AMS526: Numerical Analysis I (Numerical Linear Algebra)

Lecture 11: Stability of Algorithms

Xiangmin Jiao

SUNY Stony Brook

October 16, 2008

Accuracy

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

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

  i.e., $\leq C_1 \epsilon_{\text{machine}}$ as $\epsilon_{\text{machine}} \to 0$, where constant C_1 may depend on the condition number and the algorithm itself

- In most cases, we expect

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

  i.e., $\leq C \kappa \epsilon_{\text{machine}}$ as $\epsilon_{\text{machine}} \to 0$, where constant C should be independent of κ and value of x (although it may depends on the dimension of x)

- How do we determine whether an algorithm is accurate or not?
  - An effective solution is *backward error analysis*

Stability

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

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

  for some x with $\|\tilde{x} - x\| / \|x\| = O(\epsilon_{\text{machine}})$

Stability

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

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

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

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

  for some $\boldsymbol{x}$ with $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\| / \|\boldsymbol{x}\| = O(\epsilon_{\text{machine}})$

Stability

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

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

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

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

  for some x with $\|\tilde{x} - x\|/\|x\| = O(\epsilon_{\text{machine}})$
- 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 $\tilde{f}$ for problem f is *stable* if (for all x)
$$\|\tilde{f}(x) - f(\tilde{x})\|/\|f(\tilde{x})\| = O(\epsilon_{\text{machine}})$$
for some x with $\|\tilde{x} - x\|/\|x\| = O(\epsilon_{\text{machine}})$
- We say an algorithm is *backward stable* if it gives "exactly the right answer to nearly the right question"
- More formally, an algorithm $\tilde{f}$ for problem f is stable if (for all x)
$$\tilde{f}(x) = f(\tilde{x})$$
for some x with $\|\tilde{x} - x\|/\|x\| = O(\epsilon_{\text{machine}})$
- Is stability or backward stability stronger?
  - Backward stability is stronger.
- Does (backward) stability depend on condition number of $f(x)$?

Stability

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

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

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

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

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

Stability of Floating Point Arithmetic

- Backward stability of floating point operations is implied by these two floating point axioms:
  1. $\forall x \in \mathbb{R}$, $\exists \epsilon$, $|\epsilon| \leq \epsilon_{\text{machine}}$ s.t. $\text{fl}(x) = x(1+\epsilon)$
  2. For floating-point numbers x, y, $\exists \epsilon$, $|\epsilon| \leq \epsilon_{\text{machine}}$ s.t. $x \circledast y = (x * y)(1+\epsilon)$
- Example: Subtraction $f(x_1, x_2) = x_1 - x_2$ with floating-point operation

$$\tilde{f}(x_1, x_2) = \text{fl}(x_1) \ominus \text{fl}(x_2)$$

  - Axiom 1 implies $\text{fl}(x_1) = x_1(1+\epsilon_1)$, $\text{fl}(x_2) = x_2(1+\epsilon_2)$, for some $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{\text{machine}}$
  - Axiom 2 implies $\text{fl}(x_1) \ominus \text{fl}(x_2) = (\text{fl}(x_1) - \text{fl}(x_2))(1+\epsilon_3)$ for some $|\epsilon_3| \leq \epsilon_{\text{machine}}$
  - Therefore,

$$\begin{aligned} \text{fl}(x_1) \ominus \text{fl}(x_2) &= (x_1(1+\epsilon_1) - x_2(1+\epsilon_2))(1+\epsilon_3) \\ &= x_1(1+\epsilon_1)(1+\epsilon_3) - x_2(1+\epsilon_2)(1+\epsilon_3) \\ &= x_1(1+\epsilon_4) - x_2(1+\epsilon_5) \end{aligned}$$

  where $|\epsilon_4|, |\epsilon_5| \leq 2\epsilon_{\text{machine}} + O(\epsilon^2_{\text{machine}})$

Stability of Floating Point Arithmetic Cont'd

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

Accuracy of Backward Stable Algorithm

Theorem

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

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

Accuracy of Backward Stable Algorithm

Theorem

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

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

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

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

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