On Numerical Analysis (third week)

Hamad El Kahza

Conditioning and stability, LU factorization

1 Conditioning and Stability

In numerical methods, it is important to know how reliable the numerical results produced by some algorithm are, and how sensitive the algorithm is with respect to the inevitable noise in the input variables and system parameters, due possibly to some observational error and truncation error.

- We need to know how sensitive the output of a given method is with respect to the various perturbations that may occur in computational process.
- if a large difference in the output may be caused by a small change in the input, then the result is not reliable.
- How sensitive a problem is and whether the numerical solution is reliable or not depends on whether the problem is well behaved or ill behaved:
  - If a small change in the input causes a small change in the output, the system is *well conditioned* or *well behaved*.
  - If a small change in the input can cause a large change in the output, the system is *ill conditioned* or *ill behaved*.

How well or ill conditioned a system is can be measured quantitatively by the *absolute* or *relative condition numbers*:

- The *absolute condition number* $\hat{\kappa}$ is the upper bound of the ratio between the change in output and change in input:

$$\hat{\kappa} \geq \frac{||\text{change in output}||}{||\text{change in input}||}, \qquad ||\text{change in output}|| \;\leq\; \hat{\kappa}\, ||\text{change in input}||$$

- The *relative condition number* κ is the upper bound of the ratio between the relative change in output and relative change in input:

$$\kappa \geq \frac{||\text{change in output}||/||\text{output}||}{||\text{change in input}||/||\text{input}||}, \qquad \frac{||\text{change in output}||}{||\text{output}||} \;=\; \kappa\,\frac{||\text{change in input}||}{||\text{input}||}$$

Here $||x||$ is the norm (representing “size”) of any variable x (scalar, vector, matrix, function, etc.).

1.1 System with single input and output variables

When the perturbation δx is small, the Taylor expansion of the function can be approximated as

$$f(x+\delta x) = f(x) + f'(x)\delta x + f''(x)\frac{\delta x^2}{2} + \cdots \approx f(x) + f'(x)\delta x$$

i.e.,

$$\delta y = f(x+\delta x) - f(x) \approx f'(x)\delta x$$

Taking the absolute value or modulus on both sides, we get

$$\big|\delta y\big| = \big|f(x+\delta x) - f(x)\big| \approx \big|f'(x)\delta x\big| \le \big|f'(x)\big|\,\big|\delta x\big| = \hat{\kappa}\big|\delta x\big|$$

where $\hat{\kappa} = |\delta y|/|\delta x| = |f'(x)|$ is the absolute condition number. Dividing both sides by $|y| = |f(x)|$, we get

$$\frac{\big|\delta y\big|}{\big|y\big|} = \frac{\big|f(x+\delta x) - f(x)\big|}{\big|f(x)\big|} \approx \frac{\big|f'(x)\big|\,\big|\delta x\big|}{\big|f(x)\big|} = \frac{\big|x\big|\,\big|f'(x)\big|}{\big|f(x)\big|}\,\frac{\big|\delta x\big|}{\big|x\big|} = \kappa\frac{\big|\delta x\big|}{\big|x\big|}$$

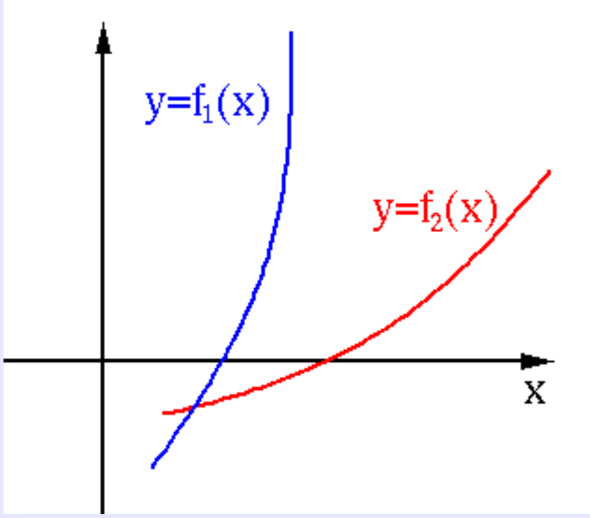
where κ is the relative condition number:

$$\kappa = \frac{|\delta y|/|y|}{|\delta x|/|x|} = \frac{|x|\,|f'(x)|}{|f(x)|} = \frac{|f'(x)|}{|f(x)|/|x|}$$

The discussion above is to evaluate the output $y = f(x)$ of a system as a function of the input x. However, if the task is to solve a given equation $f(x) = 0$, the problem needs to be converted into the form of evaluating the function $x = g(y) = f^{-1}(y)$ at $y = 0$. The absolute condition number of this function $g(y) = f^{-1}(y)$ is

$$\hat{\kappa} = |g'(y)| = \left|\frac{d}{dy}[f^{-1}(y)]\right| = \frac{1}{\big|f'(x)\big|}$$

In the figure below, $|f_1'(x)| > |f_2'(x)|$, therefore for the problem of solving $f(x) = 0$, $f_1(x)$ is better conditioned than $f_2(x)$, but for the problem of evaluating $y = f(x)$, $f_1(x)$ is more ill-conditioned than $f_2(x)$.



Example: Consider the function

$$y = f(x) = \frac{x}{1-x}, \qquad f'(x) = \frac{1}{(1-x)^2}$$

x	-0.99	-0.98	0.98	0.99
y=f(x)	-0.4975	-0.4949	49.0	99.0

at $x = -0.99$,

$$\kappa = \frac{|f'(x)|}{|f(x)|/|x|} = \frac{1}{|1-x|} = \frac{1}{1.99} \approx 0.5$$

$$|\delta x| = |-0.99 - (-0.98)| = 0.01, \quad |\delta y| = |-0.4975 - (-0.4949)| = 0.0026$$

$$\frac{|\delta y|}{|\delta x|} = \frac{0.0026}{0.01} = 0.26, \qquad \frac{|\delta y|/|y|}{|\delta x|/|x|} = \frac{0.0026/0.4975}{0.01/0.99} = 0.5174$$

At this point, the problem of evaluating $y = f(x)$ is well-conditioned.

-

- the problem of solving the equation $f(x) = 0$ is very ill-conditioned, as in the neighborhood of the root $x = 0$, $1/f'(x)$ is very large, any value in a wide range in x could result in $f(x) \approx 0$, and therefore considered as a solution, which is certainly not reliable.

- at $x = 0.99$,

$$\kappa = \frac{|f'(x)|}{|f(x)|/|x|} = \frac{1}{|1-x|} = \frac{1}{0.01} = 100$$

$$|\delta x| = |0.99 - 0.98| = 0.01, \quad |\delta y| = |99 - 49| = 50$$

$$\frac{|\delta y|}{|\delta x|} = \frac{50}{0.01} = 5000, \qquad \frac{|\delta y|/|y|}{|\delta x|/|x|} = \frac{50/49}{0.01/0.98} = 100$$

  At this point, the function is ill-conditioned.

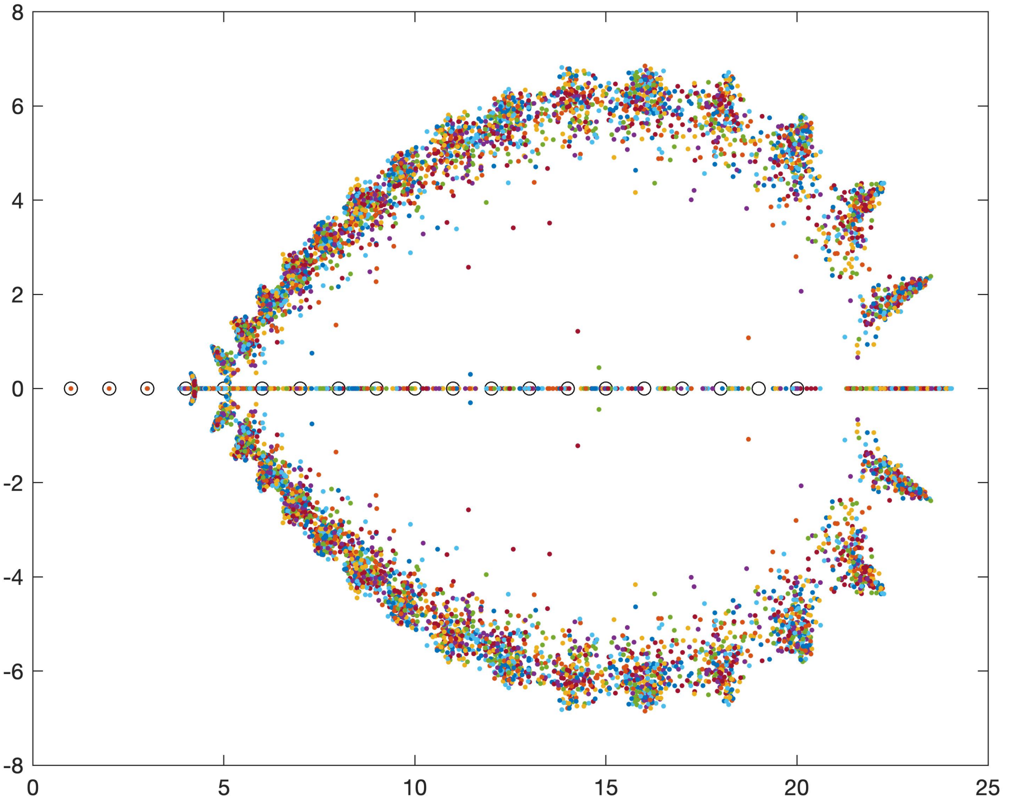
We see the function is well-conditioned at $x = -0.99$ but ill-conditioned at $x = 0.99$.

Wilkinson Polynomial Let $f : \mathbb{C}^{n+1} \to \mathbb{C}^n$ be the function that maps a collection of $n+1$ coefficients $(c_n, c_{n-1}, \ldots, c_0)$ to the n roots of the polynomial $c_n x^n + c_{n-1}x^{n-1} + \ldots + c_2 x^2 + c_1 x + c_0$. Finding polynomial roots is an extremely ill-conditioned problem in general, so the condition number of f is likely very large. To see this, consider the *Wilkinson polynomial*, made famous by James H. Wilkinson in 1963:

$$w(x) = \prod_{r=1}^{20}(x - r) = x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + \cdots.$$

Let $\tilde{w}(x)$ be $w(x)$ where the effect of perturbing the coefficients can be more dramatically shown using the Wilkinson polynomial. The following code computes and compares the roots of $\tilde{w}(x)$ and $w(x)$ using MATLAB :

```
w = poly(1:20);
plot(1:20,0,'ko'),
hold on
for k = 1:500
    q = w.*(1+1e-9*randn(size(w)));  % relative perturbations
    r = roots(q);
    plot(real(r),imag(r),'.')
end
```



1.2 Systems with Multiple input and output variables

The results above can be generalized to a system of multiple inputs $\mathbf{x} = [x_1, \cdots, x_N]^T$ and multiple outputs $\mathbf{y} = [y_1, \cdots, y_M]^T$ represented by a function $\mathbf{y} = \mathbf{f}(\mathbf{x})$. A change $\delta\mathbf{x}$ in the input will cause certain change in the output:

$$\delta\mathbf{y} = \mathbf{f}(\mathbf{x} + \delta\mathbf{x}) - \mathbf{f}(\mathbf{x})$$

we have $\kappa = ||\mathbf{A}||\;||\mathbf{A}^{-1}||$. If the singular values of $\mathbf{A}$ are $\{\sigma_1 \geq \cdots \geq \sigma_n\}$, the singular values of $\mathbf{A}^{-1}$ are $\{1/\sigma_n \geq \cdots \geq 1/\sigma_1\}$, and their norms can be written in terms of their greatest and smallest eigenvalues, respectively: $||\mathbf{A}|| = \sigma_{max} = \sigma_1$ and $||\mathbf{A}^{-1}|| = 1/\sigma_{min} = 1/\sigma_n$. Now we have

$$\kappa(\mathbf{A}) = ||\mathbf{A}||\;||\mathbf{A}^{-1}|| = \frac{\sigma_{max}}{\sigma_{min}}$$

We see that the condition number of $\mathbf{A}$ is large if its σ_{max} and σ_{min} are far apart, but it is small if otherwise.

In fact, $\kappa(\mathbf{A})$ is a measurement of how close $\mathbf{A}$ is to singularity. When $\mathbf{A}$ is singular, one or more of its singular values are zero, i.e., $\sigma_{min} = 0$, then $\kappa(\mathbf{A}) = \infty$.

In other words, the absolute condition number of f is the limit of the change in output over the change of input. Similarly, the *relative condition number* of f is the limit of the relative change in output over the relative change in input,

For example, the matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0000000001 \end{bmatrix}$$

is extremely ill-conditioned, with $\kappa(A) \approx 4 \times 10^{10}$. Solving the systems $A =$ _1 and $A =$ _2 can result in wildly different answers, even when _1 and _2 are extremely close.

```
>>> import numpy as np
>>> from scipy import linalg as la

>>> A = np.array([[1, 1], [1, 1+1e-10]])
>>> np.linalg.cond(A)
39999991794.058899

# Set up and solve a simple system of equations.
>>> b1 = np.array([2, 2])
>>> x1 = la.solve(A, b1)
>>> print(x1)
[ 2.  0.]

# Solve a system with a very slightly different vector b.
>>> b2 = np.array([2, 2+1e-5])
>>> la.norm(b1 - b2)
>>> x2 = la.solve(A, b2)
>>> print(x2)
[-99997.99172662  99999.99172662]   # This solution is hugely different!
```

Another example: Consider solving the linear system $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 3.0 & 2.0 \\ 1.0 & 4.0 \end{bmatrix}, \qquad \mathbf{A}^{-1} = \begin{bmatrix} 0.8 & -0.4 \\ -0.2 & 0.6 \end{bmatrix}$$

The singular values of $\mathbf{A}$ are $\sigma_1 = 2.5583$ and $\sigma_2 = 0.9772$, the condition number is

$$\kappa = ||\mathbf{A}||\ ||\mathbf{A}^{-1}|| = \frac{\sigma_1}{\sigma_2} = 2.618$$

which is small, indicating this is a well-behaved system. Given two similar inputs $\mathbf{b}_1 = [1,\ 1]^T$ and $\mathbf{b}_2 = [0.99,\ 1.01]^T$ with $\delta\mathbf{b} = [0.01,\ -0.01]^T$, we find the corresponding solutions:

$$\mathbf{x}_1 = \mathbf{A}^{-1}\mathbf{b}_1 = \begin{bmatrix} 0.4 \\ 0.4 \end{bmatrix}, \qquad \mathbf{x}_2 = \mathbf{A}^{-1}\mathbf{b}_2 = \begin{bmatrix} 0.388 \\ 0.408 \end{bmatrix}, \qquad \delta\mathbf{x} = \begin{bmatrix} 0.012 \\ -0.008 \end{bmatrix}$$

We have

$$||\delta\mathbf{b}|| = 0.0141, \quad ||\mathbf{b}_1|| = 1.4142, \quad ||\delta\mathbf{x}|| = 0.0144, \quad ||\mathbf{x}_1|| = 0.5657$$

and

$$\frac{||\delta\mathbf{x}||/||\mathbf{x}_1||}{||\delta\mathbf{b}||/||\mathbf{b}_1||} = \frac{0.0255}{0.01} = 2.5495$$

Now consider a different matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1.000 & 1.000 \\ 1.001 & 0.999 \end{bmatrix}, \qquad \mathbf{A}^{-1} = \begin{bmatrix} -999 & 1000 \\ 1001 & -1000 \end{bmatrix}$$

The singular values of $\mathbf{A}$ are $\sigma_1 = 1.0$ and $\sigma_2 = 0.0005$, the condition number is

$$\kappa = ||\mathbf{A}||\ ||\mathbf{A}^{-1}|| = \frac{\sigma_1}{\sigma_2} = 2000$$

indicating matrix $\mathbf{A}$ is close to singularity, and the system is ill-conditioned. The solutions corresponding to the same two inputs $\mathbf{b}_1 = [1,\ 1]^T$ and $\mathbf{b}_2 = [0.99,\ 1.01]^T$ are

$$\mathbf{x}_1 = \mathbf{A}^{-1}\mathbf{b}_1 = \left[\begin{array}{c} 1 \\ 1 \end{array}\right], \qquad \mathbf{x}_2 = \mathbf{A}^{-1}\mathbf{b}_2 = \left[\begin{array}{r} 20.99 \\ -19.01 \end{array}\right], \qquad \delta\mathbf{x} = \left[\begin{array}{r} -19.99 \\ 20.01 \end{array}\right]$$

We can further find

$$||\delta\mathbf{b}|| = 0.0141, \quad ||\mathbf{b}_1|| = 1.4142, \quad ||\delta\mathbf{x}|| = 28.2843, \quad ||\mathbf{x}_1|| = 1.4142$$

and the condition number is:

$$\frac{||\delta\mathbf{x}||/||\mathbf{x}_1||}{||\delta\mathbf{b}||/||\mathbf{b}_1||} = \frac{200}{0.01} = 2000$$

We see that a small relative change $||\delta\mathbf{b}||/||\mathbf{b}_1|| = 0.01$ in the input caused a huge change $||\delta\mathbf{x}||/||\mathbf{x}_1|| = 20$ in the output (2000 times greater).

2 LU factorization and pivoting

Pivoting Strategies

Ex: $\left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \xrightarrow{\text{Exchange Eq1 and Eq2}} \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right]$, solution $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Suppose small errors!

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \to \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & \underbrace{1-10^{20}}_{\approx -10^{20} \text{ in the computer}} & \underbrace{2-10^{20}}_{\approx -10^{20}} \end{array}\right]$$

$$\Rightarrow \quad x \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Need to pivot even if $a_{ii} \neq 0$

Partial pivoting! Find largest pivot element

$$\Rightarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 10^{-20} & 1 & 1 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & \underbrace{1-10^{-20}}_{\approx 1} & \underbrace{1-2\cdot 10^{-20}}_{\approx 1} \end{array}\right]$$

$\Rightarrow \quad x \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ Exact!

Matrix Factorization

Gaussian Elimination without pivoting can be written as

$$\begin{bmatrix} x & \cdots & x \\ \vdots & \ddots & \vdots \\ x & \cdots & x \end{bmatrix} = \begin{bmatrix} 1 & & & 0 \\ x & 1 & & \\ \vdots & x & \ddots & \\ x & \cdots & x & 1 \end{bmatrix} \begin{bmatrix} x & - & \cdots & x \\ & \ddots & & \vdots \\ & 0 & \ddots & \vdots \\ & & & x \end{bmatrix}$$

$$A \quad = \quad L \qquad U$$

Solve $Ax=b \Leftrightarrow LUx=b$

1) Set $y=Ux$, solve $Ly=b$

   L is lower-triangular $\Rightarrow$ "forward solve"

2) Solve $Ux=y$

   U is upper-triangular $\Rightarrow$ "back solve"

Useful if solving many systems

$$\begin{aligned} Ax &= b_1 \\ Ax &= b_2 \\ &\vdots \end{aligned}$$

Ex:

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & -m_1 & 1 & \\ & -m_2 & & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9-5m_1 & 10-6m_1 & 11-7m_1 & 12-8m_1 \\ 13-5m_2 & 14-6m_2 & 15-7m_2 & 16-8m_2 \end{bmatrix}$$

Ex:

$$\underset{M^{(1)}}{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}}\underset{A}{\begin{bmatrix} 1 & 4 & 7 \\ \boxed{2} & 6 & 8 \\ \boxed{3} & 6 & 9 \end{bmatrix}} = \underset{A^{(2)} = M^{(1)}A}{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{bmatrix}}$$

$m_{21} = \frac{2}{1}$

$m_{31} = \frac{3}{1}$

$$\underset{M^{(2)}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}}\underset{A^{(2)}}{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & \boxed{-6} & -12 \end{bmatrix}} = \underset{U = A^{(3)} = M^{(2)}A^{(2)}}{\begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 0 \end{bmatrix}}$$

$m_{32} = \frac{-6}{-3} = 2$

$$M^{(2)}M^{(1)}A = A^{(3)} = U$$

$$A = \left[M^{(2)}M^{(1)}\right]^{-1}U = \underbrace{\left[M^{(1)}\right]^{-1}\left[M^{(2)}\right]^{-1}}_{L}U$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{22} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\Rightarrow \quad A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 0 \end{bmatrix}$$

Confirm:

$$\underbrace{\begin{bmatrix} 1 & & \\ -2 & 1 & \\ -3 & & 1 \end{bmatrix}}_{M^{(1)}} \underbrace{\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & & 1 \end{bmatrix}}_{[M^{(1)}]^{-1}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I}$$

$$\underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ & -2 & 1 \end{bmatrix}}_{M^{(2)}} \underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ & 2 & 1 \end{bmatrix}}_{[M^{(2)}]^{-1}} = I$$

$$[M^{(1)}]^{-1}[M^{(2)}]^{-1} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 3 & 2 & 1 \end{bmatrix}$$