

Numerical Linear Algebra for Computational Science and Information Engineering

Lecture 00 Introduction

Nicolas Venkovic
nicolas.venkovic@tum.de

Group of Computational Mathematics
School of Computation, Information and Technology
Technical University of Munich

Summer 2025



Outline

- 1 Numerical linear algebra:
The "Why?", the "What?" and "What's next?" 1
- 2 Computational mathematics @ TUM 4
- 3 Class overview 6
- 4 Homework assignment 13

Numerical linear algebra: The "Why?", the "What?" and "What's next?"

Why do we need numerical linear algebra?

Prevalence of linear algebra in research and industry:

- Scientific computing (often explicitly related to problems of linear algebra)
- Simulations (e.g., discretized PDEs, ODEs, DAEs, ...)
- Machine learning (e.g., PCA, RKHS, ...)
- Optimization

Methods taught in linear algebra:

- Solving linear systems: Cramer's rule and Gaussian elimination
- Matrix factorizations: Cholesky, LU and QR
- Eigenvalue problems: Root-finding of characteristic polynomials

Limitations of methods taught in linear algebra:

- **No general exact method for eigenvalues of matrices** larger than 4×4
- **Computational infeasibility** of analytical methods **for large systems**
- **Numerical instability** leading to significant **errors**
- **Inability to handle special structures** (e.g., sparsity, implicitness)

**Need for methods to solve challenging problems
efficiently**

What is numerical linear algebra?

What does numerical linear algebra (NLA) address:

- Efficient algorithms for large-scale problems of linear algebra
- Techniques for maintaining numerical stability and accuracy
- Methods for exploiting matrix structure (e.g., sparsity, symmetry, low-rank)
- Matrix-free formulations for implicit operators
- Error analysis and conditioning

Scope of NLA:

- Focuses on both theoretical analysis (e.g., convergence, conditioning and stability) and practical implementation.
- Incorporates aspects of performance and computer architecture for method development, e.g.,
 - Repurposing solvers into preconditioners for higher performance.
 - Designing sparse data structures and BLAS kernels to optimize memory usage.

Key branches of NLA:

- **Direct methods** vs **iterative methods** (stationary vs Krylov)
- **Dense matrices** vs **sparse matrices** and **matrix-free operators**

What's coming next in numerical linear algebra?

Recent developments in NLA research show promise for significant performance improvements, but face challenges in widespread adoption:

- ▶ **Randomization:** Randomly reduces problem dimensions while preserving inherent structure with high probability. Associated challenges are
 - Ensuring convergence and assessing stability.
 - Defining and implementing reliably amortized sketching procedures.
- ▶ **Communication avoidance:** Redesigns algorithms to minimize data movement between processors or memory hierarchies. Challenges are
 - Maintaining numerical stability with reduced communication.
- ▶ **Mixed precision:** Utilizes different numerical precisions within a single algorithm to optimize performance. Associated challenges are
 - Ensuring overall stability and accuracy with lower precision components.
 - Developing adaptive strategies for precision selection.
- ▶ **Extension of state-of-the art methods to tensor data:** Assessing convergence and stability when using low-rank approximation of tensor data, e.g., Tucker and Tensor-Train formats.

Computational mathematics @ TUM

Computational mathematics @ TUM, Campus Heilbronn

- ▶ Chair of COmputational MAtematics (COMA) @ TUM: Prof. **Hartwig Anzt**
 - Professor @ TUM since 01/2024
 - Director of the Innovative Computing Lab @ University of Tennessee (2022-23)
 - Junior Professor @ KIT (2021-22)
- ▶ Recent and ongoing PhD theses directed by Hartwig:
 - Symbolic LU factorizations on GPUs (**Tobias Ribizel**, ongoing)
 - Scalable domain decomposition on GPUs (**Fritz Göbel**, ongoing)
 - Numerical compression in scientific computing (**Thomas Grützmacher**, ongoing)
 - Mixed precision algebraic multigrids on GPUs (**Yu-Hsiang Mike Tsai**, 2024)
 - Asynchronous and batched iterative solvers on GPUs (**Pratik Nayak**, 2023)
- ▶ Ongoing projects in COMA:
 - **Ginkgo**: Portable high-performance numerical linear algebra library
 - **Sparse BLAS**: Basic linear algebra subprograms for sparse matrices
 - **ICON**: Large legacy codebase for climate modeling
 - **OGL/OpenFOAM**: Large legacy codebase for computational fluid dynamics
 - **MicroCARD**: Numerical modeling of cardiac electrophysiology
 - **PDExa**: Optimized software methods and technologies for PDEs
 - **nekRS**: Fast and scalable computational fluid dynamics software package

Computational mathematics @ TUM, Campus Heilbronn, cont'd

► Course instructor:



Nicolas Venkovic, Postdoc. in COMA @ TUM

- Postdoctoral Researcher @ TUM since 07/2024
- Software Developer @ NXP Semiconductors (09/2022 - 04/2024)
- PhD in Applied Mathematics & Scientific Computing @ Cerfacs (09/2023)
- MSE in Applied Mathematics & Statistics @ Johns Hopkins University (2018)

► Recent and ongoing research efforts:

- Randomization of block iterative methods
- Low-rank approximation of integral operators
- Iterative methods for parameter-dependent problems

Class overview

Objectives

Upon completion of the class, students should:

- ▶ Understand why numerical linear algebra (NLA) is essential in practice, and what it encompasses.
- ▶ Be familiar with the challenges of the main problems and procedures of NLA, such as orthogonalization, least-squares solving, factorization, linear solving, preconditioning, eigenvalue solving, ...
- ▶ Understand the definitions, implementations and main properties of the methods presented in class for solving these problems.
- ▶ Recognize some pathological behaviors of these methods and be able to explain unwanted numerical behaviors.
- ▶ Know the strengths and limits of the presented methods, and know when to use them in practice.
- ▶ Have the foundational background to explore methods for solving NLA problems beyond those covered in class.

Content

- ▶ We will cover the following core topics:
 - Lecture 01: Essentials of linear algebra
 - Lecture 02: Essentials of the Julia language
 - Lecture 03: Floating-point arithmetic and error analysis
 - Lecture 04: Direct methods for dense linear systems
 - Lecture 05: Sparse data structures and basic linear algebra subprograms
 - Lecture 06: Introduction to direct methods for sparse linear systems
 - Lecture 07: Orthogonalization and least-squares problems
 - Lecture 08: Basic iterative methods for linear systems
 - Lecture 09: Basic iterative methods for eigenvalue problems
 - Lecture 10: Locally optimal block preconditioned conjugate gradient
 - Lecture 11: Arnoldi and Lanczos procedures
 - Lecture 12: Krylov subspace methods for linear systems
 - Lecture 13: Multigrid methods
 - Lecture 14: Preconditioned iterative methods for linear systems
 - Lecture 15: Restarted Krylov subspace methods
 - Lecture 16: Elements of randomized numerical linear algebra
 - Lecture 17: Introduction to communication-avoiding algorithms
 - Lecture 18: Matrix function evaluation

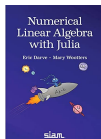
Content, cont'd

- ▶ If time permits, additional lectures may be picked among the following based on students' interests:
 - Lecture E1: More on eigensolvers, e.g., FEAST, Jacobi-Davidson, ...
 - Lecture E2: Multilinear algebra and tensor decompositions
 - Lecture E3: Introduction to mixed precision algorithms
- ▶ The duration of a lecture is not tied to a single class period. Some lectures will span less than one class, while others will cover multiple meetings. Several lectures can be covered within one meeting.
- ▶ Most lectures will begin by establishing formal foundations, and some will be followed by practice sessions using **Julia notebooks** or **exercise sheets** to test, visualize, and deepen your understanding of the methods and concepts introduced.

Reading material

► Main references:

- **Lecture slides** and **Julia notebooks** uploaded gradually on [Moodle](#), throughout the semester.
Content also uploaded at venkovic.github.io/NLA-for-CS-and-IE.html.



- Darve, E., & Wootters, M. (2021). Numerical linear algebra with Julia. Society for Industrial and Applied Mathematics (SIAM).[†]

- GitHub repository:
https://github.com/ericdarve/numerical_linear_algebra

► Most used supplemental reference:

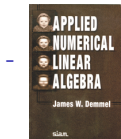


- Saad, Y. (2003). Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics (SIAM).[†]

[†] SIAM members get 30% off book prices. SIAM membership is free for students.

Reading material, cont'd₁

► Other useful supplemental references:



- Demmel, J. W. (1997). Applied numerical linear algebra. Society for Industrial and Applied Mathematics (SIAM).[†]



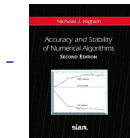
- Golub, G. H., & Van Loan, C. F. (2013). Matrix computations. JHU press.



- Greenbaum, A. (1997). Iterative methods for solving linear systems. Society for Industrial and Applied Mathematics (SIAM).[†]

[†] SIAM members get 30% off book prices. SIAM membership is free for students.

Reading material, cont'd₂



- Higham, N. J. (2002). Accuracy and stability of numerical algorithms. Society for Industrial and Applied Mathematics (SIAM).[†]



- Trefethen, L. N., & Bau, D. (2022). Numerical linear algebra. Society for Industrial and Applied Mathematics (SIAM).[†]

[†] SIAM members get 30% off book prices. SIAM membership is free for students.

Course evaluation

► Final exam (determines base grade):

10:00-11:30
Thursday, 31 July 2025
Campus Heilbronn
D.2.01

- Questions in relation to the homework problems, material presented in class, and practice sessions.

► Homework policy:

- Submit your solution to one problem per lecture.
- Deadline: 2 weeks after the end of the lecture.
- Regular submissions upgrade the final grade by one step.
 - Example: 2.0 becomes 1.7.
 - Notes: This upgrade applies to passing grades only.
1.0 remains the highest possible grade.

Course evaluation

► Retake exam:

10:00-11:30
Monday, 29 September 2025
Campus Heilbronn
D.2.01

- Questions in relation to the homework problems, material presented in class, and practice sessions.

► Homework policy:

- Submit your solution to one problem per lecture.
- Deadline: 2 weeks after the end of the lecture.
- Regular submissions upgrade the final grade by one step.
 - Example: 2.0 becomes 1.7.
 - Notes: This upgrade applies to passing grades only.
1.0 remains the highest possible grade.

Homework assignment

Homework assignment

Send an email to me with the subject line NLA-YourLastName addressing the following points:

- ① Briefly describe your background, if any, in numerical linear algebra, e.g.,
 - Courses taken, practical experience, self-study, ...
- ② Identify a specific area of interest in numerical linear algebra:
 - A problem, method or concept you either work on, want to understand better, or are curious about.
- ③ Regarding the course syllabus:
 - List 2-3 topics you're most excited to learn about.
 - Mention any topics you feel you already have a strong grasp of, or simply would not mind skipping.
- ④ Suggest elective topics, listed in the slides or not, that you'd like to see covered, in case time permits.

Note: Your responses could help adapt certain aspects of the course to the class's needs and interests.