Outline

Conditioning and Condition Numbers

Overview of Error Analysis

- Error analysis is important subject of numerical analysis
- Given a problem f and an algorithm $\tilde{f}$ with an input x, the absolute error is $\left\|\tilde{f}(x) - f(x)\right\|$ and relative error is $\left\|\tilde{f}(x) - f(x)\right\| / \|f(x)\|$
- What are possible sources of errors?

Overview of Error Analysis

- Error analysis is important subject of numerical analysis
- Given a problem f and an algorithm $\tilde{f}$ with an input x, the absolute error is $\left\|\tilde{f}(x) - f(x)\right\|$ and relative error is $\left\|\tilde{f}(x) - f(x)\right\| / \|f(x)\|$
- What are possible sources of errors?
  - Round-off error (input, computation), truncation (approximation) error
- We would like the solution to be accurate, i.e., with small errors
- The error depends on property (conditioning) of the problem, property (stability) of the algorithm

Absolute Condition Number

- Condition number is a measure of sensitivity of a problem
- Absolute condition number of a problem $\mathbf{f}$ at $\mathbf{x}$ is

$$\hat{\kappa} = \lim_{\varepsilon \to 0} sup_{\|\delta \mathbf{X}\| \leq \varepsilon} \frac{\|\delta \mathbf{f}\|}{\|\delta \mathbf{X}\|}$$

  where $\delta \mathbf{f} = \mathbf{f}(\mathbf{x} + \delta \mathbf{x}) - \mathbf{f}(\mathbf{x})$
- Less formally, $\hat{\kappa} = sup_{\delta \mathbf{X}} \frac{\|\delta \mathbf{f}\|}{\|\delta \mathbf{X}\|}$ for infinitesimally small $\delta \mathbf{x}$
- If $\mathbf{f}$ is differentiable, then

$$\hat{\kappa} = \|\mathbf{J}(x)\|$$

  where $\mathbf{J}$ is the Jacobian of $\mathbf{f}$ at $\mathbf{x}$, with $J_{ij} = \frac{\delta f_i}{\delta x_j}$, and the matrix norm is induced by vector norms on $\delta \mathbf{f}$ and $\delta \mathbf{x}$
- Question: What is absolute condition number of $\mathbf{f}(\mathbf{x}) = \alpha \mathbf{x}$?
- Question: Is absolute condition number scale invariant?

Relative Condition Number

- Relative condition number of a problem $\mathbf{f}$ at $\mathbf{x}$ is

$$\hat{\kappa} = \lim_{\varepsilon \to 0} sup_{\|\delta \mathbf{X}\| \leq \varepsilon} \frac{\|\delta \mathbf{f}\| / \|\mathbf{f}(\mathbf{x})\|}{\|\delta \mathbf{X}\| / \|\mathbf{x}\|}$$

- Less formally, $\hat{\kappa} = sup_{\delta \mathbf{X}} \frac{\|\delta \mathbf{f}\|/\|\mathbf{f}(\mathbf{x})\|}{\|\delta \mathbf{X}\|/\|\mathbf{x}\|}$ for infinitesimally small $\delta \mathbf{x}$
- Note: we can use different types of norms to get different condition numbers
- If $\mathbf{f}$ is differentiable, then

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

- Question: What is relative condition number of $\mathbf{f}(\mathbf{x}) = \alpha \mathbf{x}$?
- Question: Is absolute condition number scale invariant?
- In numerical analysis, we in general use relative condition number
- A problem is well-conditioned if κ is small and is ill-conditioned if κ is large

Condition Numbers

- Absolute condition number of a problem $\mathbf{f}$ at $\mathbf{x}$ is

$$\hat{\kappa} = \lim_{\varepsilon \to 0} sup_{\|\delta \mathbf{X}\| \le \varepsilon} \frac{\|\delta \mathbf{f}\|}{\|\delta \mathbf{X}\|}$$

where $\delta \mathbf{f} = \mathbf{f}(\mathbf{x} + \delta \mathbf{x}) - \mathbf{f}(\mathbf{x})$

- Less formally, $\hat{\kappa} = sup_{\delta \mathbf{X}} \frac{\|\delta \mathbf{f}\|}{\|\delta \mathbf{X}\|}$ for infinitesimally small $\delta \mathbf{x}$

- Relative condition number of a problem $\mathbf{f}$ at $\mathbf{x}$ is

$$\hat{\kappa} = \lim_{\varepsilon \to 0} sup_{\|\delta \mathbf{X}\| \le \varepsilon} \frac{\|\delta \mathbf{f}\| / \|\mathbf{f}(\mathbf{x})\|}{\|\delta \mathbf{X}\| / \|\mathbf{x}\|}$$

- Less formally, $\hat{\kappa} = sup_{\delta \mathbf{X}} \frac{\|\delta \mathbf{f}\| / \|\mathbf{f}(\mathbf{x})\|}{\|\delta \mathbf{X}\| / \|\mathbf{x}\|}$ for infinitesimally small $\delta \mathbf{x}$

Examples

- Example: Function $f(x) = \sqrt{x}$
  - Absolute condition number of $\mathbf{f}$ at $\mathbf{x}$ is $\hat{\kappa} = \|\mathbf{J}\| = 1/(2\mathbf{x})$
  - Relative condition number $\hat{\kappa} = \frac{\|\mathbf{J}(x)\|}{\|\mathbf{f(x)}\|/\|\mathbf{x}\|} = \frac{1/(2\sqrt{\mathbf{x}}}{\sqrt{\mathbf{x}}/\mathbf{x}} = 1/2$
- Example: Function $f(x) = x_1 - x_2$, where $x = (x_1, x_2)^T$
  - Absolute condition number of f at x in ∞-norm is $\hat{\kappa} = \|\mathbf{J}\|_\infty = \|(1,-1)\|_\infty = 2$
  - Relative condition number$\hat{\kappa} = \frac{\|\mathbf{J}(x)\|}{\|\mathbf{f(x)}\|/\|\mathbf{x}\|} = \frac{2}{|x_1 - x_2|/max\{|x_1|,|x_2|\}}$
  - κ is arbitrarily large (f is ill-conditioned) if $x_1 \approx x_2$ (hazard of cancellation error)
- Note: From now on, we will talk about only relative condition number

Condition Number of Matrix-Vector Product

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

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

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

$$\kappa \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$$

- 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 $\mathbf{A} \in \mathbb{C}^{m \times n}$

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

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

$$\kappa \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$$

  - 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: $\leq \infty$ (is ∞ if $\mathbf{x} \in null(\mathbf{A})$).
- What is the condition number for $f(\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}$?

Condition Number of Matrix-Vector Product

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

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

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

$$\kappa \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$$

  - 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: $\leq \infty$ (is ∞ if $\mathbf{x} \in null(\mathbf{A})$).
- What is the condition number for $f(\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}$?
  - Answer: $\kappa \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$

Condition Number of a Matrix

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

$$\kappa \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$$

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

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

- For 2-norm, $\kappa(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}$
- 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

Accuracy and Stability

Accuracy

- Roughly speaking, accuracy means that "error" is small in an asymptotic sense, say $O(\varepsilon_{machine})$
- When we say $O(\varepsilon_{machine})$, we are thinking of a series of idealized machines for which $\varepsilon_{machine}$ 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.,

$$\left\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\right\| / \|\mathbf{f}(\mathbf{x})\| = O(\varepsilon_{machine}),$$

  i.e., $\leq C_{1\varepsilon_{machine}}$ as $\varepsilon_{machine} \to 0$, where constant C_1 may depend on the condition number and the algorithm itself

- In most cases, we expect

$$\left\|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\right\| / \|\mathbf{f}(\mathbf{x})\| = O(\kappa\varepsilon_{machine}),$$

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

More on Accuracy

- 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**

Stability

Stable Algorithm

- 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 $\mathbf{x}$)

$$\left\|\tilde{\mathbf{f}}(\mathbf{x}) - f(\tilde{\mathbf{x}})\right\| / \left\|f(\tilde{\mathbf{x}})\right\| = O(\varepsilon_{machine}),$$

Stability

Stable Algorithm

- 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 $\mathbf{x}$)

$$\left\|\tilde{\mathbf{f}}(\mathbf{x}) - f(\tilde{\mathbf{x}})\right\| / \|f(\tilde{\mathbf{x}})\| = O(\varepsilon_{machine}),$$

Backward Stable Algorithm

- 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 $\|\tilde{\mathbf{x}} - \mathbf{x}\| / \|\mathbf{x}\| = O(\varepsilon_{machine})$
- Backward stability is stronger! And it does not depend on condition number of

Stability of Floating Point Arithmetic

- Backward stability of floating point operations is implied by these two floating point axioms:
  1. $\forall x \in \mathbb{R}, \exists \varepsilon, |\varepsilon| \leq \varepsilon_{machine}$ s.t. $fl(x) = x(1+\varepsilon)$
  2. For floating-point numbers $x, y, \exists \varepsilon, |\varepsilon| \leq \varepsilon_{machine}$ s.t. $x \circledast y = (x * y)(1+\varepsilon)$

Stability of Floating Point Arithmetic

- 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 $fl(x_1) = x_1(1+\varepsilon_1), fl(x_2) = x_2(1+\varepsilon_2)$, for some $|\varepsilon_1|, |\varepsilon_2| \leq \varepsilon_{machine}$
- Axiom 2 implies $fl(x_1) \ominus fl(x_2) = (fl(x_1) - fl(x_2))(1+\varepsilon_3)$ for some $|\varepsilon_3| \leq \varepsilon_{machine}$
- Therefore,

$$\begin{aligned} fl(x_1) \ominus fl(x_2) &= (x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2))(1+\varepsilon_3) \\ &= x_1(1+\varepsilon_1)(1+\varepsilon_3) - x_2(1+\varepsilon_2)(1+\varepsilon_3) \\ &= x_1(1+\varepsilon_4) - x_2(1+\varepsilon_5) \end{aligned}$$

where $|\varepsilon_4|, |\varepsilon_5| \leq 2\varepsilon_{machine} + O(\varepsilon_{machine}^2)$

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 $\otimes$ and $\oplus$ is not backward stable
- Example: $f(x) = x + 1$ computed as $\tilde{\mathbf{f}}(x) = fl(x) \oplus 1$ 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 f with condition number κ using floating-point numbers satisfying the two axioms, then

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

Proof: Not Included - Compensation against online classes!

Backward Stability of Householder QR

- For a QR factorization $\mathbf{A} = \mathbf{QR}$ computed by Householder triangularization, the factors $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$ satisfy

$$\tilde{\mathbf{Q}}\tilde{\mathbf{R}} = \mathbf{A} + \delta\mathbf{A}, \qquad \|\delta\mathbf{A}\| \,/\, \|\mathbf{A}\| = O(\varepsilon_{machine}),$$

  i.e. exact QR factorization of a slightly perturbed $\mathbf{A}$ (we will not prove it in class)
- $\tilde{\mathbf{R}}$ is $\mathbf{R}$ computed by algorithm using floating points
- However, $\tilde{\mathbf{Q}}$ is product of exactly unitary reflectors

$$\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}}_1\tilde{\mathbf{Q}}_2\cdots\tilde{\mathbf{Q}}_n$$

  where $\tilde{\mathbf{Q}}_k$ is given by computed $\tilde{\mathbf{v}}_k$, since $\mathbf{Q}$ is not formed explicitly