 \\ S_n &= \sum_{k=1}^{n} a_k \end{aligned}$$

Observe that

$$\begin{aligned} S &= S_n + \sum_{k=n+1}^{\infty} a_k \\ &= S_n + \sum_{k=n+1}^{\infty} Cq^k + O(\delta^n) \end{aligned}$$

$$
\begin{aligned}
=&S_n + \frac{Cq^{n+1}}{1-q} + O(\delta^n) \\
=&S_n + \frac{a_n^2}{a_{n-1}-a_n} + O(\delta^n)
\end{aligned}
$$

The last inequality above used the fact that

$$
\begin{aligned}
q =& \frac{a_n}{a_{n-1}} + O\left(\left(\frac{\delta}{q}\right)^n\right) \\
a_n =& Cq^n + O(\delta^n) \\
\Rightarrow & Cq^{n+1} = \left(\frac{a_n}{a_{n-1}} + O\left(\left(\frac{\delta}{q}\right)^n\right)\right)(a_n - O(\delta^n)) = \frac{a_n^2}{a_{n-1}} + O(\delta^n), \text{ and} \\
& \frac{1}{1-q} = \frac{a_{n-1}}{a_{n-1}-a_n} + O\left(\left(\frac{\delta}{q}\right)^n\right)
\end{aligned}
$$

Let

$$
\begin{aligned}
\Delta S_{n-1} :=& S_n - S_{n-1} = a_n \\
\Delta^2 S_{n-2} :=& a_n - a_{n-1} = \Delta a_{n-1}
\end{aligned}
$$

We then have

$$
S_n + \frac{a_n^2}{a_{n-1}-a_n} = S_n - \frac{(\Delta S_{n-1})^2}{\Delta^2 S_{n-2}}
$$

which gives the name "Δ^2"

4 Root Finding Fixed point iterations are based on a theorem: Let $\phi : (a,b) \mapsto (a,b)$ be continuous. Further, let $x_{k+1} = \phi(x_k)$, $x_0 \in (a,b)$, and

$$
\forall x, y \in (a,b),\ \exists \rho < 1 \text{ s.t. } \left|\phi(x) - \phi(y)\right| \le \rho \left|x - y\right|
$$

moreover, assume that $\exists \alpha \in (a,b)$ s.t. $\phi(\alpha) = \alpha$. Then $\forall x_0 \in (a,b)$, $x_n \to \alpha$ as $n \to \infty$ (linear convergence). Note that possible underlying connection to Lipschitz continuity here. And recall that optimization can be more or less considered as a root finding procedure of the first-order optimality condition. The examples given here are chord method (corresponding to gradient descent), and Newton-Raphson method (local quadratic convergence).

5 Lagrange Interpolation The problem *Lagrange Interpolation* aims to solve is, given $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, find coefficients $a_0, \ldots, a_n$ for $p \in \mathbb{P}_n$ s.t.

$$p(x) := \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

One way to get the coefficients for the polynomial is to use *Lagrange coefficients*:

$$\phi_k(x) := \prod_{i=0,\ i\neq k}^{n} \frac{x - x_i}{x_k - x_i}$$

and

$$p(x) = \sum_{k=0}^{n} y_k \phi_k(x)$$

as we can observe that

$$\phi_j(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Now define *Lagrange interpolation* as a map $\mathcal{L}_n : \mathcal{C}(a,b) \mapsto \mathbb{P}_n$, where $\{x_0, \ldots, x_n\} \subset (a,b)$ are distinct and fixed; i.e., to take $n+1$ points on f and construct the Lagrange polynomial passing through these $n+1$ points. Note that $\mathcal{L}_n$ is a projection, i.e. $\mathcal{L}_n\mathcal{L}_n = \mathcal{L}_n$. Recall we have seen how Lagrange generalized mean value theorem to higher-orders for Taylor series before, and here is the origin of Lagrange Theorem:

Let f be $n+1$th order differentiable in (a,b), and $x \in (a,b)$. Then $\exists \xi = \xi(x)$ s.t.

$$\min\{x_0, \ldots, x_n, x\} < \xi < \max\{x_0, \ldots, x_n, x\}, \text{ and} \tag{1}$$

$$f(x) - (\mathcal{L}_n f)(x) = \frac{(x - x_0)\cdots(x - x_n)}{(n+1)!} f^{(n+1)}(\xi)$$

The idea of proof is to construct the *Lagrange reminder*:

$$R(x) := f(x) - (\mathcal{L}_n f)(x);\ A := \frac{R(x)}{\prod_{i=0}^{n}(x - x_i)}$$

then the function

$$F(z) := f(z) - (\mathcal{L}_n f)(x) - A \prod_{i=0}^{n} (z - x_i)$$

has $n+2$ distinct zeros $\{x_0,\ldots,x_n,x\}$; $F'(z)$ has $n+1$ distinct zeros; ...; $F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - A(n+1)! = 0$ for some ξ in the convex hull as described in (1). This implies

$$f(x) - (\mathcal{L}_n f)(x) = R(x) = A\prod_{i=0}^{n}(x-x_i) = \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!}f^{(n+1)}(\xi)$$

Note here how Rolle's theorem can be used to bridge the gap between higher-order in the last of the proof. Another interesting thing is that, Taylor's series expansion can be considered as Lagrange interpolation with repeated x_j.

6 Runge's Phenomenon *Runge's phenomenon* refers to the phenomenon that for a typical analytic function, equispaced Lagrange interpolation tends to oscillate more towards the boundary – that is, it tends to interpolate better in the middle. A typical analytic function will have $f^{(n)}(x) \sim \frac{n!}{\delta^n}$, and will have error$\sim \frac{\pi(x)}{\delta^n}$ for $\pi(x) = (x-x_0)\cdots(x-x_n)$. This suggests that *high-order polynomials on equispaced grid is not a good idea*, rather, it's a better idea to pick more points around the edge. Alternatively, it might be a better idea to approximate a function not by interpolating at certain points, but rather to minimize the upper bound of the approximation error norm – which leads to the discussion of the following three sections.

7 Weierstrass Approximation Theorem The *Weierstrass Approximation Theorem* states that a polynomial is dense in the space of continuous function in uniform norm: Let $f \in \mathcal{C}[a,b]$ and $\varepsilon > 0$; then $\exists n \in \mathbb{N}$, $\exists q \in \mathbb{P}_n(x)$ s.t.

$$\max_{x\in[a,b]} |f(x) - q(x)| \le \varepsilon$$

Bernstein proposed a constructive proof back in 1904. WLOG, $[a,b] = [0,1]$. Define *Bernstein polynomials* to have coefficients

$$\beta_{n,j}(x) = \binom{n}{j} x^j (1-x)^{n-j},\ j = 0,\ldots,n$$

i.e., binomial polynomial if you study stats... It has a few simple properties:

1. $\beta_{n,j}(x) > 0,\ \forall x \in (0,1)$
2. $\sum_{j=0}^{n} \beta_{n,j}(x) = 1$
3. $\sum_{j=0}^{n} \frac{j}{n}\beta_{n,j}(x) = x$
4. $\sum_{j=0}^{n} \frac{j^2}{n^2}\beta_{n,j}(x) = \left(1 - \frac{1}{n}\right)x^2 + \frac{1}{n}x$

And the interpolation proceeds as: let $x_j = \frac{j}{n}$, $j = 0, 1, \ldots, n$, and let $B_n f(x) = \sum_{j=0}^{n} f\left(x_j\right) \beta_{n,j}(x)$. Observe that

$$\begin{aligned} f(x) - B_n f(x) &= f(x) \sum_{j=0}^{n} \beta_{n,j}(x) - \sum_{j=0}^{n} f\left(x_j\right) \beta_{n,j}(x) \\ &= \sum_{j=0}^{n} \left[f(x) - f\left(x_j\right) \right] \beta_{n,j}(x) \end{aligned}$$

Now split the function into two components:

$$\begin{aligned} R_\delta(x) &:= \left| \sum_{\left|x - x_j\right| \le \delta} \left[f(x) - f\left(x_j\right) \right] \beta_{n,j}(x) \right| \\ &\le \underbrace{\left| \sum_{j=0}^{n} \beta_{n,j}(x) \right|}_{=1} \cdot \underbrace{\max_{y \in [0,1],\ \left|x - y\right| \le \delta} \left| f(x) - f(y) \right|}_{=:\omega(\delta)} \\ S_\delta(x) &:= \left| \sum_{\left|x - x_j\right| > \delta} \left[f(x) - f\left(x_j\right) \right] \beta_{n,j}(x) \right| \end{aligned}$$

and construct interpolation sequence of $\xi_1, \xi_2, \ldots, \xi_p$ between x and x_j s.t. the distance (in Euclidean norm) between two neighbor points $\le \delta$, then

$$\begin{aligned} \left| f(x) - f\left(x_j\right) \right| &\le \left| f(x) - f(\xi_1) \right| + \left| f(\xi_1) - f(\xi_2) \right| + \cdots + \left| f\left(\xi_p\right) - f\left(x_j\right) \right| \\ &\le (p+1)\, \omega(\delta) \\ &\le \left(1 + \frac{\left|x - x_j\right|}{\delta} \right) \omega(\delta) \end{aligned}$$

This further implies that

$$\left|S_\delta(x)\right| \le \underbrace{\sum_{\left|x - x_j\right| > \delta} \omega(\delta)\, \beta_{n,j}(x)}_{\le \omega(\delta)} + \frac{\omega(\delta)}{\delta} \underbrace{\sum_{\left|x - x_j\right| > \delta} \left| x - x_j \right| \beta_{n,j}(x)}_{=:A} \le \left(1 + \frac{1}{4\delta^2 n} \right) \omega(\delta)$$

where above inequality uses the fact that

$$\begin{aligned}\delta A \leq & \sum_{|x-x_j|>\delta} \left(x-x_j\right)^2 \beta_{n,j}(x) \\ \leq & \sum_{j=0}^{n} \left(x-x_j\right)^2 \beta_{n,j}(x) \\ = & x^2 \sum_{j=0}^{n} \beta_{n,j}(x) - 2x \sum_{j=0}^{n} \frac{j}{n} \beta_{n,j}(x) + \sum_{j=0}^{n} \frac{j^2}{n^2} \beta_{n,j}(x) \\ = & x^2 - 2x^2 + \left(1-\frac{1}{n}\right)x^2 + \frac{1}{n}x \\ = & \frac{x(1-x)}{n} \\ \leq & \frac{1}{4n}\end{aligned}$$

Hence,

$$|f(x) - B_n f(x)| \leq \left(2 + \frac{1}{4n\delta^2}\right)\omega(\delta),\ \forall \delta > 0 \text{ and } x \in [0,1]$$

Pick $\delta = \frac{1}{\sqrt{n}}$ completes the proof.

8 Minimax polynomials The *minimax polynomial* refers to the polynomial of a given degree that minimizes the uniform norm of the error for a continuous function on a closed interval, and its existence is ensured by the following theorem: Let $f \in \mathcal{C}[0,1]$ and $n \in \mathbb{N}_0$. Then $\exists p \in \mathbb{P}_n$ s.t.

$$\|f - p\|_\infty = \inf_{q \in \mathbb{P}_n} \|f - q\|_\infty$$

such q is called a *minimax polynomial* of degree n for f (on $[0,1]$).

The proof follows from continuous function achieves minimizer over a compact set (Weierstrass Theorem): For the sake of simplicity, let $a \in \mathbb{R}^{n+1}$ denote the coefficient vector for a nth order polynomial q, and

$$E(a) := \|f - q\|_\infty = \max_{x \in [0,1]} |f(x) - q(x)|$$

First we are to prove the continuity of E:

$$\begin{aligned}|E(a + \delta a)| \leq & \left| \|f - q - \delta q\|_\infty - \|f - q\|_\infty \right| \\ \leq & \|\delta q\|_\infty \\ \leq & |\delta a_0| + \cdots + |\delta a_n|\end{aligned}$$

Now let $K := \left\{a \in \mathbb{R}^{n+1} | E(a) \leq \|f\|_\infty + 1\right\}$. Then:

1. K is closed, because $K = E^{-1}([0, \|f\|_\infty + 1])$ (pre-image of a closed set under continuous mapping is closed)
2. K is bounded, because $\|q\|_\infty \leq \underbrace{\|f - q\|_\infty}_{=:E(a)} + \|f\|_\infty$ and

$$\|a\| \leq \text{constant} \cdot \|q\|_\infty \Rightarrow E(a) \to \infty \text{ as } \|a\| \to \infty$$

3. Nonempty, because $0 \in K$

Thus, by Weierstrass Theorem, $\exists a^* \in K$ s.t. $E(a^*) = \inf_{a \in K} E(a)$ – but we still have to prove that $E(a^*) = \inf_{a \in \mathbb{R}^{n+1}} E(a)$:

$$E(a^*) \leq E(0) = \|f\|_\infty \leq \|f\|_\infty + 1 < E(a),\ \forall a \in \mathbb{R}^{n+1} \setminus K$$

9 Equioscillation Theorems Two important theorems are given to characterize minimax polynomials.

The first one is *De la Vallee Poussin Theorem*: $\forall f \in \mathcal{C}[a,b],\ n \in \mathbb{N}_0,\ p \in \mathbb{P}_n$, if

$$f\left(x_j\right) - p\left(x_j\right) = (-1)^j e_j,\ \forall j = 0, 1, \ldots, n+1$$

where $a_0 \leq x_0 < x_1 < \cdots < x_{n+1} \leq b$, and $\operatorname{sgn} e_j =$ constant for $j = 0, 1, \ldots n+1$; then[1]

$$E_n(f) := \min_{q \in \mathbb{P}_n} \|f - q\|_\infty \geq \min_j \left|e_j\right|$$

The proof is by contradiction: assume that the conclusion is false, then

$$\begin{aligned} p\left(x_j\right) - q\left(x_j\right) &= (-1)^j e_j + \underbrace{f\left(x_j\right) - q\left(x_j\right)}_{<\left|e_j\right|,\ \forall j = 0,1,\ldots,n+1} \\ &\Rightarrow p - q \text{ has } n+1 \text{ (distinct) zeros} \\ &\Rightarrow p \equiv q \end{aligned}$$

but it contradicts our assumption on p and q

The second one is *Chebyshev's Oscillation Theorem*, which characterizes the minimax polynomials: $p \in \mathbb{P}_n$ is a minimax polynomial for $f \in \mathcal{C}[0,1]$ iff $f - p$ takes the value $\pm\|f - p\|_\infty$,

[1] existence of minimax polynomial was proved in Section 8

with alternating changes of sign, at least $n+2$ times in $[0,1]$. Moreover, this minimax polynomial is unique.

For statement besides uniqueness: Proof for "$\Leftarrow$" is done by DLVP, $\|f-p\|_\infty \le E_n(f) \Rightarrow \|f-p\|_\infty = E_n(f)$ by minimality of $E_n(f)$; proof for "$\Rightarrow$" is done by contradiction: assume the conclusion is false, i.e., $f-p$ takes the value $\pm\|f-p\|_\infty$ of k times for some $2 \le k \le n+1$[2], and let $\delta := \pm\|f-p\|_\infty$; then $f(x_i)-p(x_i) = (-1)^j \delta$ for $j = 1,\dots,k$. And WLOG this allows us to (quasi-)partition $[0,1]$ into k intervals split by $\xi_1, \xi_2, \dots, \xi_{k-1}$ s.t. on

$$\begin{aligned}(0,\xi_1),(\xi_2,\xi_3),\cdots &: -\delta \le f-p \le \delta-\varepsilon\\ (\xi_1,\xi_2),(\xi_3,\xi_4),\cdots &: -\delta+\varepsilon \le f-p \le \delta\end{aligned}$$

for some $\varepsilon > 0$. Now let $r(x) = \pm(x-\xi_1)\cdots(x-\xi_{k-1})$ – we'll discuss choice of sign shortly after, and let $q(x) := p(x)-\alpha\cdot r(x)$ for some small $\alpha > 0$ s.t. $\|\alpha r\|_\infty \le \frac{\varepsilon}{2}$, then $f-q = f-p+\alpha r$. Thus on

$$\begin{aligned}(0,\xi_1),(\xi_2,\xi_3),\cdots &: -\delta < -\delta+\alpha r \le f-q \le \delta-\frac{\varepsilon}{2}\\ (\xi_1,\xi_2),(\xi_3,\xi_4),\cdots &: -\delta+\frac{\varepsilon}{2} \le f-q \le \delta+\alpha r < \delta\end{aligned}$$

and we choose the sign of $r(x)$ s.t. $r > 0$ on the first line above and $r < 0$ on the second line above. Then q actually takes strictly less error than p, which contradicts that p is the minimax polynomial.

For uniqueness statement: let p, q both be minimax polynomials, and let $r := \frac{p+q}{2}$. Then

$$\begin{aligned}&|f-r| \le \frac{1}{2}|f-p| + \frac{1}{2}|f-q| \le E_n(f)\\ \Rightarrow &|f-r| = E_n(f) \text{ at } n+2 \text{ distinct points}\\ \Rightarrow &f-p = f-q = \pm E_n(f) \text{ at those points – because } f-p = -(f-q) \Rightarrow f-r = 0\\ \Rightarrow &p = q \text{ at } n+2 \text{ distinct points}\\ \Rightarrow &p \equiv q\end{aligned}$$

10 Chebyshev Polynomials Recall that the Runge's phenomenon suggests that the equispaced interpolation of polynomials does not approximate the function well, then we aims to position the interpolation points over a non-equal grid to approximate the function better. For example, we are to find the minimax polynomial in $\mathbb{P}_n$ for $f(x) = x^{n+1}$. Recall that sin, cos usually brings oscillations, but they are not polynomials, then

[2] $k \ge 2$ because it's a minimax polynomial

Chebyshev introduced a polynomial variant from it:

$$t_n(x) := \cos(n \arccos x)$$

which will gives us $t_n(x) = 1$, $t_1(x) = x$. Recall that

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2\cos(n\theta)\cos\theta$$

translate this into $t_n(x)$, it is

$$t_{n+1}(x) = 2t_n(x)x - t_{n-1}(x)$$

these are called *Chebyshev polynomials*, the zeros of $t_{n+1}(x)$ satisfy $(n+1)\arccos x = \frac{\pi}{2} + k\pi$ for $k = 0, 1, \ldots, n$.

II Equation Solving

11 Gaussian Elimination The idea of *Gaussian Elimination*, is based on use upper rows to eliminate front-end matrix terms – one term at a time; and the resulting matrix will be an upper-triangular matrix. e.g.:

$$A = \underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 3 \\ 8 & 7 & 9 \end{bmatrix}}_{A_1} \to \underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 3 & 5 \end{bmatrix}}_{A_2} \to \underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}}_{A_3}$$

written in matrix form of above example, it will be

$$A_2 = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}}_{\Lambda_1} A_1, \; A_3 = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}}_{\Lambda_2} A_2$$

and note such Λ_k are always lower-triangular – in fact it only has nonzero entries at the kth column and all diagonal entries being 1, as we use upper rows to eliminate elements from lower rows.

12 LU-decomposition As a summary, in the example, $A_3 = \underbrace{\Lambda_2 \Lambda_1}_{\Lambda} A$; as a product of lower-triangular matrices, Λ is also lower-triangular, hence Λ^{-1} is also lower-triangular.

And the decomposition for full rank matrix $A = \Lambda^{-1}A_3$ is called *LU-decomposition*, in practice:

1. LU-decomposition has arithmetic complexity of roughly $\frac{1}{3}n^3$ multiplications;
2. LU-decomposition breaks down if $(A_k)_{k,k} = 0$ for some k
3. L and U can be stored in a single $n \times n$ array (because Λ^{-1} always has diagonal elements all being 1)

LU decomposition of A exists iff all principal minors of A are nonzero. If exists, LU decomposition is unique. Prove by noticing that Gaussian elimination always preserves principal minors. For uniqueness, let

$$LU = \hat{L}\hat{U} \Rightarrow \underbrace{\hat{L}^{-1}L}_{\text{lower-trig}} = \underbrace{\hat{U}U^{-1}}_{\text{upper-trig}} = I \Rightarrow \hat{U} = U,\ \hat{L} = L$$

Now the issue still remains if we encounter $(A_k)_{k,k} = 0$ for some k. To solve this issue, and also to make most prominent values (measured by Euclidean norm) up to the top to ensure numerical stability, *pivoting* is introduced. *Partial pivoting* means row interchanges (arithmetic complexity n^2); and *complete pivoting* refers to row and column interchanges (arithmetic complexity $\frac{1}{3}n^3$). An example for partial pivoting row interchange:

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} a_2 \\ a_4 \\ a_1 \\ a_3 \end{bmatrix}$$

and the *permutation matrix* P has properties that $PP^T = P^TP = I$, and the product of permutation matrix is still a permutation matrix (recall it just interchanges rows). Now partial pivoting LU decomposition performs pivoting after each elimination step, specifically,

$$\begin{aligned} U =& \Lambda_n P_n \Lambda_{n-1} P_{n-1} \cdots \Lambda_1 P_1 A \\ =& \Lambda_n \underbrace{\left(P_n \Lambda_{n-1} P_n^{-1}\right)}_{\Lambda'_{n-1}} \underbrace{\left(P_n P_{n-1} \Lambda_{n-2} P_{n-1}^{-1} P_n^{-1}\right)}_{\Lambda'_{n-2}} \cdots \underbrace{\left(P_n \cdots P_2 \Lambda_1 P_2^{-1} \cdots P_n^{-1}\right)}_{\Lambda'_1} P_n \cdots P_2 P_1 A \end{aligned}$$

where Λ_i' are *unit lower triangular matrices* – note that they are not lower triangular. And let $\Lambda' = \Lambda_n \Lambda_{n-1}' \Lambda_{n-2}' \cdots \Lambda_1'$, then $U = \Lambda' P A$, this gives

$$PA = LU$$

which is called *PLU-decomposition*. From the above pivoting process, it can be concluded that every square matrix has a PLU-decomposition.

13 Orthogonalization and QR-decomposition A matrix $Q \in \mathbb{R}^{n\times n}$ is called *orthogonal* if $Q^T Q = I$, i.e., if its column vectors form a orthonormal basis of $\mathbb{R}^n$. The idea of QR-decomposition comes from

$$Ax = QRx = b \Rightarrow QRx = b \Rightarrow Rx = Q^T b$$

then $Rx = Q^T b$ can be solved by back-substitution. If A, B are orthogonal, then AB and BA are both orthogonal. This allows us to perform QR-decomposition by a series of steps and times an orthogonal matrix at each step.

Recall that the projection of a on b is defined to be

$$\mathrm{proj}_b a := \frac{\langle a, b\rangle}{\|a\| \cdot \|b\|} \cdot \|a\| \cdot \frac{b}{\|b\|} = \frac{\langle a, b\rangle}{\langle b, b\rangle} b$$

First we'll have a look at Gram-Schmidt method: let $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$ be column vectors of $A \in \mathbb{R}^{n\times m}$, we can then form a orthonormal basis $q_1, q_2, \ldots, q_m$ by letting

$$\begin{aligned} q_1 &= \frac{a_1}{\|a_1\|}; \\ q_2' &= a_2 - \langle a_2, q_1\rangle q_1,\ q_2 = \frac{q_2'}{\left\|q_2'\right\|}; \\ &\vdots \\ q_m' &= a_m - \sum_{k=1}^{m-1} \langle a_m, q_k\rangle q_k,\ q_m = \frac{q_m'}{\|q_m'\|}. \end{aligned}$$

where $\|\cdot\|$ denote Euclidean norm. i.e., in each Gram-Schmidt step, first take off the projection of a_k onto the existing orthonormal basis s.t. the remaining vector will be orthogonal to the existing basis, then normalize a_k. Applying Gram-Schmidt to perform QR-decomposition, each Gram-Schmidt step can be considered as multiplication with a triangular matrix (i.e., step k will normalize $a_k^{(k)}$, and subtract the projections on q_k from

$a^{(k)}_{k+1}, a^{(k)}_{k+2}, \ldots, a^{(k)}_m$):

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_m \end{bmatrix} \begin{bmatrix} \frac{1}{\langle q_1, a_1 \rangle} & -\frac{\langle q_1, a_2 \rangle}{\langle q_1, a_1 \rangle} & -\frac{\langle q_1, a_3 \rangle}{\langle q_1, a_1 \rangle} & \cdots & -\frac{\langle q_1, a_m \rangle}{\langle q_1, a_1 \rangle} \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} q_1 & a^{(2)}_2 & a^{(2)}_3 & \ldots & a^{(2)}_m \end{bmatrix}$$

Or view a_k as the sum of its projections on $q_1, q_2, \ldots, q_k$, from which we formulate

$$A = \begin{bmatrix} q_1 & q_2 & \ldots & q_m \end{bmatrix} \begin{bmatrix} \langle q_1, a_1 \rangle & \langle q_1, a_2 \rangle & \cdots & \langle q_1, a_m \rangle \\ 0 & \langle q_2, a_2 \rangle & & \langle q_2, a_m \rangle \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \langle q_m, a_m \rangle \end{bmatrix} = QR$$

where q_k is obtained using Gram-Schmidt – note that this ensures all the 0s below diagonal.

14 QR-decomposition by Triangularization *Triangularization* refers to the idea of triangularizing a matrix by zeroing its below-diagonal entries. Here two methods are discussed.

The first one is *triangularization by givens rotation*: a *(clockwise) givens rotation matrix*[3] is defined by

$$G = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

And for $a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, to zero the second entry, i.e., to ensure

$$Ga = \begin{bmatrix} a_1 \cos\theta + a_2 \sin\theta \\ -a_1 \sin\theta + a_2 \cos\theta \end{bmatrix} = \begin{bmatrix} \sqrt{a_1^2 + a_2^2} \\ 0 \end{bmatrix}$$

we only need to let

$$\sin\theta = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}$$

[3] recall we have seen them in complex analysis

$$\cos\theta = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}$$

Note that givens rotation matrix is orthogonal; and generalization to zeroing the n−dimensional vector entry a_{k+1} can be simply done by taking an identity matrix and mutate $\begin{bmatrix} I_{kk} & I_{k(k+1)} \\ I_{(k+1)k} & I_{(k+1)(k+1)} \end{bmatrix}$ to be the givens rotation matrix. Then for $A \in \mathbb{R}^{n\times m}$, we can zero the entries of a_i, $\forall i = 1, 2, \dots, m$ in an order of $n, n-1, \dots, i+1$ – we have to stop at $i+1$ for a_i as further zeroing will mutate the sparse patterns for zeroed $a_1, a_2, \dots, a_{i-1}$. This way, we can obtain an upper-triangular matrix.

The second QR-decomposition method is *Householder's reflector*. Different from how givens rotations method rotates the vector to zeroing an entry, Householder's method will reflect the vector by a hyperplane H s.t. the reflection can point to the desired direction – one column vector at a time. Specifically, for $a_1 \in \mathbb{R}^n$, we try to multiply by an orthogonal matrix Q_1 s.t. $Q_1 a_1 = \|a_1\| e_1$ where e_1 having first entry being 1 and rest of entries being 0, the hyperplane H will be orthogonal to $v := \|a_1\| e_1 - a_1$, therefore

$$Q_1 = I - 2\frac{vv^T}{v^T v}$$

where Q_1 is orthogonal. In general,

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$

where $I \in \mathbb{R}^{(k-1)\times(k-1)}$, and $F \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$ s.t. $F\tilde{a}_k^{(k)} = \left\|\tilde{a}_k^{(k)}\right\| e_1$, where $\tilde{a}_k^{(k)} \in \mathbb{R}^{n-k+1}$ is $\left(\left(a_k^{(k)}\right)_k, \left(a_k^{(k)}\right)_{k+1}, \dots, \left(a_k^{(k)}\right)_n\right)$; i.e., the upper-left sub-matrix I together with the two zero sub-matrices are to preserve obtained $a_1^{(k)}, a_2^{(k)}, \dots, a_{k-1}^{(k)}$ from the first $k-1$ steps, and F is reflecting $\tilde{a}_k^{(k)}$ to obtain $a_k^{(k+1)}$ – which is to be preserved later; i.e., $a_k^{(k+1)} = a_k^{(k+2)} = \cdots = a_k^{(\min(m,n-1))}$.

15 Conditioning of $Ax = b$ Consider a linear system with numerical error:

$$\begin{aligned} &(A + \delta A)(x + \delta x) = b + \delta b \\ \Leftrightarrow &(A + \delta A)\,\delta x = \delta b - \delta A x \end{aligned}$$

From where, we would expect for all invertible A, $A+\delta A$ will also be invertible if δA is small. To solve this issue, first we'll look at *induced*[4] *matrix norm*, defined for any $A \in \mathbb{R}^{n\times m}$ by

$$\|A\| := \sup_{x\in\mathbb{R}^m} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|\le 1} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|$$

Immediately following from the definition, we have

- $\|\alpha A\| = |\alpha|\,\|A\|$ (*absolutely homogeneous*)
- $\|A+B\| \le \|A\| + \|B\|$ (*triangle-inequality*)
- $\|A\| \ge 0$ and $\|A\| = 0 \Leftrightarrow A = 0$ (*positive-definiteness*)

and the well-known *Frobenius norm*:

$$\|A\|_F := \sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n} \left|a_{ij}\right|^2} = \operatorname{tr}\left(A^T A\right) = \operatorname{tr}\left(AA^T\right)$$

which leads to the use fact that: $\exists \alpha, \beta > 0$ s.t. $\alpha \|A\|_F \le \|A\|_* \le \beta \|A\|_F$, where $\|\cdot\|_*$ denotes any induced norm.

And for matrix geometric series, we are thinking of

$$(I-K)^{-1} = I + K + K^2 + \cdots \tag{2}$$

and (2) converges iff the $\ell-2$ norm of all eigenvalues of A are strictly less than 1 – recall that $I-K$ is invertible iff 1 is not an eigenvalue of K. And specific for convergence proof, let

$$B_l := I + K + \cdots + K^l$$

then we have

$$\begin{aligned}\|B_{l+m} - B_l\| &= \left\|K^{l+1} + \cdots + K^{l+m}\right\| \\ &\le \|K\|^{l+1} + \cdots + \|K\|^{l+m} \\ &\le \frac{\|K\|^{l+1}}{1-\|K\|}\end{aligned}$$

if $\|K\| < 1$. Then $\{B_l\}$ is Cauchy, which implies that $\exists B \in \mathbb{R}^{n\times m}$ s.t. $B_l \to B$ as $l \to \infty$.

Now go back to our problem, we need $A+\delta A = A\left(I + A^{-1}\delta A\right)$ to be invertible, then we'll have $(A+\delta A)^{-1} = \left(I + A^{-1}\delta A\right)^{-1} A^{-1}$; and $\left(I + A^{-1}\delta A\right)^{-1}$ exists if $\left\|A^{-1}\delta A\right\| < 1$, note that

[4] *induced* means matrix norm induced by vector norms

$\left\|A^{-1}\delta A\right\| \leq \left\|A^{-1}\right\| \|\delta A\|$, then we only need $\|\delta A\| < \frac{1}{\|A^{-1}\|}$. The rest of error analysis follows from matrix norm properties and matrix geometric series properties[5]. Eventually, it can be derived that

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A\| \left\|A^{-1}\right\|}{1 - \left\|A^{-1}\delta A\right\|} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$$

where we define the *condition number* as

$$\kappa(A) = \|A\| \left\|A^{-1}\right\|$$

16 Backward Error Analysis For floating point addition, we can treat them *as if* input were perturbed; e.g.:

$$x_1 \oplus x_2 = (x_1 + x_2)(1 + \delta) = (1 + \delta) x_1 + (1 + \delta) x_2 =: \tilde{x}_1 + \tilde{x}_2$$

Applying this treatment to the entire algorithm, we get *backward error analysis (BEA)*. Let $\tilde{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$ be some algorithmic realization of $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, then BEA refers to the idea of model the errors committed within $\tilde{f}$ by error in the input data. The algorithm is called *stable* if $\exists \tilde{x} \approx x$ s.t. $f(\tilde{x}) \approx \tilde{f}(x)$. If it is possible to make $f(\tilde{x}) = f(x)$, then the algorithm is called *backward stable*.

[5] which is frequently used when we deal with matrix inverse

References

Tsogtgerel, Gantumur (2020). *MATH 598 Lecture Notes, Fall 2020*.

Jiao, Xiangmin (2012). *Lecture 13: Householder Reflectors;Updating QR Factorization*. URL: http://www.ams.sunysb.edu/~jiao/teaching/ams526_fall12/lectures/lecture13.pdf.