Lecture 09
Accuracy and Stability

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Outline

1. Floating Point Arithmetic
2. Accuracy and Stability
3. Stability of Algorithms

Outline

Condition Number of Matrix-Vector Product

- Consider $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, with $\mathbf{A} \in \mathbb{C}^{m \times n}$

$$\kappa = \frac{\|\mathbf{J}\|}{\|\mathbf{f}(\mathbf{x})\|/\|\mathbf{x}\|} = \frac{\|\mathbf{A}\|\|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|}$$

- If $\mathbf{A}$ is square and nonsingular, since $\|\mathbf{x}\|/\|\mathbf{A}\mathbf{x}\| \leqslant \|\mathbf{A}^{-1}\|$,

$$\kappa \leqslant \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

  - Question: For what $\mathbf{x}$ is equality achieved if 2-norm is used?

Condition Number of Matrix-Vector Product

- Consider $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, with $\mathbf{A} \in \mathbb{C}^{m \times n}$

$$\kappa = \frac{\|\mathbf{J}\|}{\|\mathbf{f}(\mathbf{x})\| / \|\mathbf{x}\|} = \frac{\|\mathbf{A}\| \|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|}$$

- If $\mathbf{A}$ is square and nonsingular, since $\|\mathbf{x}\| / \|\mathbf{A}\mathbf{x}\| \leqslant \|\mathbf{A}^{-1}\|$,

$$\kappa \leqslant \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

  - Question: For what $\mathbf{x}$ is equality achieved if 2-norm is used?
  - Answer: $\mathbf{x}$ is equal to right singular vector corresponding to smallest singular value of $\mathbf{A}$

Condition Number of Matrix-Vector Product

- Consider $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, with $\mathbf{A} \in \mathbb{C}^{m\times n}$

$$\kappa = \frac{\|\mathbf{J}\|}{\|\mathbf{f}(\mathbf{x})\|/\|\mathbf{x}\|} = \frac{\|\mathbf{A}\|\|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|}$$

- If $\mathbf{A}$ is square and nonsingular, since $\|\mathbf{x}\|/\|\mathbf{A}\mathbf{x}\| \leqslant \|\mathbf{A}^{-1}\|$,

$$\kappa \leqslant \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

  - Question: For what $\mathbf{x}$ is equality achieved if 2-norm is used?
  - Answer: $\mathbf{x}$ is equal to right singular vector corresponding to smallest singular value of $\mathbf{A}$
  - Question: What is condition number of $\mathbf{A}\mathbf{x}$ if $\mathbf{A}$ is singular?

Condition Number of Matrix-Vector Product

- Consider $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, with $\mathbf{A} \in \mathbb{C}^{m \times n}$

$$\kappa = \frac{\|\mathbf{J}\|}{\|\mathbf{f}(\mathbf{x})\|/\|\mathbf{x}\|} = \frac{\|\mathbf{A}\|\|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|}$$

- If $\mathbf{A}$ is square and nonsingular, since $\|\mathbf{x}\|/\|\mathbf{A}\mathbf{x}\| \leqslant \|\mathbf{A}^{-1}\|$,

$$\kappa \leqslant \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

  - Question: For what $\mathbf{x}$ is equality achieved if 2-norm is used?
  - Answer: $\mathbf{x}$ is equal to right singular vector corresponding to smallest singular value of $\mathbf{A}$
  - Question: What is condition number of $\mathbf{A}\mathbf{x}$ if $\mathbf{A}$ is singular?
  - Answer: $\leqslant \infty$ (is ∞ if $\mathbf{x} \in null(\mathbf{A})$.

- What is the condition number for $\mathbf{f}(\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}$?

Condition Number of Matrix-Vector Product

- Consider $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, with $\mathbf{A} \in \mathbb{C}^{m\times n}$

$$\kappa = \frac{\|\mathbf{J}\|}{\|\mathbf{f}(\mathbf{x})\|/\|\mathbf{x}\|} = \frac{\|\mathbf{A}\|\|\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|}$$

- If $\mathbf{A}$ is square and nonsingular, since $\|\mathbf{x}\|/\|\mathbf{A}\mathbf{x}\| \leqslant \|\mathbf{A}^{-1}\|$,

$$\kappa \leqslant \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$$

  - Question: For what $\mathbf{x}$ is equality achieved if 2-norm is used?
  - Answer: $\mathbf{x}$ is equal to right singular vector corresponding to smallest singular value of $\mathbf{A}$
  - Question: What is condition number of $\mathbf{A}\mathbf{x}$ if $\mathbf{A}$ is singular?
  - Answer: $\leqslant \infty$ (is ∞ if $\mathbf{x} \in null(\mathbf{A})$.
- What is the condition number for $\mathbf{f}(\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}$?
  - Answer: $\kappa \leqslant \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$.

Condition Number of Matrix

- We define condition number of matrix $\mathbf{A}$ as

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

- It is the upper bound of the condition number of $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ for any $\mathbf{x}$.
- Another way to interpret $\kappa(\mathbf{A})$ is

$$\kappa(\mathbf{A}) = \sup_{\delta\mathbf{x},\mathbf{x}} \frac{\|\delta\mathbf{f}\|/\|\delta\mathbf{x}\|}{\|\mathbf{f}\|/\|\mathbf{x}\|} = \frac{\sup_{\delta\mathbf{x}} \|\mathbf{A}\delta\mathbf{x}\|/\|\delta\mathbf{x}\|}{\inf_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|/\|\mathbf{x}\|}$$

- For 2-norm, $\kappa(\mathbf{A}) = \sigma_1/\sigma_n$
- Note about the distinction between the condition number of a problem (the map $\mathbf{f}(\mathbf{x})$) and the condition number of a problem instance (the evaluation of $\mathbf{f}(\mathbf{x})$ for specific $\mathbf{x}$)
- Note: condition number of a problem is a property of a problem, and is independent of its algorithm

Outline

1. Floating Point Arithmetic

2. Accuracy and Stability

3. Stability of Algorithms

Accuracy

- Roughly speaking, accuracy means that "error" is small in an asymptotic sense, say $O(\epsilon_{machine})$
- Notation $\phi(t) = O(\psi(t))$ means $\exists C$ s.t., $|\phi(t)| \leqslant C|\psi(t)|$ as t approaches 0 (or ∞)
  - Example: $\sin^2 t = O(t^2)$ as $t \to 0$.
- If ϕ depends on s and t, then $\phi(s,t) = O(\psi(t))$ means $\exists C$ s.t., $\phi(s,t) \leqslant C|\psi(t)|$ for any s as t approaches 0 (or ∞)
  - Example: $\sin^2 t \sin^2 s = O(t^2)$ as $t \to 0$.
- When we say $O(\epsilon_{machine})$, we are thinking of a series of idealized machines for which $\epsilon_{machine}$ can be arbitrarily small

More on Accuracy

- An algorithm $\tilde{\mathbf{f}}$ is accurate if relative error is in the order of machine precision, i.e.,

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| / \|\mathbf{f}(\mathbf{x})\| = O(\epsilon_{machine})$$

  i.e. $\leqslant C_1 \epsilon_{machine}$ as $\epsilon_{machine} \to 0$, where constant C_1 may depend on the condition number and the algorithm itself
- In most cases, we expect

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| / \|\mathbf{f}(\mathbf{x})\| = O(\kappa \epsilon_{machine})$$

  i.e., $\leqslant C \kappa \epsilon_{machine}$ as $\epsilon_{machine} \to 0$, where constant C should be independent of κ and value of $\mathbf{x}$ (although it may depends on the dimension of $\mathbf{x}$)
- How do we determine whether an algorithm is accurate or not?
  - It turns out to be an extremely subtle question
  - A forward error analysis (operation by operation) is often too difficult and impractical, and cannot capture dependence on condition number
  - An effective solution is backward error analysis

Outline

Stability

- We say an algorithm is stable if it gives "nearly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}})\|/\|\mathbf{f}(\tilde{\mathbf{x}})\| = O(\epsilon_{machine})$$

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- We say an algorithm is backward stable if it gives "exactly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

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

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- Is stability or backward stability stronger?

Stability

- We say an algorithm is stable if it gives "nearly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}})\| / \|\mathbf{f}(\tilde{\mathbf{x}})\| = O(\epsilon_{machine})$$

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\| / \|\mathbf{x}\| = O(\epsilon_{machine})$.
- We say an algorithm is backward stable if it gives "exactly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

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

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\| / \|\mathbf{x}\| = O(\epsilon_{machine})$.
- Is stability or backward stability stronger?
  - Backward stability is stronger.

Stability

- We say an algorithm is stable if it gives "nearly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}})\|/\|\mathbf{f}(\tilde{\mathbf{x}})\| = O(\epsilon_{machine})$$

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- We say an algorithm is backward stable if it gives "exactly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

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

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- Is stability or backward stability stronger?
  - Backward stability is stronger.
- Does (backward) stability depend on condition number of $\mathbf{f}(\mathbf{x})$?

Stability

- We say an algorithm is stable if it gives "nearly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}})\|/\|\mathbf{f}(\tilde{\mathbf{x}})\| = O(\epsilon_{machine})$$

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- We say an algorithm is backward stable if it gives "exactly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{\mathbf{f}}$ for problem $\mathbf{f}$ is stable if (for all $\mathbf{x}$)

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

  for some $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$.
- Is stability or backward stability stronger?
  - Backward stability is stronger.
- Does (backward) stability depend on condition number of $\mathbf{f}(\mathbf{x})$?
  - No.

Stability of Floating Point Arithmetic

- Backward stability of floating point operations is implied by these two floating point axioms:
  - $\forall x \in \mathbb{R}, \exists \epsilon, |\epsilon| \leqslant \epsilon_{machine}$, s.t., $fl(x) = x(1+\epsilon)$
  - For floating-point numbers x, y, $\exists \epsilon, |\epsilon| \leqslant \epsilon_{machine}$ s.t. $x \circledast y = (x * y)(1+\epsilon)$
- Example: Subtraction $f(x_1, x_2) = x_1 - x_2$ with floating-point operation $\tilde{f}(x_1, x_2) = fl(x_1) \ominus fl(x_2)$
  - Axiom 1 implies $fl(x_1) = x_1(1+\epsilon_1), fl(x_2) = x_2(1+\epsilon_2)$, for some $|\epsilon_1|, |\epsilon_2| \leqslant \epsilon_{machine}$
  - Axiom 2 implies $fl(x_1) \ominus fl(x_2) = (fl(x_1) - fl(x_2))(1+\epsilon_3)$ for some $|\epsilon_3| \leqslant \epsilon_{machine}$
  - $$\begin{aligned} fl(x_1) \ominus fl(x_2) &= (x_1(1+\epsilon_1) - x_2(1+\epsilon_2))(1+\epsilon_3) \\ &= x_1(1+\epsilon_1)(1+\epsilon_3) - x_2(1+\epsilon_2)(1+\epsilon_3) \\ &= x_1(1+\epsilon_4) - x_2(1+\epsilon_5) \end{aligned}$$

    where $|\epsilon_4|, |\epsilon_5| \leqslant 2\epsilon_{machine} + O(\epsilon_{machine}^2)$

Stability of Floating Point Arithmetic Cont'd

- Example: Inner product $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^*\mathbf{y}$ using floating-point operations $\otimes$ and $\oplus$ is backward stable
- Example: outer product $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}^*$ using floating-point operations $\otimes$ and $\oplus$ is not backward stable
- Example: $f(x) = x + 1$ is computed as $\tilde{f}(x) = fl(x) \oplus 1$ is not backward stable
- Example: $f(x, y) = x + y$ computed as $\tilde{f}(x, y) = fl(x) \oplus fl(y)$ is backward stable

Accuracy of Backward Stable Algorithm

Theorem

If a backward stable algorithm $\tilde{\mathbf{f}}$ *is used to solve a problem* $\mathbf{f}$ *with condition number* κ *using floating-point numbers satisfying the two axioms, then*

$$\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| = O(\kappa(\mathbf{x})\epsilon_{machine})$$

Proof

Backward stability means $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{f}(\tilde{\mathbf{x}})$ for $\tilde{\mathbf{x}}$ such that
$\|\tilde{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{x}\| = O(\epsilon_{machine})$
Definition of condition number gives

$$\|\mathbf{f}(\tilde{\mathbf{x}}) - \mathbf{f}(\mathbf{x})\|/\|\mathbf{f}(\mathbf{x})\| \leqslant (\kappa(\mathbf{x}) + o(1))\|\tilde{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{x}\|$$

where $o(1) \to 0$ as $\epsilon_{machine} \to 0$.
Combining the two gives desired result.