

SF2524 - Matrix computations for large-scale systems

≈ Numerical linear algebra for large-scale systems

Intro lecture, October 31, 2017



KTH - SCI

[About the lecturer](#)

[About the students](#)

[About the topic](#)

[About the course](#)

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

Lecture 1

- About the lecturer
- About the students
- About the topic
- About the course

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



KTH - SCI

About the lecturer

About the students

About the topic

About the course



KTH - SCI

About the Lecturer

About the lecturer

About the students

About the topic

About the course



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 lecturer

About the students

About the topic

About the course

CV - continued

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



KTH - SCI

[About the lecturer](#)[About the students](#)[About the topic](#)[About the course](#)

Teaching portfolio - Elias Jarlebring

- Experience: All university levels
bachelor, master, PhD-level (+high-school 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



KTH - SCI

[About the lecturer](#)[About the students](#)[About the topic](#)[About the course](#)



KTH - SCI

About the students

About the lecturer

About the students

About the topic

About the course

Students from programmes

- 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)



KTH - SCI

[About the lecturer](#)[About the students](#)[About the topic](#)[About the course](#)

Students from countries

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

Beware: Different student background \Rightarrow Different skill set.



KTH - SCI

About the topic

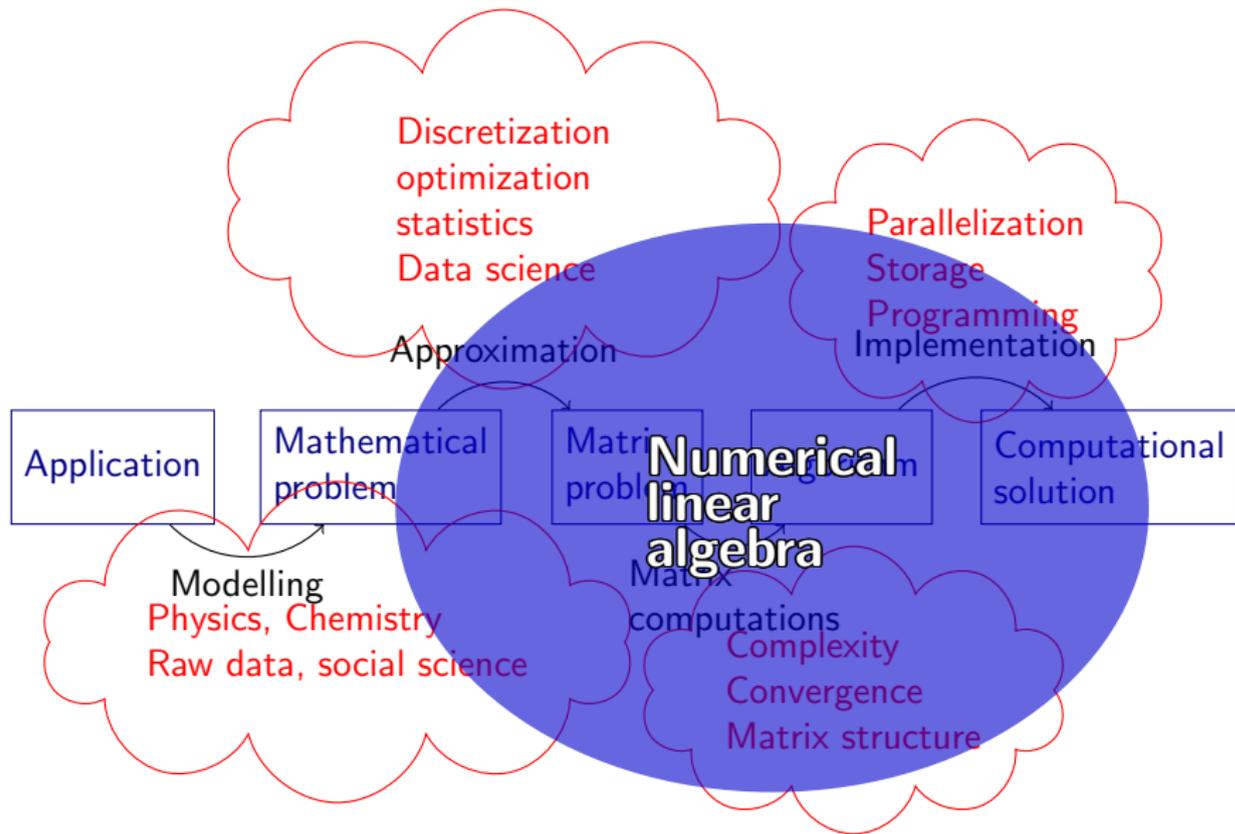
About the lecturer

About the students

About the topic

About the course

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



KTH - SCI

Applications / motivation

Applications arise in essentially all scientific fields

- Physics, mechanics, astronomy, etc
- Chemistry, quantum chemistry, biology,
- Data science and data analysis
- 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 lecturer](#)[About the students](#)[About the topic](#)[About the course](#)



KTH - SCI

About the course - SF2524

About the lecturer

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)



KTH - SCI

[About the lecturer](#)

[About the students](#)

[About the topic](#)

[About the course](#)

Why these topics?

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

Lectures: approx 13 lectures

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



KTH - SCI

Lecture overview (preliminary)

- Lecture 1-4: Block 1: Eigenvalue algorithms (part 1)
 - Power method, Rayleigh quotient iteration
 - Krylov methods
- Lecture 4-9: Block 2: Linear systems of equations
 - Krylov methods: GMRES, CG, BiCGstab
- Lecture 10-11: Block 3: Eigenvalue algorithms (part 2): QR-method
- Lecture 12-15: Block 4: Functions of matrices
 - Scaling-and-squaring, Krylov methods, ...

[About the lecturer](#)[About the students](#)[About the topic](#)[About the course](#)



KTH - SCI

Course webpage

- Online learning platform: CANVAS
- Course registration necessary to obtain complete access.
- Most course material online
- New: Quiz (for your training)

[About the lecturer](#)

[About the students](#)

[About the topic](#)

[About the course](#)

Literature

- Lecture notes PDFs online. References to pages in [TB].
- Numerical Linear Algebra* by Lloyd N. Trefethen and David Bau [TB], available in k arbokhandeln



KTH - SCI

About the lecturer
About the students
About the topic
About the course

Lecture notes in numerical linear algebra
QR algorithm

QR algorithm

The aim of the previous lecture was to obtain a factorization of a matrix $A \in \mathbb{C}^{n \times n}$ directly given as the eigenvalues. More precisely, if we can compute $\sqrt{\lambda}$ for each λ .

Practically, however, this is not possible.

Let $P^* P = I$ and U is upper triangular. Then the eigenvalues of A are given by the diagonal elements of U .

We will now introduce the QR method \mathcal{Q} which can be seen as a method that computes a factorization of a matrix in terms of unitary transformations. The total complexity of the algorithm is essentially $\mathcal{O}(n^3)$, which can only be reduced by parallelizing other numerical algorithms.

For the convergence we will assume that all eigenvalues are algebraically simple and that the matrix is not defective. The transformation matrix for the convergence is based on the vector iteration and will be used later in the lecture.

The convergence of the QR method can be considered as a stability problem. In this case, we will have to formally consider the discussion on the case where the matrix is not well conditioned.

Basic version of QR method

In the basic version, the QR method is tightly coupled with the QR factorization. Consider for the moment a QR factorization of the matrix A .

Let $Q \in \mathbb{C}^{n \times n}$ and R is upper triangular. We will now consider the action of multiplication product $Q^* A$ and R and obtain A_1 .

$A_1 = Q^* A R$

Since $Q^* A R$ is a similarity transformation of A , A_1 has the same eigenvalues as A . More importantly, we will later see that by repeating the process, the matrix A_k will become closer and closer to upper triangular until the non-zero elements are all on the diagonal.

Reference: [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100].

Lecture notes in numerical linear algebra
Convergence of the Arnoldi method for eigenvalue problems

Convergence of the Arnoldi method for eigenvalue problems

Recall that the aim of this lecture is to obtain the Arnoldi method for computing a few eigenvalues and eigenvectors of a matrix $A \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{n \times k}$ and $R \in \mathbb{C}^{k \times k}$ such that $AQ = QR$.

The algorithm approximates the eigenvalues and eigenvectors based on the approximation $A \approx QR$.

The matrix $R \in \mathbb{C}^{k \times k}$ is a Hessenberg matrix and can be generated as a by-product of the Arnoldi method. We will later see that Q is a thin QR factorization of A and R is a $k \times k$ matrix.

Approximation error and angle between angle with eigenvalue

In this lecture, we will consider the convergence of the Arnoldi method. We will see that the quantity $\|A - QR\|$ is small, more precisely, we will see that $\|A - QR\| \leq \mathcal{O}(\epsilon)$.

Let $A \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{n \times k}$ and $R \in \mathbb{C}^{k \times k}$ such that $AQ = QR$.

Then we can see that the orthogonality of Q directly gives us a solution to a minimization problem.

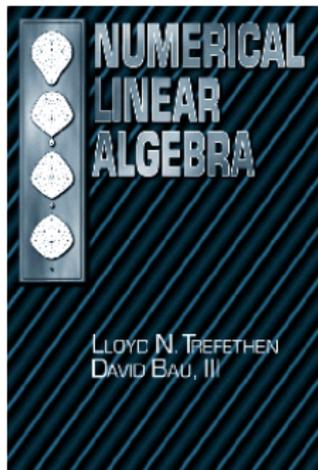
Let $A \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{n \times k}$ and $R \in \mathbb{C}^{k \times k}$ such that $AQ = QR$.

Then we can see that the orthogonality of Q directly gives us a solution to a minimization problem.

Let $A \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{n \times k}$ and $R \in \mathbb{C}^{k \times k}$ such that $AQ = QR$.

Then we can see that the orthogonality of Q directly gives us a solution to a minimization problem.

Reference: [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100].





MATLAB®



KTH - SCI

Programming language

You are allowed to solve the homework/exam problems with either

- MATLAB; or
- Julia language (warning experimental)

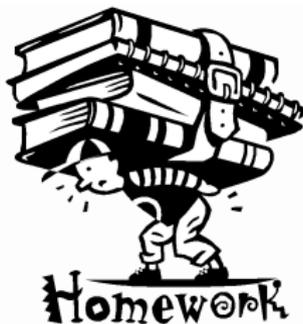
Live programming in lectures will be in MATLAB.

About the lecturer

About the students

About the topic

About the course



Homework

- 3× 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 lecturer

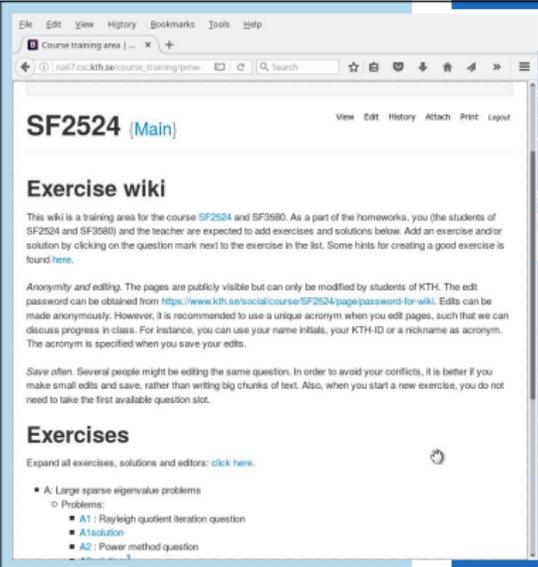
About the students

About the topic

About the course

Course wiki: active learning

- Students create problems and solutions
- Optional part of homework
- Moderation by Parikshit 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!



File Edit View History Bookmarks Tools Help

Course training area | ...

no67.cockth.se/course_training/pmaw

SF2524 (Main) View Edit History Attach Print Layout

Exercise wiki

This wiki is a training area for the course SF2524 and SF3580. As a part of the homeworks, you (the students of SF2524 and SF3580) and the teacher are expected to add exercises and solutions below. Add an exercise and/or solution by clicking on the question mark next to the exercise in the list. Some hints for creating a good exercise is found [here](#).

Anonymity and editing. The pages are publicly visible but can only be modified by students of KTH. The edit password can be obtained from <https://www.kth.se/social/course/SF2524/page/password-for-wiki>. Edits can be made anonymously. However, it is recommended to use a unique acronym when you edit pages, such that we can discuss progress in class. For instance, you can use your name initials, your KTH-ID or a nickname as acronym. The acronym is specified when you save your edits.

Save often. Several people might be editing the same question. In order to avoid your conflicts, it is better if you make small edits and save, rather than writing big chunks of text. Also, when you start a new exercise, you do not need to take the first available question slot.

Exercises

Expand all exercises, solutions and editors: [click here](#).

- A: Large sparse eigenvalue problems
 - Problems:
 - A1: Rayleigh quotient iteration question
 - A1solution
 - A2: Power method question



H - SCI

e lecturer

e students

e topic

e course



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 attendance in the lectures is important“
- “After the second lecture, I thought, wow this is totally different“

[About the lecturer](#)

[About the students](#)

[About the topic](#)

[About the course](#)

Time to start the lecture ...

SF2524
SF3580

Elias Jarlebring



KTH - SCI

About the lecturer

About the students

About the topic

About the course

Time to start the lecture ...

SF2524
SF3580

Elias Jarlebring



KTH - SCI

Fundamental eigenvalue techniques (block 1)

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

[About the lecturer](#)

[About the students](#)

[About the topic](#)

[About the course](#)



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

On the one hand, power iteration is of limited use, for several reasons. First, it only finds only the eigenvector corresponding to the largest eigenvalue. Second, the convergence is slow, halving the error only by a constant factor $\approx |\lambda_2/\lambda_1|$ at each iteration. Thirdly, the quality of the lines depends on having a large 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 squish the differences between eigenvalues.

Inverse Iteration

For any $\mu \in \mathbb{C}$ 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 eigenvalue is $(\lambda_i - \mu)^{-1}$, where λ_i are the eigenvalues of A . This suggests an idea: Suppose μ is close to an eigenvalue λ_j of A . Then $(\lambda_j - \mu)^{-1}$ will be much larger than $(\lambda_i - \mu)^{-1}$ for all $i \neq j$. Thus, if we apply power iteration to $(A - \mu I)^{-1}$, the process will converge rapidly to λ_j . This idea is called *inverse iteration*.

Algorithm 27.2. Inverse Iteration

$x^{(k)}$ is some vector with $\|x^{(k)}\| = 1$
for $k = 1, 2, \dots$
 Solve $(A - \mu I)x^{(k+1)} = x^{(k)}$ for $x^{(k+1)}$ apply $(A - \mu I)^{-1}$
 $\lambda^{(k+1)} = x^{(k+1)T} x^{(k)}$ *Rayleigh quotient*
 $x^{(k+1)} = x^{(k+1)} / \|x^{(k+1)}\|$ *Rayleigh quotient*

What if μ is an eigenvalue of A , so that $A - \mu I$ is singular? What if μ is not an eigenvalue, so that $A - \mu I$ is not diagonalizable (but an accurate solution of $(A - \mu I)x = x^{(k)}$ cannot be expected)? These apparent pitfalls of inverse iteration appear as traps in our Exercise 27.5.

Like power iteration, inverse iteration exhibits only linear convergence. Unlike power iteration, however, we can choose the direction that will be found by applying 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 an eigenvalue of A than to the others, then the largest eigenvalue of $(A - \mu I)^{-1}$ will be much larger than the others. Using this 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 $|\lambda_j - \mu| < |\lambda_k - \mu|$ for each $k \neq j$. Furthermore,

Algorithm 27.1. Power Iteration

$x^{(0)}$ is some vector with $\|x^{(0)}\| = 1$
for $k = 1, 2, \dots$
 $x^{(k)} = Ax^{(k-1)}$ apply A
 $\lambda^{(k)} = x^{(k)T} x^{(k-1)}$ *Rayleigh quotient*
 $x^{(k)} = x^{(k)} / \|x^{(k)}\|$ *Rayleigh quotient*

In this and the algorithm below, we give no attention to termination conditions, describing the loop only by the negative expression “for $k = 1, 2, \dots$ ”. Of course, in practice, termination conditions are very important, and this is one of the points where (especially) students who do not know how to use MATLAB or Mathematica likely to be required to program an individual MATLAB code.

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

$$x^{(k)} = \alpha_1 q_1 + \alpha_2 q_2 + \dots + \alpha_n q_n.$$

Since $x^{(0)}$ is a multiple of q_1 , we have the same constants α_i :

$$\begin{aligned} x^{(0)} &= \alpha_1 q_1 \\ x^{(1)} &= \lambda_1 \alpha_1 q_1 + \alpha_2 q_2 + \dots + \alpha_n q_n \\ x^{(2)} &= \lambda_1^2 \alpha_1 q_1 + \alpha_2 \lambda_2 q_2 + \dots + \alpha_n \lambda_n q_n \\ &\vdots \\ x^{(k)} &= \lambda_1^k \alpha_1 q_1 + \alpha_2 \lambda_2^k q_2 + \dots + \alpha_n \lambda_n^k q_n. \end{aligned} \quad (27.6)$$

From here we obtain the following conclusion:

Theorem 27.1. Suppose $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$ and $\alpha_1 \neq 0$. Then the limit of Algorithm 27.1 is

$$\begin{aligned} x^{(k)} &\rightarrow \alpha_1 q_1 = \frac{q_1}{\|q_1\|}, & \lambda^{(k)} &\rightarrow \lambda_1 = 0 \text{ and } \frac{q_1^T x^{(k)}}{\|x^{(k)}\|} \rightarrow \lambda_1. \end{aligned}$$

As $k \rightarrow \infty$, this λ says exactly that at each step k , one or the other choice of sign to be taken, and then the subsequent fixed value.

The first equation follows from (27.6), where $\alpha_i = \alpha_i \lambda_i^k / \lambda_1^k$ for k large enough. The second follows from this and (27.6). If $\lambda_1 = 0$, then the λ says an off $\pm \alpha_1 / \alpha_2$, whereas $\lambda_1 = 0$ does otherwise. \square

The λ in (27.6) and its similar equalities below are very surprising. There is an eigenvalue λ_j to avoid these complications, which is to speak of convergence of subspaces, not vectors – to say that $\|x^{(k)}\|$ converges to $\|q_j\|$ for

suppose $\lambda_j^{(k)} \neq \lambda_j$. Then the choice of Algorithm 27.2 which

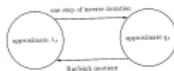
$$x^{(k+1)} = [Ax^{(k)} - \lambda_j x^{(k)}] / \left\| [Ax^{(k)} - \lambda_j x^{(k)}] \right\|, \quad (x^{(k)} - \lambda_j) = 0 \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda_i} \right)^k$$

as $k \rightarrow \infty$, where the λ_j says has the same meaning as in Theorem 27.1.

Inverse iteration is one of the most reliable tools of numerical linear algebra for the n -th standard method of computing one of your eigenvalues (if a matrix) if the eigenvalue is already known. In this case, Algorithm 27.2 is applied in series, since the the solution of the bounding quantity is dependent on k .

Rayleigh-Quotient Iteration

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



(The two are actually able to get from an eigenvalue λ_k to an eigenvalue λ_{k+1} in a step of inverse iteration, and also to a particular eigenvalue λ_j). This idea is so naturally 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

$x^{(k)}$ is some vector with $\|x^{(k)}\| = 1$
 $\lambda^{(k)} = x^{(k)T} A x^{(k)}$ is corresponding Rayleigh-quotient
for $k = 1, 2, \dots$
 Solve $(A - \lambda^{(k)} I)x^{(k+1)} = x^{(k)}$ for $x^{(k+1)}$ apply $(A - \lambda^{(k)} I)^{-1}$
 $\lambda^{(k+1)} = x^{(k+1)T} x^{(k)}$ *Rayleigh-quotient*
 $x^{(k+1)} = x^{(k+1)} / \|x^{(k+1)}\|$ *Rayleigh-quotient*