Lecture 9
Conditioning and Stability I

MIT 18.335J / 6.337J

Introduction to Numerical Methods

Per-Olof Persson (persson@mit.edu)

October 10, 2007

Conditioning

- *Absolute Condition Number* of a differentiable problem f at x:

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} = \|J(x)\|$$

  where the Jacobian $J_{ij} = \partial f_i / \partial x_j$, and the matrix norm is induced by the norms on δf and δx

- *Relative Condition Number*

$$\kappa = \sup_{\delta x} \left(\frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right) = \frac{\|J(x)\|}{\|f(x)\|/\|x\|}$$

Conditioning

- **Example**: The function $f(x) = \alpha x$
  - Absolute condition number $\hat{\kappa} = \|J\| = \alpha$
  - Relative condition number $\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{\alpha}{\alpha x/x} = 1$
- **Example**: The function $f(x) = \sqrt{x}$
  - Absolute condition number $\hat{\kappa} = \|J\| = \frac{1}{2\sqrt{x}}$
  - Relative condition number $\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/(2\sqrt{x})}{\sqrt{x}/x} = \frac{1}{2}$
- **Example**: The function $f(x) = x_1 - x_2$ (with ∞-norms)
  - Absolute condition number $\hat{\kappa} = \|J\| = \|(1,-1)\| = 2$
  - Relative condition number $\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{2}{|x_1 - x_2|/\max\{|x_1|,|x_2|\}}$
  - Ill-conditioned when $x_1 \approx x_2$ (cancellation)

Condition of Matrix-Vector Product

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

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

- For A square and nonsingular, use $\|x\|/\|Ax\| \le \|A^{-1}\|$:

$$\kappa \le \|A\|\|A^{-1}\|$$

  (equality achieved for the last right singular vector $x = v_m$)
- Also the condition number for $f(b) = A^{-1}b$ (solution of linear system)
- *Condition number of matrix* A:

$$\kappa(A) = \|A\|\|A^{-1}\| = [\text{ for 2-norm }] = \frac{\sigma_1}{\sigma_m}$$

Condition of System of Equations

- For fixed b, consider $f(A) = A^{-1}b$
- Perturb A by δA and find perturbation δx:

$$(A + \delta A)(x + \delta x) = b$$

- Use $Ax = b$ and assume $(\delta A)(\delta x) \approx 0$:

$$(\delta A)x + A(\delta x) = 0 \quad \Longrightarrow \quad \delta x = -A^{-1}(\delta A)x$$

- Condition number of problem f:

$$\kappa = \frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta A\|}{\|A\|} \leq \frac{\|A^{-1}\|\|\delta A\|\|x\|}{\|x\|} \Big/ \frac{\|\delta A\|}{\|A\|} = \|A^{-1}\|\|A\| = \kappa(A)$$

Accuracy

- Consider an *algorithm* $\tilde{f}$ for a *problem* f
- A computation $\tilde{f}(x)$ has *absolute error* $\|\tilde{f}(x) - f(x)\|$ and *relative error*

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

- The algorithm is *accurate* if (for all x)

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

where $O(\epsilon_{\text{machine}})$ is “on the order of $\epsilon_{\text{machine}}$” (more next slide)

- Constant in $O(\epsilon_{\text{machine}})$ is likely to be large in many problems, since because of rounding we are not even using the correct x

More on $O(\epsilon_{\text{machine}})$

- The notation $\varphi(t) = O(\psi(t))$ means there is a constant C such that, for t close to a limit (often 0 or ∞), $|\varphi(t)| \leq C\psi(t)$
- **Example**: $\sin^2 t = O(t^2)$ as $t \to 0$ means $|\sin^2 t| \leq Ct^2$ for some C
- If φ depends on additional variables, the notation

$$\varphi(s,t) = O(\psi(t)) \qquad \text{uniformly in } s$$

  means there is a constant C such that $|\varphi(s,t)| \leq C\psi(t)$ for any s
- **Example**: $(\sin^2 t)(\sin^2 s) = O(t^2)$ uniformly as $t \to 0$, but not if $\sin^2 s$ is replaced by s^2
- In bounds such as $\|\tilde{x} - x\| \leq C\kappa(A)\epsilon_{\text{machine}}\|x\|$, C does not depend on A or b, but it might depend on the dimension m

Stability

- An algorithm $\tilde{f}$ for a problem f is *stable* if (for all x)

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

  for some $\tilde{x}$ with

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

- "Nearly the right answer to nearly the right question"
- An algorithm $\tilde{f}$ for a problem f is *backward stable* if (for all x)

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

- "Exactly the right answer to nearly the right question"

Stability of Floating Point Arithmetic

- The two floating point axioms imply backward stability for the operations $\circledast$

  (1) For all $x \in \mathbb{R}$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{machine}}$ such that
  $\text{fl}(x) = x(1+\epsilon)$

  (2) For all floating point x, y, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{machine}}$ such that
  $x \circledast y = (x * y)(1+\epsilon)$

- **Example**: Subtraction $f(x_1, x_2) = x_1 - x_2$ with floating point algorithm

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

- (1) implies

$$\text{fl}(x_1) = x_1(1+\epsilon_1), \qquad \text{fl}(x_2) = x_2(1+\epsilon_2)$$

  for some $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{\text{machine}}$

Stability of Floating Point Arithmetic

(example continued)

- (2) implies

$$\text{fl}(x_1) \ominus \text{fl}(x_2) = (\text{fl}(x_1) - \text{fl}(x_2))(1+\epsilon_3)$$

  for some $|\epsilon_3| \leq \epsilon_{\text{machine}}$

- Combine:

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

  for some $|\epsilon_4|, |\epsilon_5| \leq 2\epsilon_{\text{machine}} + O(\epsilon_{\text{machine}}^2)$

- Therefore, $\text{fl}(x_1) \ominus \text{fl}(x_2) = \tilde{x}_1 - \tilde{x}_2$

Stability of Floating Point Arithmetic

- **Example**: Inner product $f(x,y) = x^*y$ computed with $\otimes$ and $\oplus$ is backward stable (more later)
- **Example**: Outer product $f(x,y) = xy^*$ computed with $\otimes$ is *not* backward stable (unlikely that $\tilde{f}$ is rank-1)
- **Example**: $f(x) = x + 1$ computed by $\tilde{f}(x) = \text{fl}(x) \oplus 1$ is *not* backward stable (consider $x \approx 0$)
- **Example**: $f(x,y) = x + y$ computed by $\tilde{f}(x,y) = \text{fl}(x) \oplus \text{fl}(y)$ is backward stable

Accuracy of a Backward Stable Algorithm

- If a backward stable algorithm is used to solve a problem f with condition number κ, the relative errors satisfy

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

- *Proof*. Backward stability means $\tilde{f}(x) = f(\tilde{x})$ for $\tilde{x}$ such that

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

The definition of condition number gives

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

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