Numerical Calculus

Course 3

2025

Anca Ignat

Binary machine numbers[1]

In 1985 IEEE published a report entitled Binary Floating Point Arithmetic Standard 754-1985 and un update in 2008, IEEE 754-2008, that provides standards for binary and decimal floating point numbers, formats for data interchange, algorithms for rounding arithmetic operations, and handling exceptions. These standards are followed by computer manufacturers that use floating-point hardware.

[1] R.L.Burden, J.D. Faires, A.M. Burden, *Numerical Analysis*, 10th ed., Cengage Learning, Boston, USA

A 64-bit binary representations is done in the following way: the first bit is the sign bit, the next 11 bits represent the exponent ***c*** which are followed by 52 bits that contain information regarding the binary fraction part, ***f***, also called mantissa:

$$(-1)^s 2^{c-1023}(1+f) \ .$$

0 10000000011 1011100100010000000000000000000000000000000000000000 = 27.56640625

[27.5664062499999982236431605997495353221893310546875, 27.5664062500000017763568394002504646778106689453125).

The smallest positive number that can be represented using this type of representation, is when $\boldsymbol{s = 0,\ c = 1,\ f = 0}$, that is:

$$\boldsymbol{z = 2^{-1022}(1+0) \approx 0.22251 \times 10^{-307}}$$

and the biggest one is for $\boldsymbol{s = 0, c = 2046,\ f = 1 - 2^{-52}}$

$$\boldsymbol{Z = 2^{1023}(2 - 2^{-52}) \approx 0.17977 \times 10^{309}}.$$

The numbers that occur in computations and are smaller than $\boldsymbol{z}$ are usually set to 0 (*underflow*) and those that are bigger than $\boldsymbol{Z}$ usually stop the computations (*overflow*).

Note that number 0 has two representations: $\boldsymbol{s=0, c=1, f=0}$ and $\boldsymbol{s=1, c=1, f=0}$.

Decimal Machine Numbers

Assume that the machine numbers are represented in the normalized decimal floating-point form:

$$\boldsymbol{\pm 0.d_1 d_2 \ldots d_k \times 10^n \quad 1 \le d_1 \le 9\ ,\ \ 0 \le d_i \le 9,\ i = 2,\ldots,k} \ \ \text{-}$$

This are ***k***-digit decimal machine numbers. Any positive real number ***y*** can be expressed by:

$$\boldsymbol{y = 0.d_1 d_2 \ldots d_k d_{k+1} d_{k+2} \ldots \times 10^n}$$

A ***k***-digit representation of ***y*** can be obtained by a simple chopping operation:

$$\boldsymbol{fl(y) = 0.d_1d_2\ldots d_k \times 10^n} \quad .$$

Another method, is the rounding operations:

$$\boldsymbol{fl(y) = 0.\delta_1\delta_2\ldots\delta_k \times 10^n}$$

If $\boldsymbol{d_{k+1} \geq 5}$ we add ***1*** to $\boldsymbol{d_k}$ to obtain ***fl(y)*** (*round up*), otherwise

a ***k***-digit chop is performed (*round down*).

A number $\boldsymbol{r^*}$ approximates the number $\boldsymbol{r}$ to $\boldsymbol{t}$ significant digits if $\boldsymbol{t}$ is the largest nonnegative integer for which:

$$\frac{|r - r^*|}{|r|} \leq 5 \times 10^{-t} \ .$$

For chopping we have:

$$\left| \frac{y - fl(y)}{y} \right| \leq 10^{-k+1}$$

And for rounding:

$$\left| \frac{y - fl(y)}{y} \right| \leq 0.5 \times 10^{-k+1}.$$

Finite-Digit Arithmetic

$$x +_c y = fl(fl(x) + fl(y))$$
$$x -_c y = fl(fl(x) - fl(y))$$
$$x \times_c y = fl(fl(x) \times fl(y))$$
$$x \div_c y = fl(fl(x) \div fl(y))$$

Sources of error in numerical computations

1. Errors in input data:
   - systematic errors or temporary perturbations in measurements,
   - round-off errors whenever dealing with irrational numbers, or with infinite decimal representation: 1/3, π, 1/7,…
2. Round-off errors during computations:
   - the result of elementary operations is not, always, exact

3. Discretization errors:
    - the solution of the solved problem is the limit of a sequence, the sum of an infinite series,
    - nonlinear functions are approximated by linear functions, computing the approximate derivative or integral for a function
4. Simplifications in the mathematical model
    - idealizations, ignoring some parameters
5. Human errors, bugs in numerical software.

Absolute Error, Relative Error

a – exact value

ã – approximate value

Absolute error : $a - \tilde{a}$ or $|a - \tilde{a}|$ or $\|a - \tilde{a}\|$

$$a = \tilde{a} \pm \Delta_a, \; |a - \tilde{a}| \le \Delta_a$$

Relative error: $a \neq 0$ $\quad \dfrac{a - \tilde{a}}{a}$ or $\dfrac{|a - \tilde{a}|}{|a|}$ or $\dfrac{\|a - \tilde{a}\|}{\|a\|}$

$\dfrac{|a - \tilde{a}|}{|a|} \le \delta_a$ (δ_a usually expressed in %).

If we have the following approximations: 1kg ±5g, 50g±5g, the absolute errors are the same, but for the first quantity the relative error is 0,5% and for the second, the relative error is ***10%***.

$$\boldsymbol{a_1 = \tilde{a}_1 \pm \Delta_{a_1}\ , a_2 = \tilde{a}_2 \pm \Delta_{a_2}\ ,}$$

$$\boldsymbol{a_1 \pm a_2 = (\tilde{a}_1 \pm \tilde{a}_2) \pm \left(\Delta_{a_1} \pm \Delta_{a_2}\right)}$$

$$\boldsymbol{\Delta_{a_1+a_2} \leq \Delta_{a_1} + \Delta_{a_2}\,.}$$

$\boldsymbol{a_1}$ has $\boldsymbol{\delta_{a_1}}$ relative error and $\boldsymbol{a_2}$ has relative error $\boldsymbol{\delta_{a_2}}$:

$$\boldsymbol{a = a_1 * a_2} \text{ or } \boldsymbol{\frac{a_1}{a_2}} \ \rightarrow\ \boldsymbol{\delta_a = \delta_{a_1} + \delta_{a_2}}\,.$$

Conditioning $\leftarrow\rightarrow$ stability

The conditioning of a problem describes the sensibility/variation of the solution with respect to the input data, assuming that all the computations are exact (regardless of the algorithm employed for solving the problem).

Let $\boldsymbol{x}$ be the exact input data, $\tilde{\boldsymbol{x}}$ an approximation for the input data, $\boldsymbol{P(x)}$ the exact solution of the problem, and $\boldsymbol{P(\tilde{x})}$ the solution with $\tilde{\boldsymbol{x}}$ as input. We assume that the computations were exact in obtaining the solutions ***P(x)*** and $\boldsymbol{P(\tilde{x})}$.

O problem is considered ***ill-conditioned*** if the difference between ***P(x)*** and $\boldsymbol{P(\tilde{x})}$ is large even if the relative error in input data $\dfrac{\|\boldsymbol{x-\tilde{x}}\|}{\|\boldsymbol{x}\|}$ is small.

The numerical conditioning of a problem is expressed as the ratio between output and input relative errors:

$$\boldsymbol{k(x)} = \frac{\dfrac{\| \boldsymbol{P(x)} - \boldsymbol{P(\tilde{x})} \|}{\| \boldsymbol{P(x)} \|}}{\dfrac{\| \boldsymbol{x} - \boldsymbol{\tilde{x}} \|}{\| \boldsymbol{x} \|}} \quad \text{for } \boldsymbol{x} \neq \boldsymbol{0} \text{ and } \boldsymbol{P(x)} \neq \boldsymbol{0}$$

A small value for ***k(x)*** characterizes a well-conditioned problem. Conditioning is a local property (it depends on the input ***x***). A problem is well-conditioned if it is well-conditioned for all input data.

Relative error in the output ≈

Condition number × Relative error in the input

Consider Wilkinson polynomial:

$$\boldsymbol{w(x) = (x-1)(x-2)\cdots(x-20) = x^{20} - 210x^{19} + P_{18}(x)}$$

If one changes the coefficient (***-210)*** of $\boldsymbol{x^{19}}$ with:

$$\boldsymbol{-210 - 2^{-23} = -210.0000001192}$$

the new polynomial’s roots (with 5 significant digits) are:

$\mathbf{1.00000\ 2.00000\ 3.00000\ 4.00000\ 5.00000\ 6.00001\ 6.99970\ 8.00727}$

$\mathbf{8.91725\ 20.84691\ 10.09527 \pm 0.64350}\boldsymbol{i}\ \mathbf{11.79363 \pm 1.65233}\boldsymbol{i}$

$\mathbf{13.99236 \pm 2.51883}\boldsymbol{i}\ \mathbf{16.73074 \pm 2.81262}\boldsymbol{i}\ \mathbf{19.50244 \pm 1.94033}\boldsymbol{i}$

For solving a problem $\boldsymbol{P}$, an algorithm $\tilde{\boldsymbol{P}}$ is implemented and run on a computer. Taking into account the inexact storing of data and inaccurate computations the exact solution and the computed one can be different:

$$\boldsymbol{P}(\boldsymbol{x}) \neq \tilde{\boldsymbol{P}}(\boldsymbol{x})$$

Numerical stability measures the magnitude of the numerical errors after running the implemented algorithm, assuming that the input data are exact, i.e., $\|\boldsymbol{P(x)} - \boldsymbol{\tilde{P}(x)}\|$ or $\dfrac{\|\boldsymbol{P(x)} - \boldsymbol{\tilde{P}(x)}\|}{\|\boldsymbol{P(x)}\|}$.

A ***numerically stable algorithm*** is an algorithm for which the relative error is the same order as the rounding error.

A ***numerically stable algorithm*** used for solving a **well-conditioned problem** yields results with very good precision.

An algorithm $\tilde{\boldsymbol{P}}$ for solving problem $\boldsymbol{P}$ is numerically stable if one of the following condition is fulfilled:

1. $\tilde{P}(x) \approx P(x)$ for all input ***x***;

2. there exists $\tilde{x}$ near ***x***, s.t. $\tilde{P}(x) \approx P(\tilde{x})$

x = exact data,

P(x) = exact solution using exact input,

$\tilde{P}(x)$ = „*computed*" solution using algorithm $\tilde{P}$ with exact input data

Solving linear systems of equations

- 1900 B.C., Babylon – first problems related to simultaneous linear equations
- 300 B.C. Babylon – problem on clay tablets:

  ” *There are two fields whose total area is 1800 square yards. One produces grain at the rate of 2/3 of a bushel per square yard while the other produces grain at the rate of 1/2 a bushel per square yard. If the total yield is 1100 bushels, what is the size of each field?”*

- 200-100 B.C. China – *Nine Chapters on the Mathematical Art* – a method very similar to Gaussian Elimination for solving a 3-dimensional linear system of equations (,,*There are three types of corn, of which three bundles of the first, two of the second, and one of the third make 39 measures. Two of the first, three of the second and one of the third make 34 measures. And one of the first, two of the second and three of the third make 26 measures. How many measures of corn are contained of one bundle of each type?")*

- 1545, Cardan – in *Ars Magna*, proposes a rule (*regula de modo)* for solving a system of 2 equations with 2 unknowns (similar to Cramer's rule)
- 1683, Seki Kowa, Japan – introduces the idea of „*determinant*"- „*Method of solving the dissimulated problems"*. He computes measures that we today call determinants for matrices ***2x2***, ***3x3***, ***4x4***, ***5x5*** related to solving equations but not systems of linear equations.

- 1683, Leibniz in a letter to l’Hôpital explains that the system of linear equations:

$$10+11x+12y=0$$
$$20+21x+22y=0$$
$$30+31x+32y=0$$

has a solution because:

10*21*32+11*22*30+12*20*31=10*22*31+11*20*32+12*21*30

(the condition that the determinant of the coefficients matrix must be 0).

Leibniz was convinced that good mathematical notation was the key to progress so he experimented with different notation for coefficient systems. His unpublished manuscripts contain more than 50 different ways of writing coefficient systems which he worked on during a period of 50 years beginning in 1678. Leibniz uses the notion of „*resultant”* instead of determinant and he proved Cramer’s rule for „resultants”. He knew that every determinant could be expanded using any column (Laplace expansion).

- 1750, Cramer introduces a formula based on determinants for solving a system of linear equations – *Cramer's rule* – *„Introduction in the analysis of algebraic curves"* (he presents an algorithm for general *n x n* systems):

  ‚One finds the value of each unknown by forming ***n*** *fractions of which the common denominator has as many terms as there are permutations of* ***n*** *things'*
- 1764 Bezout, 1771 Vandermonde, 1772 Laplace – rules for computing determinants

- 1773 Lagrange – first implicit use of matrices related to bilinear forms that appear in the problem of optimization for real functions depending on 2 or more variables (he wanted to characterize the optimum points for functions depending on 2 or more variables)

- 1800-1801, Gauss introduces for the first time the term „*determinant”* (determines the properties of the quadratic form) – *Disquisitiones arithmeticae(1801)*; he describes the matrix multiplication and matrix inverse notions in the context of quadratic forms and the associated bi-dimensional table of coefficients. Gauss proposed the *Gaussian elimination* method for solving a linear system with 6 equations in six unknowns, when he studied the orbit of Pallas asteroid.

- 1812, Cauchy uses the term of „*determinant*” in the modern sense.
- 1826, Cauchy finds the eigenvalues and deduces results for diagonalizing a matrix. He introduces the idea of similar matrices and he proves that these matrices have the same characteristic equations. He also proves that every symmetric real matrix is diagonalizable.

- 1850, Sylvester introduces for the first time the term ***matrix*** (from Latin, „uterus, womb" – a place something is formed or produced, „*an oblong arrangement of terms*")
- 1855, Cayley – matrix algebra, first abstract definition of a matrix. He studies linear transformations and their composition which leads him to the well-known matrix operations (addition, multiplication, scalar multiplication, inverse)

- 1858, Cayley in *Memoir on the theory of matrices: „there would be many things to say about this theory of matrices which should, it seems to me, precede the theory of determinants”*
- Jordan (1870 – Treatise on substitutions and algebraic equations – Jordan canonical form), Frobenius (1878 – On linear substitutions and bilinear forms, rank of a matrix)

- 1890, Weierstrass – On determinant theory, the axiomatic definition for the determinant
- 1925, Heisenberg reinvents matrix algebra for quantum mechanics
- 1947, vonNeuman & Goldstine introduce conditioning numbers when analysing rounding errors
- 1948, Turing introduces the LU decomposition of a matrix i
- 1958, Wilkinson - QR factorization …

Proposition

Let $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ for which there exists a matrix natural norm s.t. $\|\boldsymbol{A}\| < \mathbf{1}$. Then there exist the matrices $(\boldsymbol{I}_{\boldsymbol{n}} \pm \boldsymbol{A})^{-1}$, and the following relations hold:

$$\frac{\mathbf{1}}{\mathbf{1}+\|\boldsymbol{A}\|} \leq \left\|(\boldsymbol{I} \pm \boldsymbol{A})^{-1}\right\| \leq \frac{\mathbf{1}}{\mathbf{1}-\|\boldsymbol{A}\|}\,.$$

Error evaluation in solving linear systems

(linear systems of equations conditioning)

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{b} \in \mathbb{R}^{n}$, $\boldsymbol{x} \in \mathbb{R}^{n}$ defining the linear system of equations:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$\boldsymbol{A} \textbf{ non-singular } \Leftrightarrow \textbf{ det}\, \boldsymbol{A} \neq \boldsymbol{0} \Rightarrow \exists \textbf{ solution } \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

We consider the following errors in data input:

- $\Delta \boldsymbol{A} \in \mathbb{R}^{n\times n}$ **absolute error for** $\boldsymbol{A}$;
- $\Delta \boldsymbol{b} \in \mathbb{R}^{n}$ **absolute error for** $\boldsymbol{b}$;

Instead of solving the linear system $\boldsymbol{Ax=b}$, the following system is solved:

$$\left(\boldsymbol{A}+\Delta\boldsymbol{A}\right)\tilde{\boldsymbol{x}}=\boldsymbol{b}+\Delta\boldsymbol{b}$$

one obtains the solution $\tilde{\boldsymbol{x}}$:

$$\tilde{\boldsymbol{x}}=\boldsymbol{x}+\Delta\boldsymbol{x}$$

The following problems must be addressed:

1. $\boldsymbol{A}$ non-singular, $\Delta\boldsymbol{A}=\boldsymbol{?}$ s.t. $\boldsymbol{A}+\Delta\boldsymbol{A}$ is also non-singular?
2. Assume $\boldsymbol{A}$ and $\boldsymbol{A}+\Delta\boldsymbol{A}$ non-singular, which are the relations between the relative errors $\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|},\frac{\|\Delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}$ and $\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$?

1. Assume $\boldsymbol{A}$ non-singular:

$$A + \Delta A = A\left(I_n + A^{-1}\Delta A\right) \to$$

$$A + \Delta A \textbf{ non-singular } \Leftrightarrow \left(I_n + A^{-1}\Delta A\right) \textbf{ non-singular}$$

Proposition 1

Let $\boldsymbol{A}$ be non-singular and $\left\|\Delta A\right\| < \dfrac{1}{\left\|A^{-1}\right\|}$. Then $\boldsymbol{I+A^{-1}\Delta A}$ is

non-singular and the following relation holds:

$$\left\|\left(I_n + A^{-1}\Delta A\right)^{-1}\right\| \le \frac{1}{1 - \left\|A^{-1}\right\| \cdot \left\|\Delta A\right\|}.$$

Proof. We have:

$$\|\Delta A\| < \frac{1}{\|A^{-1}\|} \Rightarrow \|A^{-1}\Delta A\| \le \|A^{-1}\| \cdot \|\Delta A\| < 1 \overset{\text{Prop}}{\Rightarrow} \exists \left(I + A^{-1}\Delta A\right)^{-1}$$

$$\left\|\left(I + A^{-1}\Delta A\right)^{-1}\right\| \le \frac{1}{1 - \|A^{-1}\Delta A\|} \le \frac{1}{1 - \|A^{-1}\| \cdot \|\Delta A\|}.$$

Assume A non-singular and $\|\Delta A\| < \dfrac{1}{\|A^{-1}\|}$.

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\|A^{-1}\|\|A\|}{1 - \|A^{-1}\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right).$$

$$(A+\Delta A)(x+\Delta x)=b+\Delta b \Rightarrow (A+\Delta A)\Delta x + Ax + (\Delta A)x = b+\Delta b \Rightarrow$$

$$A\left(I+A^{-1}\Delta A\right)\Delta x = \Delta b - (\Delta A)x \Rightarrow \Delta x = \left(I+A^{-1}\Delta A\right)^{-1} A^{-1}\left[\Delta b - (\Delta A)x\right] \Rightarrow$$

$$\|\Delta x\| \le \left\|\left(I+A^{-1}\Delta A\right)^{-1}\right\| \left\|A^{-1}\right\| \left(\|\Delta b\| + \|\Delta A\|\|x\|\right) \Rightarrow$$

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\|}{1-\left\|A^{-1}\right\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\|\right) \qquad \textbf{(1)}$$

From $Ax=b$ we get $\|b\| \le \|A\|\|x\| \Rightarrow \dfrac{1}{\|x\|} \le \dfrac{\|A\|}{\|b\|}$, using this relation in (1) we have:

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\|\|A\|}{1-\left\|A^{-1}\right\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right).$$

k(A) **= ||*A*$^{-1}$|| ||*A*||** ***conditioning number*** for matrix ***A***.

Proposition 2

If matrix ***A*** is non-singular and $\left\|\Delta A\right\| < \dfrac{1}{\left\|A^{-1}\right\|}$ then:

$$\frac{\left\|\Delta x\right\|}{\left\|x\right\|} \leq \frac{k(A)}{1 - k(A)\cdot\dfrac{\left\|\Delta A\right\|}{\left\|A\right\|}}\left(\frac{\left\|\Delta b\right\|}{\left\|b\right\|} + \frac{\left\|\Delta A\right\|}{\left\|A\right\|}\right).$$

From $I_n = A\,A^{-1}$ one gets $1 = \left\|I_n\right\| \leq \left\|A\right\|\left\|A^{-1}\right\| = k(A).$

$k(A) \geq 1,\ \forall A$ but it depends on the natural matrix norm.

A matrix ***A*** for which the conditioning number is large is called ***ill-conditioned (k(A) ‚large')*****.**

Ax=b* with *k(A) large $\rightarrow$ $\dfrac{\|\Delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|}$ the relative error in the computed solution can be very large even if the relative errors $\dfrac{\|\Delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|}$ **and** $\dfrac{\|\Delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}$ are small.

Let $\boldsymbol{A}$ be a symmetric, non-singular matrix $\boldsymbol{A} = \boldsymbol{A}^T$, $\mathbf{det}\,\boldsymbol{A} \neq \boldsymbol{0}$.

Using the spectral norm (generated by the Euclidean vector norm) we have:

$$\|A\|_2 = \sqrt{\rho\left(A^T A\right)} = \sqrt{\rho\left(A^2\right)}$$

$$k(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2$$

A symmetric matrix $\boldsymbol{A}$ has only real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$,

$\boldsymbol{A}^2$ has eigenvalues $\lambda_1^2, \lambda_2^2, \ldots, \lambda_n^2$, $\boldsymbol{A}^{-1}$ has eigenvalues $\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots., \frac{1}{\lambda_n}$.

$$|\lambda_1| \le |\lambda_2| \le .... \le |\lambda_n| \Rightarrow \rho(A) = |\lambda_n| \quad \text{and} \quad \rho\left(A^{-1}\right) = \frac{1}{|\lambda_1|}$$

$$A = A^T \to \|A\|_2 = \rho(A) = |\lambda_n|, \|A^{-1}\|_2 = \rho\left(A^{-1}\right) = \frac{1}{|\lambda_1|},$$

$$k_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{|\lambda_n|}{|\lambda_1|}$$ ***spectral conditioning number.***

A orthogonal matrix $\to$ ***k₂(A)=1***

$$A^T A = A \cdot A^T = I_n \Rightarrow A^{-1} = A^T$$

$$\|A\|_2 = \sqrt{\rho\left(A^T A\right)} = \sqrt{\rho(I)} = 1 = \|A^T\|_2$$

$$k_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \|A\|_2 \cdot \|A^T\|_2 = 1,$$

Almost singular matrix with good conditioning number

$$A = \mathbf{diag}\,[1, 0.1, 0.1, ..., 0.1] \in \mathbb{R}^{100\times 100} \Rightarrow \mathbf{det}\, A = 1 \cdot (0.1)^{99} = 10^{-99}$$

$$\| A \|_2 = 1 \;, \; \| A^{-1} \|_2 = 10 \;\Rightarrow\; k_2(A) = \| A \|_2 \; \| A^{-1} \|_2 = 10$$

Very ill-conditioned matrix with non-zero determinant (**det *A*=1**)

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 2 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

$$\boldsymbol{A}^{-1} = \begin{pmatrix} 1 & -2 & 4 & \cdots & (-2)^{i-1} & \cdots & (-2)^{n-1} \\ 0 & 1 & -2 & \cdots & (-2)^{i-2} & \cdots & (-2)^{n-2} \\ \vdots & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

$\| A \|_\infty = \| A \|_1 = 3$,

$\| A^{-1} \|_\infty = \| A^{-1} \|_1 = 1 + 2 + \cdots + 2^{n-1} = 2^n - 1$

$n = 100 \;\Rightarrow\; k(A) = \| A \|_\infty \| A^{-1} \|_\infty = \| A \|_1 \| A^{-1} \|_1 = 3 \cdot (2^{100} - 1)$

$\det A = 1$

$$\begin{cases} x + y = 2 \\ x + 1.001y = 2 \end{cases} , \quad \begin{cases} x + y = 2 \\ x + 1.001y = 2.001 \end{cases} \quad k(A) = 4002$$

$$x = 2 ,\ y = 0 \qquad\qquad x = 1 ,\ y = 1$$

$$\begin{cases} 400x - 201y = 200 \\ -800x + 401y = -200 \end{cases} , \quad \begin{cases} 401x - 201y = 200 \\ -800x + 401y = -200 \end{cases}$$

$$x = -100 ,\ y = -200 \qquad\qquad x = 40000 ,\ y = 79800$$

$$k_2(A) = 2503 \qquad\qquad k_2(A) = 1002000$$

$$\begin{cases} 1.2969x + 0.8648\,y = 0.8642 \\ 0.2161x + 0.1441\,y = 0.1440 \end{cases} \qquad x = 2\,,\ \ y = -2 \qquad k_2(A) = 249730000$$

$$\bar{x} = 0.9911,\ \ \bar{y} = -0.4870\ \ ,$$

$$r = b - Az = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix} - \begin{pmatrix} 1.2969 & 0.8648 \\ 0.1441 & 0.1441 \end{pmatrix} \begin{pmatrix} 0.9911 \\ -0.4870 \end{pmatrix} = \begin{pmatrix} 10^{-8} \\ -10^{-8} \end{pmatrix}$$

Hilbert matrix

$$H = (h_{ij})_{i,j=1}^{n} \quad , \quad h_{ij} = \frac{1}{i+j-1} = \int_0^1 x^{i+j-2}\,dx$$

$$k_2(H_n) \approx \frac{(\sqrt{2}+1)^{4(n+1)}}{2^{\frac{15}{4}}\sqrt{\pi n}} \sim e^{3.5n}$$

n	$k_2(H_n)$	n	$k_2(H_n)$
1	1	7	$4.753 \cdot 10^{8}$
2	19.281	8	$1.526 \cdot 10^{10}$
3	$5.241 \cdot 10^{2}$	9	$4.932 \cdot 10^{11}$
4	$1.551 \cdot 10^{4}$	10	$1.602 \cdot 10^{13}$
5	$4.766 \cdot 10^{5}$	11	$5.220 \cdot 10^{14}$
6	$1.495 \cdot 10^{7}$	12	$1.678 \cdot 10^{16}$

$$H^{-1} = (g_{ij}) \quad g_{ij} = \frac{(-1)^{i+j}}{(i+j-1)} \frac{(n+i-1)!\,(n+j-1)!}{\left[(i-1)!(j-1)!\right]^2 (n-i)!(n-j)!}$$