Fundamental Concepts in Computational and Applied Mathematics

Juan Meza
School of Natural Sciences
University of California, Merced

Fall 2014

Recap

- Beware catastrophic cancellation – know the limits of precision
- Learn about the conditioning of your problem
- Choose algorithms known to be stable so as to not introduce any more loss of precision than necessary
    - Conditioning is fundamentally a characteristic of the problem while
    - Stability is related to algorithms

Informally

Conditioning

A problem is well conditioned if small changes in the inputs lead to small changes in the outputs.

Stability

An algorithm is stable if it solves a "nearby" problem exactly.

Claim: We cannot solve anything numerically except linear systems

- Usually start by taking a continuous problem and discretizing
- Next step is to linearize the problem (if it already isn't linear)
- Final step involves solving some form of $Ax = b$

N.B.

These days, there is usually another step where the algorithm is parallelized

Basic Linear Algebra methods

- Suppose $x, y \in R^N$ and $A \in R^{N \times N}$
- Familiar linear algebra operations
  - dot products: $x^T y$
  - matvec: Ax
  - saxpy: $ax + y$
  - Solve $Ax = b$
  - Solve $Ax = \lambda x$
- These operations have been immortalized into the BLAS and LAPACK libraries

Methods for solving $Ax = b$

- Various standard methods for solving systems of linear equations
  - LU
  - Cholesky
  - QR
  - SVD
- All have advantages and disadvantages – learn about them!
- Your choice of method will depend on the application, software availability, and time ($$) constraints

Warning

Never (EVER) solve a linear system by calculating the inverse of the matrix and multiplying!

Some Useful Terms to Know

Residual

The residual of linear system is given by $b - Ax$.

Machine Precision

Machine precision is denoted by μ. Modern computers have $\mu \approx 10^{-16}$.

Condition Number

The condition number of a matrix A is given by: $\kappa(A) = ||A|| \cdot ||A^{-1}||$.

Ill-conditioning

A problem is said to be ill-conditioned if $\mu \cdot \kappa(A) \approx 1$.

Fun Facts

- Gaussian elimination always produces solutions with relatively small residuals (minor caveats).
- Best possible error bound[1] for solving a system of linear equations can be given by

$$\frac{||x - \hat{x}||_\infty}{||x||_\infty} \leq 4\mu\kappa_\infty(A), \tag{1}$$

where μ is machine precision of the computer.

Discussion

If the residual of a computed solution is small, can I say that I have an "good" answer? Can I trust it?

[1] Golub and van Loan, Matrix Computations 3rd ed., pp. 105

Stable or Unstable? Well conditioned or ill-conditioned?

Consider $Ax = b$, where

$$A = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix}, b = [1, 1]^T.$$

Discussion

What do you think that Gaussian Elimination will do with this?
Is the problem well conditioned or ill-conditioned?

LU with Partial Pivoting (LU/PP)

- Can show that LU/PP generates the exact solution to a perturbed problem $(A + E)x = b$, such that

$$||E||_\infty \leq 8n^3 \rho ||A||_\infty \mu.$$

- *In theory*, the growth factor ρ can grow exponentially, *but in practice* it is usually of order 10.
- Consider the residual, $b - Ax$:

$$\begin{aligned} \|b - Ax\|_\infty &= \|Ex\|_\infty \\ &\leq 8n^3 \rho \|A\|_\infty \mu \|x\|_\infty \\ &\approx \mu \|A\|_\infty \|x\|_\infty. \end{aligned}$$

Comparison of some familiar algorithms for $Ax = b$

Table: Familiar $Ax = b$ solution methods

Algorithm	Work	Advantages	Disadvantages
LU	$\mathcal{O}(n^3)$	simple	unstable
LU w/PP	$\mathcal{O}(n^3)$	(usually) stable	growth factor; pivoting will change str
LU w/FP	$\mathcal{O}(n^3)$	stable	more work than PP
QR	$\mathcal{O}(n^3)$	stable,no growth	2x work of LU
SVD	$\mathcal{O}(n^3)$	stable	more than 2x work

Summary

- Basic Linear Algebra Subroutines (BLAS) are at the core of many modern computer simulations.
- Solution of linear systems is essential to your knowledge of computational mathematics.
- Small residuals are not enough to show you have a good solution – need to do a deeper analysis.

References

- **Matrix Computations, 3rd Ed.**, Gene H. Golub and Charles F. Van Loan, Johns Hopkins, 1996.