IE 495 Lecture 23

November 21, 2000

Reading for This Lecture

- Primary
  - Miller and Boxer, Pages 128-134
  - Forsythe and Mohler, Sections 9-13

Parallel Gaussian Elimination

- PRAM with n^2 processors

- Mesh with n^2 processors

Scaling

- In the "bad" example from the last lecture, what caused the trouble?
- Essentially, coefficients were too far apart in "scale".
- Ex: $10^5 + 10^{-5} = 10^5$ if $d = 5$.
- What can we do about this?

Diagonal Equivalence

- Two matrices A and A' are diagonally equivalent if
  - $A' = D_1^{-1} A D_2$
  - D_1 and D_2 are non-singular diagonal matrices
- A' is just A with the columns and rows "scaled".
- For our purposes, the elements of D_1 and D_2 will be powers of 10 (we assume this base).
- Hence, this operation merely changes the exponent.
- This operation does not change the "significands".

Computing with Scaled Matrices

- Notice that "diagonal equivalence" is an equivalence relation.
- Suppose we set $b' = D_1 b$ (similarly scaled)
  - If the same sequence of pivots is used,
  - The solutions to the these systems will have the same significands:
    - $A'x' = b'$
    - $Ax = b$
- They will differ only in their exponents.

What is the point?

- We can now see that scaling only alters the choice of pivot element.
- However, we can use scaling to change the condition number of the matrix.
- The problem of finding a scaling the minimizes the condition number of the system is difficult.
- It has been solved for certain norms, but not L_2.

Another approach

- A matrix is said to be *row equilibrated* if the maximum entry in each row is between 10^{-1} and 1.
- *Column equilibrated* is defined similarly.
- A matrix is *equilibrated* if it is both *row and column equilibrated*.
- It is unknown how to "optimally" equilibrate a matrix.
- There are heuristics for doing so approximately.
- This seems to be a good approach.

Iterative Improvement

- Iterative Procedure
  - Solve $Ax_1 = b$.
  - Compute the *residual* $r_1 = Ax_1 - b$.
  - Solve the system $Az_1 = r_1$.
  - Set $x_2 = A(x_1 + z_1)$.
- Note that r_i must be computed with more precision than the rest of the computation.

Convergence of Iterative Improvement

- The error in x_1 is related to r_1 by

$$e_1 = x_1 - A^{-1}b = A^{-1}(Ax_1 - b) = -A^{-1}r_1.$$

- Hence, $norm(e_1) \leq norm(A^{-1}) \cdot norm(r_1)$.
- Also, $norm(r_1) \approx 10^{-t}\, norm(A) \cdot norm(x_1)$.
- So finally, $norm(e_1) \approx 10^{-t} cond(A) \cdot norm(x_1)$
- If $cond(A) \approx 10^{p}$, $norm(e_1)/norm(x_1) \approx 10^{t-p}$.

Consequences

- With some care, we can assure that $norm(z_1)/norm(x_1) \approx norm(e_1)/norm(x_1) \approx 10^{t-p}$.
- Hence, $cond(A) \approx 10^t norm(z_1)/norm(x_1)$.
- Furthermore, the number of iterations needed to compute to t digits of precision is $t/(\log_{10}(norm(z_1)/norm(x_1)))$.
- If $p \geq t$, we're out of luck.

Sparsity

- Sparse matrices allow faster calculation.
- If A is sparse, we attempt to maintain that sparsity is the LU factorization.
- Markowitz's Rule
  - Let p_i be the number of nonzeros in row i and q_j the number of nonzeros in column j.
  - Pivot on the element a_{ij} such that $(p_i - 1)(q_j - 1)$ is minimized.
- Note that this is at odds with pivoting rules to limit round-off error.

Another Procedure

- Note that if *A* has no nonzeros above the diagonal in column j, then this pattern is carried into *L* and *U*.
- Hence, we try to make *A* look as much like a lower diagonal matrix as possible through premutation.
- This has good results in practice, but also must be traded off against round-off error.

A Word About Zero Tolerances

- The number zero plays a central role in these issues.
- Numbers that are very close to zero tend to cause numerical difficulties.
- Values that appear nonzero because of round-off, but whose *true value* is zero are especially dangerous.
- For this reason, practitioners usually use zero tolerances.
- This is a limit below which a value is taken to be exactly zero.
- Usually, there are several different tolerances.
- Choosing them is problematic.

Summary

- Limiting round-off error is an inexact science.
- There is some theory to guide us, but techniques based on the theory don't always work.
- You have to know your problem!
- Always remember the difference between conditioning and stability!
- Formulation can make a big difference to conditioning!!
- Changing the algorithm can improve stability.