Numerical Linear Algebra

Maya Neytcheva, TDB, February-March 2021

A little bit of repetition...

The major task of this course is to learn

how to solve $A\mathbf{x} = \mathbf{b}$ and $A\mathbf{x} = \lambda\mathbf{x}$ when A is of large size.

Why? Where do these systems arise?

Linear systems arise in numerical simulations.
Numerical simulations should be seen as a cross-disciplinary task because conducting a numerical simulation incorporates:

- Modelling
- Discretization
- Choice of a solution method
- Computer implementation
- Postprocessing

- Modelling: modelling error
- Discretization: discretization error (space, time,stability...)
- Choice of a solution method: iteration error (robustness wrt discretization parameters,)
- Implementation, computer platform, suitability (memory or computation-bound), data structures, communication layout

Course contents:

- ➜ Introduction, NLA, basic iterative schemes
- ➜ Projection methods
- ➜ Speeding up the convergence - preconditioning
- ➜ Multilevel/multigrid preconditioners
- ➜ Structured matrices, properties, preconditioning
- ➜ Num. Solution methods for eigenvalue problems
- ➜ Possibly: Matrix factorizations, LS, SVD

Learning Goals

At the end of the course, the participant should be able to

- be aware of, understand, and be able to make arguments about central issues regarding numerical solution methods for linear systems, concerning numerical efficiency, computational efficiency (complexity of the numerical algorithm), robustness with respect to problem, discretization and method parameters, possible parallelization;
- given a problem, have a clear guiding criteria how to choose suitable solution technique and can reason what could be advantageous and disadvantageous;
- demonstrate how the algorithms can be applied (on some test problems).

Course moments:

- Lectures - **very** highly recommended!
- Hands-on sessions - must do; come up with some comments and **at least two questions**, to be discussed in the beginning of the next lecture.
  Currently - 5 such planned.
- Assignments - **three** such!
  – Written report as a scientific paper in English.
  – Structure, language, derivations, algorithms, figures, conclusions, references

Assumed that you are familiar with/can:

- vector subspaces
- full (column/row) rank
- positive definite matrices
- matrix/vector norms and condition numbers
- (strictly) diagonally dominant matrix
- spectral radius
- Z-matrices, M-matrices
- Schur's lemma
- dense/sparse matrices
- direct solvers/sparse direct solvers/pivoting/complexity
- fill-in, ordering (Minimal degree, Reverse Cuthill-McKee, nested dissection)
- derive the eigenvalues of the 1D and 2D discrete Laplacian

Partial list of application fields

Stiff ODEs
BDF
Linear programming (simplex, interior point methods)
Optimization
Nonlinear equations
Elliptic PDEs
Eigensolutions
Two-point boundary value problems
Least Squares calculations

More application fields

acoustic scattering	demography	network flow
air traffic control	economics	oceanography
astrophysics	electrical eng.	petroleum eng.
biochemical	electric nets	reactor modelling
chemical eng.	climate/pollution studies	statistics
chemical kinetics	fluid flow	structural eng
circuit physics	laser optics	survey data
computer simulations	linear programming	signal processing

Some notions from matrix theory

- Revision: Vectors and matrices. Range, null space, rank of a matrix.
  Vector and matrix norms. Norm equivalence
- Matrix eigenvalues, minimal polynomial, similarity and congruent transformations, Schur's lemma. Gershgorin's theorem, Courant-Fischer lemma
- Condition number, spectral condition number,
  Spectral radius $\rho(A) = \max_{\lambda \in S(A)} (|\lambda|)$, $\|A\|_2 = \sqrt{\rho(A)}$
- Error analysis: stability of numerical algorithms

$$\|\widehat{x} - x\| = \|A^{-1}b - x\| = \|A^{-1}(b - Ax)\| = \|A^{-1}r\| \leq \|A^{-1}\|\|r\|$$

Matrix theory, cont.

- Some special classes of matrices and their properties. Unitary (orthogonal), selfadjoint, Positive definite matrices, (strict) diagonal dominance
- Schur complements
- Matrices with a special structure
- Matrix factorizations. Gaussian elimination, LU-decomposition
- Reorganizing the Gauss elimination process. Direct solution methods for sparse matrices. Ordering strategies. Sparse matrices (some issues touched, such as (re)ordering)
- Factorization of spd systems (Cholesky factorization)
- Computational cost

Some implementation-related issues

ijk

for $i = 1 : n$
    for $j = 1 : n$
        for $k = 1 : n$
            $c(i,j) = c(i,j) + a(i,k) * b(k,j)$
        end
    end
end

---

for $i = 1 : n$
    for $j = 1 : n$
        $c(i,j) = c(i,j) + a(i,:) * b(:,j)$ scalar product form
        end
    end
end

jki

for $j = 1 : n$
    for $k = 1 : n$
        for $i = 1 : n$
            $c(i,j) = c(i,j) + a(i,k) * b(k,j)$
    end
end

---

for $j = 1 : n$
    for $k = 1 : n$
        $c(:,j) = c(:,j) + a(:,k) * b(k,j)$ vector update form
    end
end

jik

for $k = 1 : n$
    for $j = 1 : n$
        for $i = 1 : n$
            $c(i,j) = c(i,j) + a(i,k) * b(k,j)$
        end
    end
end

---

for $k = 1 : n$
    $c(:,:) = c(:,:) + a(:,k) * b(k,:)$ outer product form
end

Matrix factorizations

- Schur's form: $A = U^T RU$
- LU
- LL^T or $U^T U$
- LDU

The LDU factorization

Theorem:

If all leading principal minors of A are nonsingular, then there exist unique lower-triangular matrix L, diagonal matrix D and upper-triangular matrix U, such that $A = LDU$.

How do we compute $A = LDU$?

Note: If e_j is the jth unit vector, then $Me_j = M(:,j)$.

Assume that the first $j-1$ columns of L, $j-1$ elements of D and $j-1$ rows of U are computed.

$$\begin{bmatrix} 1 & & & & & \\ * & \ddots & & & & \\ * & * & 1 & & & \\ ? & ? & ? & 1 & & \\ ? & ? & ? & ? & \ddots & \\ ? & ? & ? & ? & ? & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & & & \\ & \ddots & & & & \\ & & d_{j-1} & & & \\ & & & ? & & \\ & & & & \ddots & \\ & & & & & ? \end{bmatrix} \begin{bmatrix} 1 & * & * & ? & ? & ? \\ & \ddots & * & ? & ? & ? \\ & & 1 & ? & ? & ? \\ & & & 1 & ? & ? \\ & & & & \ddots & ? \\ & & & & & 1 \end{bmatrix}$$

How do we compute $A = LDU$?

We equate the jth column of $A = LDU$:
Denote $v = DUe_j$. Then $Ae_j = Lv$

- $v(1:j) = L(1:j, 1:j)^{-1}A(1:j, j)$ - known data
- $d(j) = v(j)$
- $U(i,j) = v(i)/d(i),\ i = 1:j-1$
- $L(j+1:n, j)v(1:j) = A(j+1, j)$

Example of implementing Cholesky factorization

```
for k=1:n
    xeuitb(A(1:k-1,k),A(1:k-1,1:k-1),A(1:k-1,k))
    A(k,k) = sqrt(A(k,k) - A(1:k-1,k)^T*A(1:k-1,k))
end
```

Computes U (which overwrites A).

BLAS `xeuitb(X,U,B)` computes $X = U^{-1}B$