Unit 6: Numerical Stability

Any real number can be written as: $\mu \times \beta^e$

where: β is the base (an integer > 1)

$\mu \in (-1, 1)$ is the mantissa

e is the exponent

~~For~~ For our discussion, $x = \mu \times \beta^e$ such that:

- $\beta = 2$ (computer use binary system)
- $\mu = \pm .\delta_0 \delta_1 \cdots \delta_{t-1}$ (μ has only t binary digits, $\delta_i \in \{0, 1\}$)
- $\delta_0 = 0$ iff $\mu = 0$ (the mantissa is normalized)
- $-L \leq e \leq U$

Example: Let $\beta = 2$, $t = 3$, $\mu = 0.101$, $e = 1$

$$x = \mu \times \beta^e$$
$$= .101 \times 2^1$$
$$= 1.25$$

Example of overflow

Since $\|x\|_2 = \sqrt{\sum_{i=0}^{n-1} x_i^2} \Leftrightarrow \|x\|_2 \leq \sqrt{n} \max_{i=0}^{n-1} |x_i|$

If more than one x_i is close to overflowing, the result will overflow since x_i^2 is overflowed.

The solution is to find k such that:

$$|x_k| = \max_{i=0}^{n-1} |x_i|$$

And then instead compute

$$\|x\|_2 = |x_k| \sqrt{\sum_{i=0}^{n-1} \left(\frac{x_i}{x_k}\right)^2}$$

Remark 6.2.1.3

Any real number stored in our computer is stored as nearby floating point number (element in F) either through rounding or truncation.

Error in storing real number as floating point number

The error is bounded by:

$$|\delta x| \leq \epsilon_{mach} |x|$$

where ϵ_{mach} is known as the machine epsilon or unit roundoff. When single precision floating point numbers are used $\epsilon_{mach} \approx 10^{-8}$ (eight decimal digits of accuracy). When double precision floating point number numbers are used $\epsilon_{mach} \approx 10^{-16}$

If $\check{x}$ is computed by truncating:

$$|\delta x| \leq 2^{-(t-p)} |x|$$

If $\check{x}$ is computed by rounding:

$$|\delta x| \leq 2^{-t} |x|$$

Machine epsilon (unit roundoff)

ϵ_{mach} is defined as the smallest positive floating point number x such that the floating point number that represents $1 + x$ is greater than 1

Python code to understand this

```
epsilon = 2**-52  # machine epsilon for double precision
print(1 + epsilon > 1)   # True
print(1 + (epsilon/2) > 1)   # False
```

Th 6.2.2.1

Given system with $\beta = 2$, u with t digits, truncation when storing

- Number 1 as floating point number in this system

$$0.\underbrace{10\ldots0}_{t \text{ digits}} \times 2^1$$

- What ϵ_{mach} of this system?

$$\epsilon_{mach} = 2^{-(t-1)}$$

Standard Computational Model (SCM)

$fl(x \ op \ \psi) = (x \ op \ \psi)(1 + \epsilon)$, $|\epsilon| \leq \epsilon_{mach}$ and $op \in \{+, -, *, /\}$

Remark 6.2.3.2

SCM can be interpreted as: These operations are performed exactly and it is only in storing the result that a roundoff error occurs.

Example

Consider $k = 4/3$. Our floating point system with $\beta = 2$, $t = 4$. Our real number $4/3 = 1.333\ldots$ is stored as $.1010 \times 2^1$, if $t = 4$ and truncation is deployed. This equals 1.25 in decimal, the relative error was 0.0625

$$\begin{aligned}
\hat{k} &= fl(4/3) \\
&= 1.25 \\
&= 1.333\ldots + (-0.0833\ldots) \\
&= 1.333\ldots \times \left(1 + \frac{-0.0833\ldots}{1.333\ldots}\right) \\
&= 4/3 \times (1 + (-0.0625)) \\
&= k(1 + \epsilon) \quad \text{where } |\epsilon| = 0.0625 \leq \epsilon_{mach} = 2^{-(t-1)} = 0.125
\end{aligned}$$

Alternative Computational Model (ACM)

$$fl(x \text{ op } y) = \frac{x \text{ op } y}{1+\epsilon}, \quad |\epsilon| \leq \epsilon_{mach}, \quad \text{op} \in \{+, -, *, /\}$$

Notice that the Taylor series expansion of $\frac{1}{1+\epsilon}$ is given by: $\frac{1}{1+\epsilon} = \underbrace{1 + (-\epsilon)} + \underbrace{O(\epsilon^2)}$

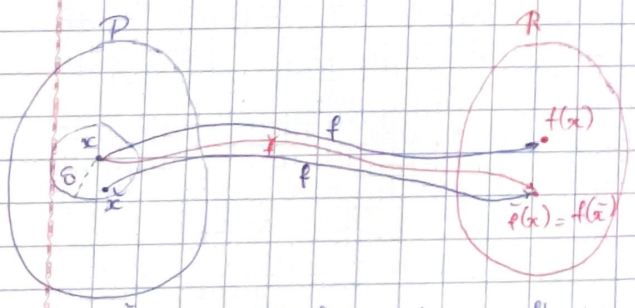
$$= (1 - \epsilon) + \epsilon^2 - \epsilon^3 + \dots$$

ACM is useful when analyzing algorithms that involve division.

Number instability

We wish to evaluate the mapping $f: D \to R$.

Let $\tilde{f}: D \to R$ is the computer implementation of f



Here, $\tilde{f}$ represent the function that use floating point arithmetic, thus incurring errors. The fact that

for a nearby value, $\tilde{x}$, the computed value equals the exact function applied to the slightly perturbed input x

$$\tilde{f}(\tilde{x}) = \tilde{f}(x)$$

means that the error in computation can be attributed to the small change in the input. It is said that

$\tilde{f}$ is a (~~more~~ numerically) stable implementation of f

Backward stable implementation

Given mapping $f: D \to R$, where $D \subset \mathbb{R}^n$, $R \subset \mathbb{R}^m$

Let $\tilde{f}$ be the computer implementation of f.

$\tilde{f}$ is backward stable (numerically stable) implementation of f if for all $x \in D$, exists $\tilde{x}$ "close" to x such that:

$$\tilde{f}(x) = f(\tilde{x})$$

Forward stable: $\tilde{f}(x) \approx f(x)$

Condition vs Stability

- Conditioning is a property of the problem. A problem is well-conditioned if a small change in the input is guaranteed to only result in a small change in output.

Stability is a property of an implementation. A numerically stable implementation when executed with an input always yields an output that can be attributed to slightly changed input.

Now,

- Well-conditioned problem + numerically stable implementation = answer that is close to the actual answer
- Else, you may not get a good answer.

Definition 6.2.6.1 Absolute value of vector & matrix

Given $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$

$$|x| = \begin{pmatrix} |\chi_0| \\ |\chi_1| \\ \vdots \\ |\chi_{n-1}| \end{pmatrix} \qquad |A| = \begin{pmatrix} |\alpha_{0,0}| & \cdots & |\alpha_{0,n-1}| \\ |\alpha_{1,0}| & & |\alpha_{1,n-1}| \\ \vdots & & \\ |\alpha_{m-1,0}| & \cdots & |\alpha_{m-1,n-1}| \end{pmatrix}$$

Definition 6.2.6.2

Let $\Delta \in \{ <, \leq, =, \geq, > \}$ and $x, y \in \mathbb{R}^n$, then:

- $|x| \; \Delta \; |y|$ iff $|\chi_i| \; \Delta \; |\psi_i|$
- $|A| \; \Delta \; |B|$ iff $|\alpha_{ij}| \; \Delta \; |\beta_{ij}|$

Hw 6.2.6.1

Let $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times n}$

<ALWAYS> $|AB| \leq |A|\,|B|$

Theorem 6.2.6.3

Let $A, B \in \mathbb{R}^{m \times n}$. If $|A| \leq |B|$ then $\|A\|_F \leq \|B\|_F$, $\|A\|_1 \leq \|B\|_1$, and $\|A\|_\infty \leq \|B\|_\infty$

Backward error analysis of dot product: general case

Lemma 6.3.2.1

Let $\epsilon_i \in \mathbb{R}$ $(0 \leq i \leq n-1)$ and $\epsilon_{mach} < 1$

$|\epsilon_i| \leq \epsilon_{mach}$. Then $\exists\ \theta_n \in \mathbb{R}$ such that:

$$\prod_{i=0}^{n-1} (1 + \epsilon_i)^{\pm 1} = 1 + \theta_n$$

with $|\theta_n| \leq \left(\frac{n}{1 - n\epsilon_{mach}} \right) \epsilon_{mach}$

Remark 6.3.2.2

The quantity θ_n is the number of error that has been accummulated. Example: $\epsilon_*^0 + \epsilon_*^1 + \epsilon_*^2 + \cdots + \epsilon_*^{n-1}$

The bound of θ_n

$$|\theta|_n \leq \frac{n\epsilon_{mach}}{1-n\epsilon_{mach}} = \gamma_n$$

For $n \geq 1$ and $n\epsilon_{mach} < 1$

With these notation, dot product simplified to:

$$\check{k} = x_0\psi_0(1+\theta_n) + x_1\psi_1(1+\theta_n) + x_2\psi_2(1+\theta_{n-1}) \cdots + x_{n-1}\psi_{n-1}(1+\theta_2)$$

$$= \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}^T \begin{pmatrix} (1+\theta_n) & & \\ & \ddots & \\ & & (1+\theta_2) \end{pmatrix} \begin{pmatrix} \psi_0 \\ \vdots \\ \psi_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}^T \left(I + \begin{pmatrix} \theta_n & & \\ & \ddots & \\ & & \theta_2 \end{pmatrix} \right) \begin{pmatrix} \psi_0 \\ \vdots \\ \psi_{n-1} \end{pmatrix}$$

$$= x^T(I + \Sigma^{(n)})y$$ Theorem 6.3.3.1

where $|\theta_j| \leq \gamma_j$, $j = 2, \ldots, n$

Lemma 6.2.3.5

If $n, b \geq 1 \Rightarrow \gamma_n < \gamma_{n+b}$

and $\gamma_n + \gamma_b + \gamma_n\gamma_b \leq \gamma_{n+b}$

This lemma is used when we want to bound $|\epsilon|$

such that $1+\epsilon = (1+\epsilon_1)(1+\epsilon_2)$

$= 1 + (\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2)$

knowing that $|\epsilon_1| \leq \gamma_n$, $|\epsilon_2| \leq \gamma_b$

Hw 6.3.3.1

Let $k \geq 0$, assume that $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{mach}$, with $\epsilon_1 = 0$ if $k = 0$

$$\begin{pmatrix} I + \Sigma^{(k)} & 0 \\ 0 & (1+\epsilon_1) \end{pmatrix} (1+\epsilon_2) = (I + \Sigma^{(k+1)})$$

Corollary 6.3.3.2

Under the assumption of Theorem 6.3.3.1, the following relations hold:

R-1B $\check{k} = (x + \delta x)^T y$, where $|\delta x| \leq \gamma_n |x|$

R-2B $\check{k} = x^T(y + \delta y)$, where $|\delta y| \leq \gamma_n |y|$

R-1F $\check{k} = x^T y + \delta k$, where $|\delta k| \leq \gamma_n |x|^T |y|$

Matrix - vector multiplication

→ next page

Giấy độ trắng tự nhiên không hại mắt
fahasa

Theorem 6.5.4.1

Let $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, consider $y = Ax$

R-1B $\check{y} = (A + \Delta A)x$ where $|\Delta A| \leq \gamma_n |A|$

R-1F $\check{y} = Ax + \delta y$ where $|\delta y| \leq \gamma_n |A||x|$

Note: $\check{y} = A(x + \delta x)$ is not possible in most cases since δx is different for every row $\tilde{a}_i^T$ of A

Matrix-matrix multiplication Theorem 6.3.5.1

Let $C \in \mathbb{R}^{m \times n}$, $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$ and consider $C = AB$, then $\exists\ \Delta C$ such that

$\check{C} = AB + \Delta C$ where $|\Delta C| \leq \gamma_k |A||B|$

Numerical stability of triangular solve

Lemma 6.4.1.2

Let $n \geq 1$, $\lambda, \nu \in \mathbb{R}$ and $x, y \in \mathbb{R}^n$. Assume $\lambda \neq 0$ and consider the computation

$\nu = (\alpha - x^T y) / \lambda$

Then $(\lambda + \delta\lambda)\nu = \alpha - (x + \delta x)^T y$,

where $|\delta x| \leq \gamma_n |x|$, $|\delta\lambda| \leq \gamma_2 |\lambda|$

Theorem 6.4.1.3 Triangular solve

Let $L \in \mathbb{R}^{n \times n}$ be nonsingular lower triangular matrix

$\check{x}$ is the computed result when executing $Lx = y$

Then $\exists$ matrix ΔL such that

$(L + \Delta L)\check{x} = y$ where $|\Delta L| \leq \max(\gamma_2, \gamma_{n-1}) \times |L|$

We can relax the bound of the above theorem abit:

$|\Delta L| \leq \gamma_n |L|$

Note that γ_n does not ~~equate~~ implies $\max(\gamma_2, \gamma_{n-1})$, especially for cases when $n = 1$

Theorem 6.4.2.1 Backward error of Crout variant for LU factorization

Let $A \in \mathbb{R}^{m \times n}$ and let the LU factorization of A be computed via the Crout variant, yielding approx factors $\check{L}$ and $\check{U}$. Then:

$(A + \Delta A) = \check{L}\check{U}$ with $|\Delta A| \leq \gamma_n |\check{L}||\check{U}|$

Numerical stability of linear solve via LU factorizati

From what we have known:

- $(A + \Delta A) = \check{L}\check{U}$, $|\Delta A| \leq \gamma_n |\check{L}||\check{U}|$
- $(L + \Delta L)\check{z} = y$, $|\Delta L| \leq \gamma_n |L|$
- $(U + \Delta U)\check{x} = z$, $|\Delta U| \leq \gamma_n |U|$

$(A + \Delta A)\check{x} = y$, $|\Delta A| \leq (3\gamma_n + \gamma_n^2)|\check{L}||\check{U}|$

Note: ΔA from solving Axy is different from ΔA from LU factorization

Thm 6.4.3.1

Consider $Ax = y$ and $(A + \Delta A)(x + \delta x) = y$

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)\frac{\|\Delta A\|}{\|A\|}}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}}$$

Theorem 6.4.3.2

Let A be nonsingular, $\|\cdot\|$ be a subordinate norm, and

$$\frac{\|\Delta A\|}{\|A\|} < \frac{1}{\kappa(A)}$$

Then $A + \Delta A$ is nonsingular

How partial pivoting is important for numerical stability

- When $A = LU$ without pivoting, we might encounter multiplier that is "small" $\Rightarrow |L|$ and $|U|$ very large $\Rightarrow |\Delta A| \leq (3\gamma_n + \gamma_n^2)|\check{L}||\check{U}|$ very very large
- When $PA = LU$ with pivoting, we pivot the largest multiplier on top and divide by it $\Rightarrow |L|$ and $|U|$ have elements $\leq 1 \Rightarrow |\Delta A| \leq (3\gamma_n + \gamma_n^2)|\check{L}||\check{U}|$ is more reasonable

$P(A + \Delta A) = \check{L}\check{U}$, where $P|\Delta A| \leq \gamma_n |\check{L}||\check{U}|$

Hw 6.4.5.2

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & & & \vdots \\ -1 & & \ddots & & \vdots \\ -1 & & & 1 & 1 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix}$$

LU-factorize A, we have:

$$U = \begin{pmatrix} 1 & \cdots & & 1 \\ 0 & 1 & & 2 \\ \vdots & \ddots & & \vdots \\ & & & 2^{n-2} \\ 0 & 0 & & 2^{n-1} \end{pmatrix}$$

Therefore, in theory even with partial pivoting, U result can get very large.

However, in practice, this does not happen and LU factorization (especially with pivoting) is stable.