

Numerical Algorithms

IE 496 Lecture 20

Reading for This Lecture

- Primary
 - Miller and Boxer, Pages 124-128
 - Forsythe and Mohler, Sections 1 and 2

Numerical Algorithms

Numerical Analysis

- So far, we have looked primarily at algorithms for discrete problems.
- Now we will consider problems from continuous mathematics.
- *Numerical analysis* is the study of algorithms for these problems.
- The main difference between the two areas is that in continuous mathematics, numbers must be approximated in general.

Problems and Algorithms

- A *problem* is a map from $f: X \rightarrow Y$, where X and Y are normed vector spaces.
- A *numerical algorithm* is a procedure which calculates $F(x) \in Y$, an approximation of $f(x)$.
- A numerical algorithm does not necessarily have to be finite.
- Some algorithms converge (hopefully quickly) to the true solution "in the limit".

Conditioning

- A problem is *well-conditioned* if $x' \approx x \Rightarrow f(x') \approx f(x)$.
- Otherwise, it is *ill-conditioned*.
- Notice that well-conditioned requires *all* small perturbations to have a small effect.
- Ill-conditioned only requires *some* small perturbation to have a large effect.
- Condition number of a problem
 - Absolute
 - Relative

Stability

- An algorithm is *stable* if $F(x) \approx f(x')$ for some $x' \approx x$.
- This says that a stable algorithm computes "nearly the right answer" to "nearly the right question".
- Notice the contrast between conditioning and stability:
 - Conditioning applies to problems.
 - Stability applies to algorithms.

Accuracy

- *Stability plus good conditioning implies accuracy.*
- If a stable algorithm is applied to a well-conditioned problem, then $F(x) \approx f(x)$.
- Conversely, if a problem is ill-conditioned, an accurate solution may not be possible or even meaningful.
- We cannot ask more of an algorithm than stability.

Examples

- Addition, subtraction, multiplication, division.
 - Addition, multiplication, division with positive numbers are well-conditioned problems.
 - Subtraction is not.
- Zeros of a quadratic equation
 - The problem of computing the two roots is well-conditioned.
 - However, the quadratic formula is not a stable algorithm.
- Solving systems of linear equations $Ax = b$.
 - Conditioning depends on the matrix A .

Floating-point Arithmetic

- The floating-point numbers F are a subset of the real numbers.
- For a real number x , let $fl(x) \in F$ denote the floating point approximation to x .
- Let \odot and \cdot represent the four floating point and exact arithmetic operations.
- Typically, there is a number $u \ll 1$ called machine epsilon, such that
 - $fl(x) = x(1 + \varepsilon)$ for some ε with $|\varepsilon| \leq u$.
 - $\forall a, b \in F, a \odot b = (a \cdot b)(1 + \varepsilon)$ for some ε with $|\varepsilon| \leq u$.

Stability of Floating Point Arithmetic

- Floating point arithmetic is stable for computing sums, products, quotients, and differences of two numbers.
- Sequences of these operations can be unstable however.
- Example
 - Assume 10 digit precision
 - $(10^{-10} + 1) - 1 = 0$
 - $10^{-10} + (1 - 1) = 10^{-10}$
- Floating point operations are not always associative.

More Bad Example

- Calculating e^{-a} with $a > 0$ by Taylor Series.
 - The round-off error is approximately u times the largest partial sum.
 - Calculating e^a and then taking its inverse gives a full-precision answer
- Roots of a quadratic ($ax^2 + bx + c$)
 - If $x_1 \approx 0$ and $x_2 \gg 0$, then the quadratic formula is unstable.
 - Computing x_2 by the quadratic formula and then setting $x_1 = cx_2 / a$ is stable.

Backward Error Analysis

- Backward error analysis is a method of analyzing round-off error and assessing stability.
- We want to show that the result of a floating-point operation has the same effect as if the original data had been perturbed by an amount in $O(u)$.
- If we can show this, then the algorithm is stable.

More examples

- Matrix factorization
 - Generally ill-conditioned.
 - There are stable algorithms, however.
- Zeros of a polynomial
 - Generally ill-conditioned.
- Eigenvalues of a matrix
 - For a symmetric matrix, finding eigenvalues is well-conditioned, finding eigenvectors is ill-conditioned.
 - For non-symmetric matrices, both are ill-conditioned.
 - In all cases, there are stable algorithms.