6 Summary

Here we summarize the most important information about theoretical and numerical linear algebra.

MORALS OF THE STORY:

I. Theoretically perfect algorithms could be very unstable numerically. Errors pop up unaviodably, and they can get amplified enormously, especially for real-life applications, when big matrices are considered.

II. Orthogonal matrices are the best for numerical stability (their condition number is 1). They should be used all the times.

Linear algebra has two fundamental problems:

(i) Solving $A\mathbf{x} = \mathbf{b}$

(ii) Diagonalizing a matrix A.

6.1 Methods for $A\mathbf{x} = \mathbf{b}$

6.1.1 Theoretical methods for $A\mathbf{x} = \mathbf{b}$.

The most common method is the **Gauss elimination**, which is equivalent to the **LU decomposition**.

Advantages of LU:

(i) Applicable for any matrix

(ii) Finds all solutions

(iii) Easy to program

(iv) Fast

Disadvantages of LU:

(i) Could easily be unstable

(ii) Does not find approximate solutions (least square)

A more advanced method is the **QR-decomposition**. Suppose first that we already have an exact QR-decomposition. Then $A\mathbf{x} = \mathbf{b}$ is reduced to $R\mathbf{x} = Q^t\mathbf{b}$ which is easy to backsolve.

Advantages of QR over LU:

(i) If you have a QR-decomposition, then solving $A\mathbf{x} = \mathbf{b}$ via $R\mathbf{x} = Q^t\mathbf{b}$ is as well-conditioned as the original problem.

(ii) Finds least square solutions as well (when no exact solution exists). When there are exact solutions, it finds all of them.

Disadvantages of QR over LU:

(i) It usually takes a more complicated algorithm to find the QR-decomposition than LU.

(ii) It is slower than LU (mainly because it has to compute lots of norms, square roots etc.)

But in general **QR-decomposition is always superior to the LU-decomposition**.

This raises the question: How to **find the QR-decomposition**? Theoretically, the QR-decomposition is obtained via **Gram-Schmidt procedure**

Advantages of finding QR with Gram-Schmidt:

(i) Applicable for any matrix

(ii) Easy to program

(iii) Fast

Disadvantages of finding QR with Gram-Schmidt:

(i) Can be unstable, especially for matrices with (almost) linearly dependent columns.

6.1.2 Numerical methods for $A\mathbf{x} = \mathbf{b}$.

LU-decomposition (=Gauss elimination) can be improved by **partial pivoting**

Advantages of partial pivoting

(i) Can improve the stability of LU quite a bit

(ii) Easy to program, it almost takes no extra effort and running time

Disadvantages of partial pivoting

(i) Only reduces instability, but in many cases still far from the optimal stability.

Given the advantages of solving $A\mathbf{x} = \mathbf{b}$ with QR, it is of primary importance to find the QR-decomposition in a numerically stable way. The way to do it is via *orthogonal matrices*, i.e. by **Householder reflections or Givens rotations**. The performance of the two methods are similar. Givens is easier to program, but performs slower.

Advantages of Householder or Givens over Gram-Schmidt

(i) They are stable

(ii) They also can be applied for any matrix, as Gram-Schmidt (though we just learned them for square matrices).

Disadvantages of Householder or Givens over Gram-Schmidt: **NONE**

Finally, certain equations $A\mathbf{x} = \mathbf{b}$ can be solved iteratively. We learned the **Jacobi and Gauss-Seidel iterations** (only for square matrices, but there are versions for general matrices). In general **Gauss-Seidel is better than Jacobi** for speed, memory requirement and programming.

Advantages of Jacobi or Gauss Seidel over the exact methods: LU and QR

(i) Stable (QR is also stable, LU is not)

(ii) They are fast for special matrices.

Disadvantages of Jacobi or Gauss Seidel over the exact methods: LU and QR

(i) For many matrices they don't converge. There are many tricks to improve this feature, but in general these methods are applicable only for special matrices – something like "strong diagonal". But for numerical solutions of differential equations such matrices arise very often.

(ii) Even if they converge, they find only one solution. Usually they perform badly for degenerate matrices (we just learned them for regular square matrices, but again, there are generalizations)

6.2 Methods for diagonalizing a square matrix A

6.2.1 Theoretical methods for diagonalizing A, finding eigenvalues, -vectors

Remember, that in general there is no exact method, only iterative ones.

The basic strategy is first finding the characteristic polynomial $p(\lambda)$ (by computing a determinant), then finding the roots of $p(\lambda)$ (usually by Newton's method – this is the iterative part of the game), finally finding the eigenvectors by solving several linear equations (one for each eigenvalue). This last problem is the one discussed in the previous section.

Advantages of diagonalizing via charateristic polynomial:

(i) Applicable for any diagonalizable matrix

(ii) Conceptually clear

(iii) Newton's method is stable and there are methods in the previous section to solve $(A - \lambda)\mathbf{x} = 0$ in a stable way.

Disadvantages of diagonalizing via charateristic polynomial:

(i) Computing large determinants (especially with a variable symbol λ) can be unstable. We did not discuss this issue

(ii) Newton's method works only if there are some a-priori guesses on the solutions. In addition, it requires modifications to deal with complex eigenvalues.

(iii) The program is quite complicated altogether, it has many steps and takes a long time (you have to solve n equations at the end).

6.2.2 Numerical methods for diagonalizing A, finding eigenvalues, -vectors

The **Power method**:

Advantages of the power method:

(i) Fast and simple for individual eigenvalues-vectors.

(ii) Can be modified (shift, inversion) to find all eigenvalues (including complex ones), but need some guess on the location of the eigenvalues (as in Newton)

(iii) Stable.

Disadvantages of the power method:

(i) There are complications for multiple eigenvalues.

(ii) Needs some initial guess on the eigenvalues

(iii) One has to find each eigenvalue-eigenvector separately, i.e. not very fast for full diagonalization.

The **Jacobi iteration**:

Advantages of the Jacobi iteration

(i) Fully diagonalizes a symmetric matrix in one iteration

(ii) Easy to program

(iii) Stable.

Disadvantages of the Jacobi iteration

(i) Works only for symmetric matrices (but this is what one needs for SVD)

(ii) Slow

The **QR-iteration**: This is the **best and most commonly used method** for eigenvalues (with its several variants and improvements)

Advantages of the QR-iteration:

(i) Gives eigenvalues for general matrices and also eigenvectors for symmetric ones.

(ii) Gives everything in one iteration

(iii) Faster than Jacobi, if one has a good QR-decomposition algorithm

(iv) Stable

Disadvantages of the QR-iteration:

(i) Cannot deal with complex or multiple eigenvalues, unless modified. Modifications exist but they are not so simple.