Lecture 32

Overview of Iterative Methods

NLA Reading Group Spring'13
by Beyza Ermis

Why Iterate?

- OBJECTIVE: To present the methods that dominate large-scale computation.
- Non-iterative or "direct" algorithms require $O(m^3)$ calculations for general matrices
- This is too large because
  - large m implies very large m^3
  - the work required is of a higher order than the order of the input, which is $O(m^2)$

Why Iterate?

- Here is a brief history of what dimensions have been considered "very large" in matrix computation:
  - 1950: $m = 20$ (Wilkinson)
  - 1965: $m = 200$ (Forsythe and Moler)
  - 1980: $m = 2000$ (LINPACK)
  - 1995: $m = 20000$ (LAPACK)
- This means that during this forty-five year period, while the speed of computers increased by a factor of 10^9, the dimensions of tractable matrix increased only by a factor of 10^3.
- This leads to the conclusion that $O(m^3)$ is a bottleneck for direct matrix algorithms.
- Reducing the computation time for matrices from $O(m^3)$ to $O(m^2)$ is the goal of the iterative methods.

Structure, Sparsity and Black Boxes

Structure:

- $O(m^3)$ cannot be beaten in every "random" case.
- Fortunately, in practice large matrices are far from random.
- Large matrices are often the result of discretization of differential or integral equations.
- Discretization is generally the approximation of infinite-dimensional (continuous) quantities by finite-dimensional quantities.
- In this sense, large m values mean to approximate ∞.
- These approximations are typically *structured*, and this *structure* can often be exploited

Structure, Sparsity and Black Boxes

Sparsity:

- The most common structure that may be encountered is *sparsity*
- *Sparsity*: a preponderance of zero entries
- For example, a finite-difference discretization of partial differential equation may lead to a matrix of dimensions $m = 10^5$ with only $\nu = 10$ entries per row.
- This kind of structure is exploited by *iterative methods*.

Structure, Sparsity and Black Boxes

Black Box:

$$x \longrightarrow \boxed{\begin{array}{c}\text{BLACK}\\ \text{BOX}\end{array}} \longrightarrow Ax.$$

- Sparsity is exploited by iterative methods in that they treat matrix multiplication as a *black box* with input x and output Ax.
- Iterative algorithms typically only require the ability to determine Ax for any given x; the details are not important and indeed may not be available.
- For the example of a sparse matrix A described above, it is easy to design a procedure to compute Ax in O(mν) rather than O(m^3).

Projection into Krylov Subspaces

- The iterative methods presented in the remaining lectures are based on the idea of reducing an *m*-dimensional problem to a lower-dimensional Krylov subspace.

- Given A and b, the associated Krylov sequence of vectors is:
  - b, Ab, A^2 b, A^3 b, ...,
  - which can be computed by the black box as
  - b, Ab, A(Ab), A(A(Ab)), ...

- The corresponding Krylov subspaces are the spaces spanned by the successively larger groups of these vectors.

Projection into Krylov Subspaces

	$Ax = b$	$Ax = \lambda x$
$A = A^*$	CG	Lanczos
$A \neq A^*$	GMRES CGN BCG et al.	Arnoldi

- where, e.g., CG stands for the conjugate gradient method and requires the matrix to be symmetric positive definite.
- In each case, the strategy is to reduce the original matrix problem to a sequence of problems of dimension n = 1,2,3,....
- When A is hermitian, the reduced matrices are tridiagonal; otherwise they are Hessenberg.
- The Arnoldi iteration, for example, approximates the eigenvalues of a large matrix by computing the eigenvalues of certain Hessenberg matrices of successively larger dimensions.

Number of Steps, Work Per Step

- Gaussian elimination, QR factorization, and other dense matrix algorithms fit the pattern
  - $O(m)$ steps
  - $O(m^2)$ operations at each step
  - $O(m^3)$ total operations.
- For iterative methods, the same figures apply but they represent a worst-case scenario.
- Iterative methods succeed when at least one of these two factors is reduced.

Number of Steps, Work Per Step

- The number of steps required for satisfactory convergence depends on spectral properties of the matrix A.
- For example, CG is guaranteed to solve an spd system $Ax = b$ quickly if the eigenvalues of A are clustered well away from the origin.
- Lanczos iteration is guaranteed to compute certain eigenvalues of a real symmetric matrix quickly if those eigenvalues are well separated from the rest of the spectrum.

Number of Steps, Work Per Step

- The work done in an iteration depends on
  - the structure of the matrix
  - advantages of the structure in the $x \rightarrow Ax$ black-box

- Ideally, iterative methods reduce the number of steps from $O(m)$ to $O(1)$ and the work per step from $O(m^2)$ to $O(m)$, thus reducing the total work from $O(m^3)$ to $O(m)$.

- Such spectacular speedups do occur in practice, but more typical might be from $O(m^3)$ to $O(m^2)$.

Exact vs Approximate Solutions

- Iterative methods are approximate, delivering approximate answers even in the presence of no rounding errors.
- Iterative methods are “engineering solution” of “little elegance and doubtful reliability”.
- However, direct methods are also inexact in floating-point arithmetic.
- Accuracy can only be achieved to $O(\varepsilon_{machine})$ anyway, whether it be because of round-off errors in an “exact” algorithm or due to other approximations made.

Exact vs Approximate Solutions

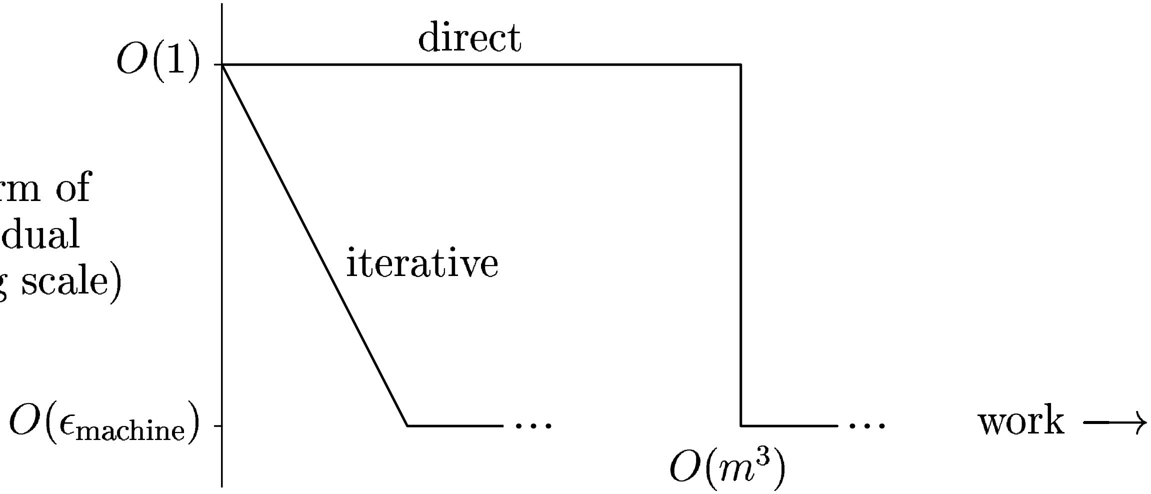

Figure 32.1. *Schematic illustration of convergence of direct and iterative methods. Under favorable circumstances, the iterative method converges geometrically until the residual is on the order of* $\epsilon_{\text{machine}}$. *The direct method makes no progress at all until* $O(m^3)$ *operations are completed, at which point the residual is again on the order of* $\epsilon_{\text{machine}}$.

Direct Methods that Beat $O(m^3)$

- There are direct algorithms that solve $Ax = b$ and related problems for dense matrices in less than $O(m^3)$ operations.

- V. Strassen in 1969 discovered an algorithm that reduces the exponent 3 to $\log_2(7) \approx 2.807$.
- In 1990, Coppersmith and Winograd reduced the exponent to $\approx$ 2.3737.
- In 2010, Stothers reduced the exponent to ≈ 2.3736.
- In 2011, Williams reduced the exponent to ≈ 2.3727.

Direct Methods that Beat $O(m^3)$

- These improvements have not really impacted practical computation because:
  - We need more information regarding their stability.
  - The values of m that beat the standard are large.

- The Strassen algorithm is faster than Gaussian elimination for *m > 100*, but because the exponent is still close to 3, there is no big improvement.

Direct Methods that Beat O(m³)

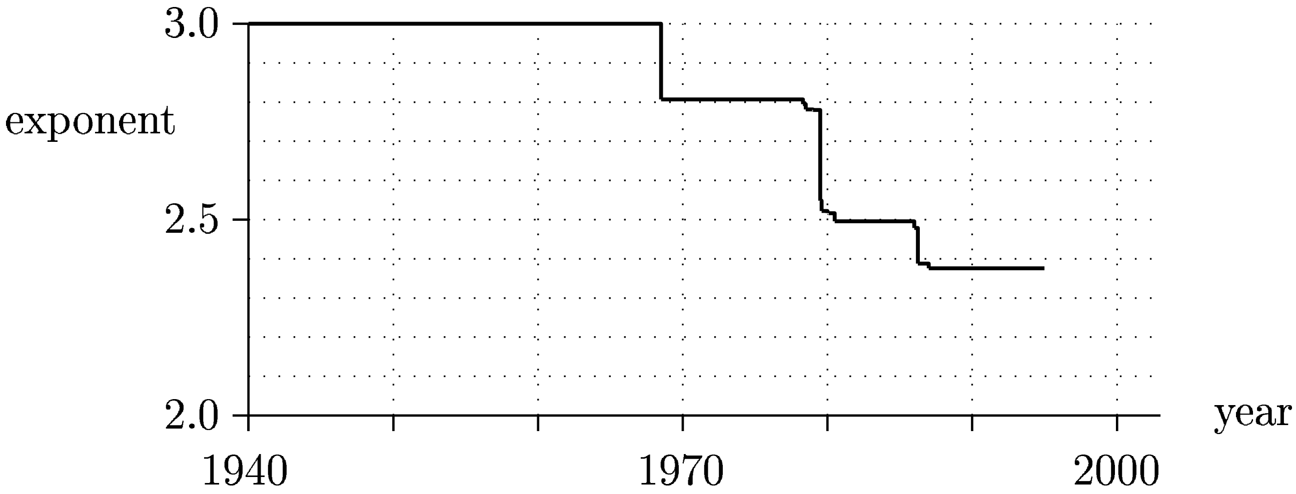

Figure 32.2. *Best known exponents for direct solution of $Ax = b$ (or equivalently, for computation of A^{-1}, AB, or $\det A$) for $m \times m$ matrices, as a function of time. Until 1968, the best known algorithms were of complexity $O(m^3)$. The currently best known algorithm solves $Ax = b$ in $O(m^{2.376})$ flops, but the constants are so large that this algorithm is impractical.*