SF2524 - Matrix computations for large-scale systems

$\approx$ Numerical linear algebra for large-scale systems

Intro lecture, October 30, 2018

Elias Jarlebring
KTH Royal Institute of Technology
Mathematics Dept. - NA division

About the teachers
About the students
About the topic
About the course

Lecture 1

- About the teachers
- About the students
- About the topic
- About the course

- Fundamental eigenvalue techniques:
  - Rayleigh quotient
  - Power method
  - Inverse iteration
  - Rayleigh qoutient iteration

About the teachers
About the students
About the topic
About the course

About the Lecturer

About the teachers
About the students
About the topic
About the course

Elias Jarlebring - Associate Professor - teacher - researcher

Background - Elias Jarlebring

- From: Vännäs/Umeå, Sweden
- MSc: KTH, Stockholm (Teknisk fysik)
- MSc thesis: TU Hamburg
- PhD: TU Braunschweig, Germany
- Post-doc: KU Leuven, Belgium
- Dahlquist fellow: KTH, Stockholm
- Assoc. Prof (Lektor): KTH, Stockholm
- Assoc. Prof (Docent): KTH, Stockholm

About the teachers
About the students
About the topic
About the course

CV - continued

- Researcher:
  - applied and computational mathematics
  - numerical linear algebra: e.g. Nonlinear eigenvalue problems
- Teacher: numerical methods and numerical linear algebra
- Hacker/programmer: Open source projects
- Language nerd: Swedish, English, German, Dutch, Russian
- Language nerd: C/C++, Assembler, Julia, Java, ...
- EU globetrotter: Sweden, Ireland, Germany, Belgium, USA

Teaching portfolio - Elias Jarlebring

- Experience: All university levels + four countries
  bachelor, master, PhD-level (+high-shool level)
- Teaching style: lectures with blended learning
  slides, blackboard, live computer demos, additional online material, quizzes, wiki activity

Student comments about E.J. as a teacher

- Germany 2004: “We don’t understand what he is saying. We can’t read what he is writing, but he is nice and draws beautiful figures.”
- Germany 2006: Clear explanations
- Sweden ∼2012: Authorative style. Strict. Structured and competent.
- Sweden ∼2016: The best learning experience I have had

About the teachers
About the students
About the topic
About the course

About the Teaching Assistant

About the teachers
About the students
About the topic
About the course

Teaching assistant: Giampaolo Mele

TA: Giampaolo Mele

- Moderator of Wiki
- Answers questions (email)
- Answers questions office hours
- Substitute lecturer two lectures
- Competent: researcher in numerical linear algebra
- Very friendly!

About the students

About the teachers
About the students
About the topic
About the course

MSc / BSc students

- Master's programme in applied and computational mathematics
- Master's programme in Computer simulation for science and engineering (COSSE, TDTNM)
- Master's programme in Machine learning
- Nordic N5TeAM Master's Programme, Applied and Engineering Mathematics (TITMM)

PhD students

Applied and computational mathematics, Mechanics, Electrical engineering

Students from countries

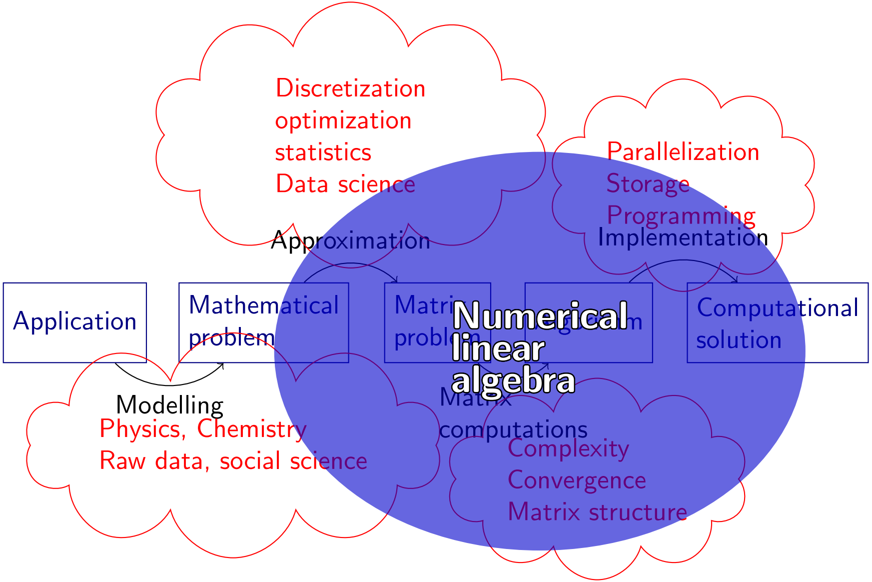
Sweden, France, Germany, USA, Denmark, Netherlands, India, South africa, China, UK, Spain, . . .

Beware: Different student background ⇒ Different skill set.

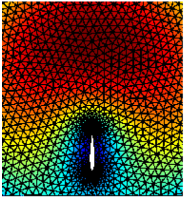
About the topic

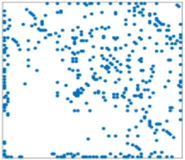
About the teachers
About the students
About the topic
About the course

Numerical linear algebra in a bigger context



Numerical linear algebra in a bigger context





Definition: Numerical linear algebra

Numerical linear algebra is the study of numerical methods for linear algebra operations, a.k.a. fun part of linear algebra.

Large-scale matrix computations

- Algorithms and methods that involve matrices of large size
- Large-scale matrix computations $\subset$ Numerical linear algebra

Applications / motivation

Applications arise in essentially all scientific fields

- Physics, mechanics, astronomy, etc
- Chemistry, quantum chemistry, biology,
- Data science, data analysis, machine learning
- Discretizations of PDEs
- ...

The predictive power of the model is often limited by the performance of the algorithms. We study the details of the algorithms.

About the course - SF2524

About the teachers
About the students
About the topic
About the course

Course contents - SF2524

A selection of topics in numerical linear algebra.
Separated into blocks:

- Background: Orthogonal matrices & Jordan decomposition
- Block 1: Large and sparse eigenvalue algorithms
- Block 2: Iterative methods for Linear systems
- Block 3: QR method
- Block 4: Matrix functions
- (Block 5: Matrix equations only PhD students SF3580)

Why these topics?

- Most mature problem classes in research on matrix comp
- Most common matrix problems in applications

Lectures

Lectures: approx 15 lectures

- Introduce you to concepts (pre-cooking)
- Sometimes more details where book not satisfactory
- Learning by watching live programming (+interaction)

Lecture overview (preliminary)

- Lecture 1-4: Block 1: Eigenvalue algorithms (part 1)
- Lecture 4-9: Block 2: Linear systems of equations
- Lecture 10-11: Block 3: Eigenvalue algorithms (part 2): QR-method
- Lecture 12-15: Block 4: Functions of matrices

About the teachers
About the students
About the topic
About the course

Practicalities

Course webpage

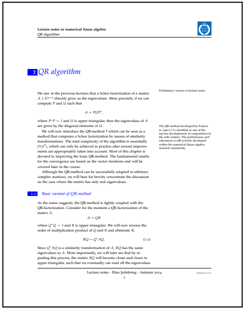
- Online learning platform: CANVAS
- Course registration necessary to obtain complete access.
- Most course material online
- Quiz (partially) mandatory

About the teachers
About the students
About the topic
About the course

About the teachers
About the students
About the topic
About the course

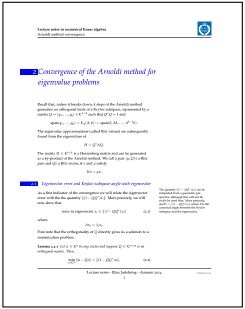
Literature

- Lecture notes PDFs online (block$X$.pdf $X = 1, 2, 3, 4, 5$). References to pages in [TB].
- *Numerical Linear Algebra* by Lloyd N. Trefethen and David Bau [TB], available in kårbokhandeln


Lecture notes in numerical linear algebra
QR algorithm

1 *QR algorithm*

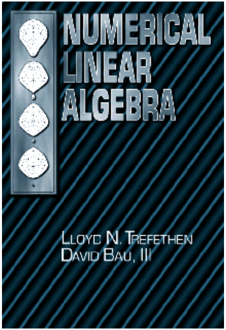
Lecture notes - Elias Jarlebring - Autumn 2014


Lecture notes in numerical linear algebra
Arnoldi method convergence

2 *Convergence of the Arnoldi method for eigenvalue problems*

2.1 *Eigenvector error and Krylov subspace angle with eigenvector*

Lecture notes - Elias Jarlebring - Autumn 2014



Programming language

SF2524: Select between

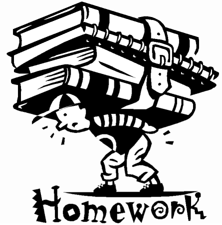
- MATLAB; or
- Julia language

SF3580:

- Julia language

- Interested in Julia: Introduction lecture 31 Oct 10:15.
- Live programming in lectures will be in MATLAB.

About the teachers
About the students
About the topic
About the course



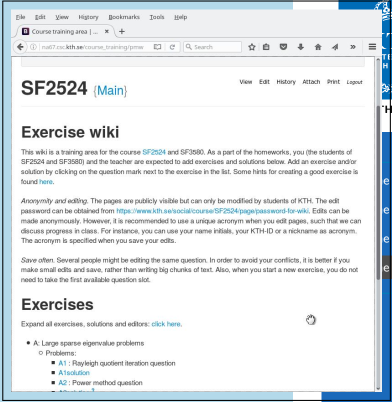
Homework

- $3\times$ homework sets: theory and hands-on practice of the methods
- Work in groups of at most two
- Compulsary, can give bonus points for exam
- Hand in correct solutions before deadline $\Rightarrow$ bonus points for exam. One report per group.
- Hand in via CANVAS by
  Uploading PDF-file with solutions and MATLAB-code

About the teachers
About the students
About the topic
About the course

Course wiki: active learning

- Students create problems and solutions
- Optional part of homework
- Moderation by Giampaolo and Elias
- Public but anonymous to outsiders
- Can lead to **wiki bonus**
- **Wiki bonus** reduces exam limits for grade A and B
- Highly collaborative training activity
- Think out of the box! Help each other! It's fun!



Course analysis and development

Greetings from “older” students:

Messages from students of previous year(s)

- “Take notes during lectures. The proofs in the book are sometimes incomplete.”
- “I first looked at the home-work and thought, this will be so much work..., and then we actually started and the tasks in the homework were specific so it went fast”
- “The homework are designed to check understanding of the actual contents of the course.”
- “High attendence in the lectures is important“
- “After the second lecture, I thought, wow this is totally different“

Course development HT19 (see course_analysis_ht18.pdf)

- New parts in homeworks
- More written material in blockX.pdf
- Quizzes integrated into homeworks

Time to start the lecture ...

About the teachers
About the students
About the topic
About the course

Time to start the lecture ...

Fundamental eigenvalue techniques (block 1)

- Rayleigh quotient
- Power method = power iteration
- Inverse iteration
- Rayleigh qoutient iteration

SF2524
SF3580

Elias Jarlebring

KTH - SCI

Algorithm 27.1. Power Iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$
for $k = 1, 2, \ldots$
  $w = Av^{(k-1)}$ — apply A
  $v^{(k)} = w/\|w\|$ — normalize
  $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

In this and the algorithms to follow, we give no attention to termination conditions, describing the loop only by the suggestive expression "**for** $k = 1, 2, \ldots$." Of course, in practice, termination conditions are very important, and this is one of the points where top-quality software such as can be found in LAPACK or MATLAB is likely to be superior to a program an individual might write.

We can analyze power iteration easily. Write $v^{(0)}$ as a linear combination of the orthonormal eigenvectors q_i:

$$v^{(0)} = a_1 q_1 + a_2 q_2 + \cdots + a_m q_m.$$

Since $v^{(k)}$ is a multiple of $A^k v^{(0)}$, we have for some constants c_k

$$\begin{aligned} v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \cdots + a_m \lambda_m^k q_m) \\ &= c_k \lambda_1^k \left(a_1 q_1 + a_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + a_m (\lambda_m/\lambda_1)^k q_m\right). \end{aligned} \quad (27.4)$$

From here we obtain the following conclusion.

Theorem 27.1. *Suppose* $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ *and* $q_1^T v^{(0)} \neq 0$. *Then the iterates of Algorithm 27.1 satisfy*

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \quad (27.5)$$

as $k \to \infty$. *The* $\pm$ *sign means that at each step* k, *one or the other choice of sign is to be taken, and then the indicated bound holds.*

Proof. The first equation follows from (27.4), since $a_1 = q_1^T v^{(0)} \neq 0$ by assumption. The second follows from this and (27.3). If $\lambda_1 > 0$, then the $\pm$ signs are all $+$ or all $-$, whereas if $\lambda_1 < 0$, they alternate. □

The $\pm$ signs in (27.5) and in similar equations below are not very appealing. There is an elegant way to avoid these complications, which is to speak of convergence of subpaces, not vectors—to say that $\langle v^{(k)} \rangle$ converges to $\langle q_1 \rangle$, for

example. However, we shall not do this, in order to avoid getting into the details of how convergence of subspaces can be made precise.

On its own, power iteration is of limited use, for several reasons. First, it can find only the eigenvector corresponding to the largest eigenvalue. Second, the convergence is linear, reducing the error only by a constant factor $\approx |\lambda_2/\lambda_1|$ at each iteration. Finally, the quality of this factor depends on having a largest eigenvalue that is significantly larger than the others. If the largest two eigenvalues are close in magnitude, the convergence will be very slow.

Fortunately, there is a way to amplify the differences between eigenvalues.

Inverse Iteration

For any $\mu \in \mathbb{R}$ that is not an eigenvalue of A, the eigenvectors of $(A - \mu I)^{-1}$ are the same as the eigenvectors of A, and the corresponding eigenvalues are $\{(\lambda_j - \mu)^{-1}\}$, where $\{\lambda_j\}$ are the eigenvalues of A. This suggests an idea. Suppose μ is close to an eigenvalue λ_J of A. Then $(\lambda_J - \mu)^{-1}$ may be much larger than $(\lambda_j - \mu)^{-1}$ for all $j \neq J$. Thus, if we apply power iteration to $(A - \mu I)^{-1}$, the process will converge rapidly to q_J. This idea is called *inverse iteration*.

Algorithm 27.2. Inverse Iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$
for $k = 1, 2, \ldots$
  Solve $(A - \mu I)w = v^{(k-1)}$ for w — apply $(A - \mu I)^{-1}$
  $v^{(k)} = w/\|w\|$ — normalize
  $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

What if μ is an eigenvalue of A, so that $A - \mu I$ is singular? What if it is nearly an eigenvalue, so that $A - \mu I$ is so ill-conditioned that an accurate solution of $(A - \mu I)w = v^{(k-1)}$ cannot be expected? These apparent pitfalls of inverse iteration cause no trouble at all; see Exercise 27.5.

Like power iteration, inverse iteration exhibits only linear convergence. Unlike power iteration, however, we can choose the eigenvector that will be found by supplying an estimate μ of the corresponding eigenvalue. Furthermore, the rate of linear convergence can be controlled, for it depends on the quality of μ. If μ is much closer to one eigenvalue of A than to the others, then the largest eigenvalue of $(A - \mu I)^{-1}$ will be much larger than the rest. Using the same reasoning as with power iteration, we obtain the following theorem.

Theorem 27.2. *Suppose* λ_J *is the closest eigenvalue to* μ *and* λ_K *is the second closest, that is,* $|\mu - \lambda_J| < |\mu - \lambda_K| \leq |\mu - \lambda_j|$ *for each* $j \neq J$. *Furthermore,*

suppose $q_J^T v^{(0)} \neq 0$. *Then the iterates of Algorithm 27.2 satisfy*

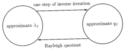
$$\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)$$

as $k \to \infty$, *where the* $\pm$ *sign has the same meaning as in Theorem 27.1.*

Inverse iteration is one of the most valuable tools of numerical linear algebra, for it is the standard method of calculating one or more eigenvectors of a matrix if the eigenvalues are already known. In this case Algorithm 27.2 is applied as written, except that the calculation of the Rayleigh quotient is dispensed with.

Rayleigh Quotient Iteration

So far in this lecture, we have presented one method for obtaining an eigenvalue estimate from an eigenvector estimate (the Rayleigh quotient), and another method for obtaining an eigenvector estimate from an eigenvalue estimate (inverse iteration). The possibility of combining these ideas is irresistible:



(The figure is oversimplified; to get from an approximate λ_J to an approximate q_J by a step of inverse iteration, one also needs a preliminary approximation to q_J.) The idea is to use continually improving eigenvalue estimates to increase the rate of convergence of inverse iteration at every step. This algorithm is called *Rayleigh quotient iteration*.

Algorithm 27.3. Rayleigh Quotient Iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$
$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ = corresponding Rayleigh quotient
for $k = 1, 2, \ldots$
  Solve $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$ for w — apply $(A - \lambda^{(k-1)} I)^{-1}$
  $v^{(k)} = w/\|w\|$ — normalize
  $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

About the teachers
About the students
About the topic
About the course