Advanced Numerical Analysis

1 1/17

Consider the function $f(x) = \sqrt[n]{x}$ where $x > 0$ and $n \in \mathbb{N}$. The Jacobian of f is the function

$$\|\mathrm{J}_f(x)\| = \frac{1}{n} x^{\frac{1}{n}-1}.$$

This is very large when x is very small. Next we have

$$\kappa_{\mathrm{rel}} = \frac{\|\mathrm{J}_f(x)\|\|x\|}{\|f(x)\|} = \frac{1}{n}.$$

d

Example 1.1. labelexample Let $f(\boldsymbol{x}) = x_1 - x_2$. The Jacobian is

$$\mathrm{J}_f(\boldsymbol{x}) = \begin{pmatrix} 1 & -1 \end{pmatrix}.$$

Then

$$\kappa_{\mathrm{abs}} = \|\mathrm{J}_f(x)\|_\infty = 2.$$

$$\kappa_{\mathrm{rel}} = \frac{\|\mathrm{J}_f(x)\|_\infty \|x\|_\infty}{\|f(x)\|_\infty} = \frac{2\max\{|x_1|, |x_2|\}}{|x_1 - x_2|}$$

Example 1.2. labelexample Let $f(\boldsymbol{x}) = A\boldsymbol{x}$ where A is an $m \times n$ matrix. Then $\mathrm{J}_f(\boldsymbol{x}) = A$ so

$$\kappa_{\mathrm{abs}} = \|\mathrm{J}_f(x)\| = \|A\|$$

and

$$\kappa_{\mathrm{rel}} = \frac{\|A\|\|x\|}{\|Ax\|} \leq 1.$$

If A is square and nonsingular, then

$$\begin{aligned}
\kappa_{\mathrm{rel}} &= \frac{\|A\|\|x\|}{\|Ax\|} \\
&= \frac{\|A\|\|A^{-1}Ax\|}{\|Ax\|} \\
&\leq \frac{\|A\|\|A^{-1}\|\|x\|}{\|Ax\|} \\
&= \|A\|\|A^{-1}\|.
\end{aligned}$$

What if the coefficient matrix has a small perturbation? How much would the true solution change?

$$\begin{cases} Ay = b & b \text{ is fixed} \\ (A + \Delta A)(y + \Delta y) = b \\ Ay + (\Delta A)y + A(\Delta y) + (\Delta A)(\Delta y) = b \end{cases}$$

The term $(\Delta A)(\Delta y)$ is a 2nd order perturbation which can be disregarded, thus $A(\Delta y) \approx -(\Delta A)y$ and thus $\Delta y \approx -A^{-1}(\Delta A)y$. Taking norms on both sides, we see that

$$\begin{aligned}\|\Delta y\| &\approx \|A^{-1}(\Delta A)y\| \\ &\leq \|A^{-1}\|\|\Delta A\|\|y\|.\end{aligned}$$

This implies

$$\begin{aligned}\frac{\|\Delta y\|}{\|y\|} &\leq \|A^{-1}\|\|\Delta A\| \\ &= \|A^{-1}\|\|A\|\frac{\|\Delta A\|}{\|A\|}.\end{aligned}$$

So we have the same relative condition number as the one for $Ax = b$ for fixed coefficient matrix A and variable vector b. So $\|A^{-1}\|\|A\|$ is a fair relative condition number to use.

Theorem 1.1. *Let λ be a simple eigenvalue of a square matrix A and let v and w be the corresponding right and left eigenvectors (i.e. $Av = \lambda v$ and $w^*A = \lambda w^*$ where $*$ denotes conjugate transpose). Let E be a small perturbation of A such that $(A+E)(v+\delta v) = (\lambda + \delta\lambda)(v + \delta v)$ and $(w+\delta w)^*(A+E) = (\lambda+\delta\lambda)(w+\delta w)^*$. Then*

$$|\delta\lambda| \leq \frac{1}{\cos(\text{angle}(v,w))}\|E\|_2.$$

Thus $1/\cos(\text{angle}(v,w))$ is an upper bound of the absolute condition number of simple eigenvalue).

Remark. Note that

$$\cos(\text{angle}(v,w)) = \frac{|v^*w|}{\|v\|_2\|w\|_2}.$$

1.1 Roots of a Polynomial

Finding roots a polynomial in monomial basis numerically is a bad idea. Here's a preliminary example:

$$(x-1)^2 = x^2 - 2x + 1 = 0.$$

Now consider a slightly perturbed problem:

$$(x-1-\delta)(x-1+\delta) = x^2 - 2x + 1 - \delta^2 = 0.$$

If $\delta < \varepsilon^{1/2}$ (e.g. $\delta = 10^{-9}$), then the exact solution is $\alpha_1 = 1 - 10^{-9}$ and $\alpha_2 = 1 + 10^{-9}$, but $1 - \delta^2 = 1$ in double position, so we will get $\alpha_1 = \alpha_2 = 1$.

In general, if a polynomial in monomial basis has a root of multiplicity m. Then an absolute perturbation in the monomial basis coefficients of magnitude $O(\delta^m)$ would be sufficient to product an absolute perturbation in this repeated root of magnitude $O(\delta)$. For instance, consider

$$(x-2)^4 = x^4 - 8x^3 + 24x^2 - 32x + 16.$$

Now changing one of the coefficients by 10^{-16} would change the root by roughly 10^{-4}. For instance, one of the roots of $(x-2)^4 - 10^{-16}$ is

$$\alpha = \frac{19999}{10000} = 1.9999.$$

For another example, consider

$$p_{24}(x) = (x-1)(x-2)\cdots(x-23)(x-24) = x^{24} + a_{23}x^{23} + \cdots + a_1 x + a_0,$$

where $a_0 = 24!$. The question we ask is: how sensitive are the roots to a perturbation of the coefficients? Let x_j be the jth root of p. First, let us rewrite this as

$$p(x_j; a_0, a_1, \ldots, a_{23}) = 0.$$

Assume there is a small perturbation in the coefficient a_i only (so $a_i \to a_i + \delta a_i$), and this leads to a small change in x_j ($x_j \to x_j + \delta \alpha_j$). Then we have

$$p(x_j + \delta\alpha_j, a_1, \ldots, a_i + \delta a_i, \ldots, a_{23}) = 0.$$

It follows that

$$p(x_j + \delta x_j; a_0, \ldots, a_i + \delta a_i, \ldots, a_{23}) - p(x_j; a_0, a_1, \ldots, a_{23}) = 0.$$

By Taylor's theorem we have

$$p(x_j + \delta x_j; a_0, \ldots, a_i + \delta a_i, \ldots, a_{23}) - p(x_j; a_0, a_1, \ldots, a_{23}) \approx \partial_{a_i} p(x_j; \boldsymbol{a})\delta a_i + \partial_{x_j} p(x_j; \boldsymbol{a})\delta x_j.$$

We find that

$$\kappa_{\text{abs}} \approx \lim_{\delta a_i \to 0} \left| \frac{(x_j + \delta x_j) - x_j}{(a_i + \delta a_i) - a_i} \right| = \lim_{\delta a_i \to 0} \left| \frac{\delta x_j}{\delta a_i} \right| = \lim_{\delta a_i \to 0} \left| \frac{\partial_{a_i} p(x_j; \boldsymbol{a})}{\partial_x p(x_j; \boldsymbol{a})} \right| = \lim_{\delta a_i \to 0} \left| \frac{x_j^i}{p'(x_j)} \right|.$$

Similarly, we have

$$\kappa_{\text{rel}} = \left| \frac{a_i x_j^{i-1}}{p'_{24}(x_j)} \right|.$$

Now consider $i = j = 18$. Then $a_{18} = 4.149 \times 10^{11}$ and $p'(x_{18}) = (17!)6!$. Then $\kappa_{\text{rel}} \approx 3.54 \times 10^{15}$.

1.2 Algorithm Stability

Let $y = f(x)$ be the exact solution to a math problem where x is the problem data. Let $\widehat{y} = \widehat{f}(x)$ be the computed solution by a specific numerical algorithm. We hope that

$$\frac{\|\widehat{f}(x) - f(x)\|}{\|f(x)\|}$$

is very small. The algorithm has a very high accuracy if this can be achieved for any valid input data. However if the problem is ill-conditioned, then such a hope is not realistic. In this case, it is reasonable to expect that the algorithm is **stable**, i.e. for any valid problem data at x, there exists a small δx with $\|\delta x\|/\|x\| = O(\varepsilon)$ such that

$$\frac{\|\widehat{f}(x) - f(x + \delta x)\|}{\|f(x + \delta x)\|} = O(\varepsilon).$$

1.2.1 Backward Stability

A stronger definition is **backward stability**: for a given problem data x, let the true solution be $f(x)$ and let $\widehat{f}(x)$ be a computed solution. If $\widehat{f}(x) = f(x + \delta x)$ for some δx satisfying $\|\delta x\|/\|x\| = O(\varepsilon_{\text{mach}})$ for any valid problem data x of interest, then this numerical algorithm is backward stable. Here, δx is called a backward error and $\widehat{f}(x) - f(x)$ is the corresponding forward error. A rule of thumb is that (relative) forward error is less than or equal to (relative) condition number times (relative) back error:

$$f(x + \delta x) - f(x) \le \kappa_{\text{rel}} \cdot \delta x$$

Consider $A = \begin{pmatrix} \varepsilon/2 & -1 \\ 1 & 1 \end{pmatrix}$. The LU factorization of A without pivoting in exact arithmetic is

$$A = \begin{pmatrix} 1 & 0 \\ 2/\varepsilon & 1 \end{pmatrix} \begin{pmatrix} \varepsilon/2 & -1 \\ 0 & 1 + 2/\varepsilon \end{pmatrix} = LU.$$

In computer arithmetic, we set $\widehat{L} = L$ and $\widehat{U} = \begin{pmatrix} \varepsilon/2 & -1 \\ 0 & 2/\varepsilon \end{pmatrix}$. In double point floating arithmetic, we have $\varepsilon = 2^{-52}$ so this can be represented on a computer. The point is that $2/\varepsilon$ is relatively large compared to 1, so in computer arithmetic we have $1 + 2/\varepsilon = 2/\varepsilon$ (try to add this in matlab, for example $1 + 2/\text{eps} - 2/\text{eps} = 2/\text{eps} - 2/\text{eps} = 0$ in matlab). Now we have

$$\widehat{L}\widehat{U} = \begin{pmatrix} \varepsilon/2 & -1 \\ 1 & 0 \end{pmatrix} = \widehat{A} = A + \delta A.$$

So $\delta A = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$ and $\|\delta A\|_\infty / \|A\|_\infty = 1/2 \gg O(\varepsilon_{\text{mach}})$ (it's nowhere close to machine coefficient). Therefore LU factorization without pivoting is not backward stable.

We have a few possible algorithms for solving the linear system $Ax = b$:

1. Gaussian elimination (GE) or LU factorization without pivoting.
2. GE/LU with partial pivoting.
3. Cramer's rule.
4. Compute A^{-1} first then multiply A^{-1} with b.
5. QR factorization of A ($A = QR$ then $x = R^{-1}Q^\top b$).

Which of these are backwards stable?

Theorem 1.2. *Let $\widehat{x}$ be a computed solution to a nonsingular linear system $Ax = b$, and let $r = b - A\widehat{x}$ be the residual vector of $\widehat{x}$. Assume that $\widehat{x}$ is the exact solution of $(A + \delta A)\widehat{x} = b$. Then*

$$\min_{\delta A} \frac{\|\delta A\|_2}{\|A\|_2} = \frac{\|r\|_2}{\|A\|_2\|\widehat{x}\|_2}.$$

Proof. Note that $(\delta A)\widehat{x} = b - A\widehat{x} = r$, so $\|\delta A\|\|\widehat{x}\| \geq \|\delta A\widehat{x}\| = \|r\|$. Therefore

$$\|\delta A\| \geq \|r\|\|\widehat{x}\|.$$

Now we show that the inequality can be achieved for some δA. Consider $\delta A = \frac{r(\widehat{x})^\top}{\widehat{x}^\top\widehat{x}}$ (a rank-1 matrix). Check

$$(A + \delta A)\widehat{x} = b.$$

We claim that

$$\|\delta A\| = \frac{\|r\|\|\widehat{x}\|}{\|\widehat{x}\|^2} = \frac{\|r\|}{\|\widehat{x}\|}.$$

□

Remark. This gives a very practical way to assess if an algorithm for solving $Ax = b$ is backward stable or not.

1.2.2 Forward Stability

If an algorithm always produces a forward error that is similar magnitude to the forward error produced by a backward stable algorithm, then this algorithm is **forward stable**.

Let A be a nonsingular matrix of order n. Assume that no zero pivot arises during the LU factorization of A without pivoting in exact arithmetic such that $A = LU$ (exactly). Then for a sufficiently small machine precision $\varepsilon = \varepsilon_{\text{mach}}$, this factorization can also be completed without breakdown in floating point arithmetic. Let $\widehat{L}$ and $\widehat{U}$ be the computed LU factorization of A. Then it can be shown that

$$\widehat{L}\widehat{U} = \widehat{A} = A + \delta A,$$

where δA satisfies

$$\frac{\|\delta A\|}{\|\widehat{L}\|\|\widehat{U}\|} = O(\varepsilon). \tag{1}$$

Similarly, let $|A|$ be the matrix obtained by taking absolute values of the elements of A. Then it can be shown that

$$|\delta A| \leq \frac{n\varepsilon}{1 - n\varepsilon}|\widehat{L}||\widehat{U}|$$

elementwise. In other words,

$$\frac{|\delta A|}{|\widehat{L}||\widehat{U}|} \leq n\varepsilon + n^2\varepsilon^2 + n^3\varepsilon^3 + \cdots = O(\varepsilon).$$

In addition, if we use such computed $\widehat{L}$ and $\widehat{U}$ factors to perform forward and backward substitutions and obtain computed solution $\widehat{x}$, then $\widehat{x}$ is the exact solution to $(A + \delta A)\widehat{x} = b$ where

$$|\delta A| \leq \frac{3n\varepsilon}{1 - 3n\varepsilon}|\widehat{L}||\widehat{U}|$$

elementwise.

To achieve backwards stability, we need $\|\delta A\|/\|A\| = O(\varepsilon)$, or $|\delta A| \leq |A|O(\varepsilon)$ elementwise. Therefore, whether LU factorization without partial pivoting is backwards stable depends on whether we have $\|\widehat{L}\|\|\widehat{U}\| \leq C_n\|A\|$ or $|\widehat{L}||\widehat{U}| \leq C_n|A|$ elementwise for some C_n that is not too large. If

$$\|\widehat{L}\|\|\widehat{U}\| \leq C_n\|A\| \text{ or } |\widehat{L}||\widehat{U}| \leq C_n|A|$$

can be achieved for some moderate C_n, then LU factorization without pivoting is backward stable. But how do we determine if C_n is moderate? To answer this question, let us define the growth factor e_n for LU factorization with or without pivoting. Let the (i,j) element of A be a_{ij}. Let $A^{(k)}$ be the intermediate matrix after the kth step of LU factorization and its (i,j) element is $a_{ij}^{(k)}$. We set

$$e_n = \frac{\max\limits_{1\leq i,j,k\leq n} |a_{ij}^{(k)}|}{\max\limits_{1\leq i,j\leq n} |a_{ij}^{(k)}|}.$$

Example 1.3. labelexample Consider the matrix $A = \begin{pmatrix} \varepsilon/2 & -1 \\ 1 & 1 \end{pmatrix}$. To perform LU factorization without pivoting, we need

$$\text{row2} \leftarrow -\frac{2}{\varepsilon}\text{row1} + \text{row2}.$$

Then we obtain $L = \begin{pmatrix} 1 & 0 \\ 2/\varepsilon & 1 \end{pmatrix}$, $U = \begin{pmatrix} \varepsilon/2 & -1 \\ 0 & 1+2/\varepsilon \end{pmatrix}$, and $A^{(1)} = \begin{pmatrix} \varepsilon/2 & -1 \\ 2/\varepsilon & 1+2/\varepsilon \end{pmatrix}$. For this example, we have

$$e_2 = \frac{1+\frac{2}{\varepsilon}}{1} = 1 + \frac{2}{\varepsilon}.$$

So for this example, as $\varepsilon \to 0$ we have $e_2 \to \infty$.

For LU factorization, we have

$$\||L||U|\|_\infty \leq (1 + 2(n^2+n)e_n)\|A\|_\infty$$

with exact L and U. If e_n is small, then

$$\||L||U|\| \leq C_n\|A\|$$

for some moderate C_n. The bottom line is that we need e_n to not approach infinity as ε approaches zero. Ideally, we hope the e_n grows mildly with n. For no pivoting, the bottom line is violated (no futher discussion is needed). For partial pivoting, consider the matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}.$$

From A we obtain

$$A^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ x & 1 & 0 & 0 & 2 \\ x & -1 & 1 & 0 & 2 \\ x & -1 & -1 & 1 & 2 \\ x & -1 & -1 & -1 & 2 \end{pmatrix}.$$

We see that $e_n = 2^{n-1}$. This is the single worst case. So LU or GE with partial pivoting is backward stable for matrices of a fixed size n, however they are not backwards stable for matrices of any size. Assume that $GEPP$ is used to solve $Ax = b$. Then the computed solution $\widehat{x}$ is the exact solution to $(A + \delta A)\widehat{x} = b$ where

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \leq \frac{3n^3 e_n \varepsilon}{1 - 3n\varepsilon}.$$

1/31

Note that (1) still holds with pivoting. Let P and Q be permutation matrices such that $PAQ = LU$. Note that $\|A\| = \|PAQ\|$ in either 1, 2, or ∞ norm (and Frobenius norm as well). We need

$$\|\widehat{L}\|\|\widehat{U}\| \leq C_n\|A\|$$

where C_n is a moderate number that depends only on n, not ε. If your $\widehat{L}$ or $\widehat{U}$ contain (very) large elements (in absolute value), then your algorithm will be unstable. Now consider the matrix

$$\begin{pmatrix} -3 & 0 & 4 & 7 & 8 \\ 2 & -6 & 1 & 2 & 5 \\ 4 & -1 & 5 & 0 & 6 \\ -8 & 0 & 11 & 3 & -9 \\ 1 & -4 & 6 & 5 & 8 \end{pmatrix}$$

For partial pivoting we would choose -8 as our first pivot, and for root pivoting we choose 11 as our first pivot.

2 QR Factorization

Let A be an $m \times n$ matrix where $n \leq m$ and suppose $A = Q_1 R_1$ where Q_1 is $m \times n$ matrix that satisfies $Q_1^\top Q_1 = 1$ (Q_1 has orthonormal columns) and R_1 is $n \times n$ upper triangular matrix. This is the economic QR factorization. The full QR factorization is $A = Q_1 Q_2 R_1$ where Q_1 is $m \times n$, Q_2 is $m \times (m-n)$, R_1 is $n \times n$, and Q is $(m-n) \times n$, and range$(Q_1) \perp$ range(Q_2), $Q_1^\top Q_1 = 1 = Q_2^\top Q_2$, $Q_1^\top Q_2 = 0$. The matrix Q_2 is primarily used for theoretical analysis, not used in actual algorithms. More specifically,

$$A = Q_1 R_1 = [q_1, \ldots, q_n]\left(r_{ij}\right).$$

Multiply the right by e_k gives us

$$Ae_k = \sum_{i=1}^{k} q_i r_{ik}.$$

So the kth column of A is a linear combination of the first k columns of Q_1. If A has full column rank n, then R_1 is nonsingular (i.e. $r_{11}, r_{22}, \ldots, r_{nn}$ are all nonzero). Therefore $AR_1^{-1} = Q_1$. The kth column of Q_1 is a linear combination of the first k columns of A. If A is of rank k ($k < n$) and assume that the first l columns of A are linearly independent but the $(l+1)$st solum of A is a linear combination of the first l columns of A. Then the top left $l \times l$ submatrix of R is nonsingular, but $r_{(l+1)(l+1)} = 0$.

How to compute a reduced QR factorization? Note that

$$q_k = \frac{a_k - r_{1k}q_1 - r_{2k}q_2 - \cdots r_{k-1,k}q_{k-1}}{r_{kk}}$$

where the coefficients $r_{ik} = q_i^\top a_k$ in practice using $r_{ik} = q_i^\top a_k$ as the coefficients during Gram-Schmidt orthogonalization is called the classical Gram-Schmidt.
Pro: all $r'_{ik}s$ can be computed in parallel.
Con: $q_1, \ldots, q_k$ tend to quickly lose orthogonality as k increases.

To improve numerical orthogonality among the q_k vectors use modified G-S. At step k

$$\begin{aligned} q_k^{(0)} &= a_k \\ q_k^{(1)} &= q_k^{(0)} - q_1(q_1^\top q_k^{(0)}) \\ &\vdots \\ q_k^{(k-1)} &= q_k^{(k-2)} - q_{k-1}(q_{k-1}^\top q_k^{(k-2)}) \end{aligned} \qquad \begin{aligned} r_{kk} &= \|q_k^{(k-1)}\|_2 \\ q_k &= \frac{q_k^{(k-1)}}{r_{kk}}. \end{aligned}$$

2.1 Modified G-S with Reorthogonalization

```
for i=1:n
    qi = ai = Aei
```

For $1 \leq i \leq n$ set $q_i = a_i = Ae_i$. For $1 \leq j \leq i-1$ set

$$r_{ji} = q_j^\top q_i$$
$$q_i = q_i - q_j r_{ji}$$

$$\begin{aligned}
&\text{for } j = 1 : i-1 \\
&\qquad \delta r_{ji} = q_j^\top q_i \\
&\qquad q_i = q_i - q_j \delta r_{ji} \\
&\qquad \text{end} \\
&\qquad r_{ii} = \|q_i\|_2 \\
&\qquad q_i = q_i / r_{ii}
\end{aligned}$$

Definition 2.1. Let $f\colon Y \to X$ be a morphism of schemes. We say f is **universally closed** if for every morphism of schemes $Z \to X$, the morphism $Y \times_X Z \to Z$ is a closed map of the underlying topological spaces.

Example 2.1. labelexample Suppose that $X = \operatorname{Spec} R$, $Y = \operatorname{Spec} A$, and $Z = \operatorname{Spec} B$ where R is a ring and A and B are R-algebras. Then the morphism of schemes $Y \times_X Z \to Z$ corresponds to the morrphism of R-algebras $B \to A \otimes_R B$ defined by $b \mapsto 1 \otimes b$. (let $\mathfrak{q} = \{b \mid 1 \otimes b \in \mathfrak{r}\}$