

Course number		G-LAS12 80038 SE76					
Course title (and course title in English)		Mathematics and Numerical Computing Mathematics and Numerical Computing		Instructor's name, job title, and department of affiliation		Graduate School of Engineering Professor,SOTOWA KENICHIRO	
Group		Interdisciplinary Graduate Courses		Field(Classification)		Statistics, Informatics and Data Science	
Language of instruction		English		Old group		Number of credits 1.5	
Number of weekly time blocks		1		Class style Seminar (Face-to-face course)		Year/semesters 2026 • Second semester	
Days and periods		Thu.4		Target year Graduate students		Eligible students For science students	
(Students of Graduate School of Engineering cannot take this course as liberal arts and general education course. Please register the course with your department.)							
[Overview and purpose of the course]							
This course aims to develop students' proficiency in numerical computation and scientific programming using Python, with a particular emphasis on eigenvalue problems and their computational methods. The course is structured in two progressive parts: the first focuses on building Python programming skills and essential mathematical tools (arithmetic operations, class design, polynomials, and orthogonal functions), while the second half concentrates on eigenvalue problems#8212their mathematical foundations, various computational algorithms, and practical applications. Through hands-on programming exercises and project work, students will gain deep understanding of eigenvalue computations and learn to apply these concepts to solve real-world problems in engineering and science.							
[Course objectives]							
Upon completion of this course, students will be able to: 1. Develop practical Python programming skills for scientific computing through implementing numerical algorithms and designing custom mathematical classes (fractions, polynomials) 2. Develop a comprehensive understanding of eigenvalue problems#8212including their mathematical foundations, geometric interpretations, and fundamental role in scientific computation#8212and appreciate the deep connections between theory and applications 3. Implement and apply various eigenvalue computation methods to solve practical problems such as polynomial equation solving, data analysis, and dimensionality reduction (PCA)							
[Course schedule and contents)]							
(1) Introduction to Python - Python programming environment, basic syntax, variable types, control structures (loops, conditionals), and simple input/output (2) Basic Arithmetics Using Python - Arithmetic operations, numerical precision, handling floating-point errors, and basic mathematical functions from NumPy (3) Manipulating Fractions Using Python Class - Design and implementation of a custom Fraction class; operator overloading; practical applications in exact arithmetic							
Continue to Mathematics and Numerical Computing(2) ↓ ↓ ↓							

Mathematics and Numerical Computing(2)

- (4) Python Class for Handling Polynomials - Design and implementation of a Polynomial class; polynomial operations (addition, multiplication, evaluation); symbolic and numerical representations
- (5) Orthogonal Functions - Introduction to orthogonal function systems; properties of orthogonal polynomials (e.g., Legendre, Chebyshev); applications in numerical analysis and approximation theory
- (6) Introduction to Eigenvalue Problem - Fundamental concepts of eigenvalues and eigenvectors; characteristic polynomial; geometric and physical interpretations; relationship to linear transformations
- (7) Estimation of Eigenvalues - Gershgorin's theorem, power iteration method, inverse power iteration method, and deflation method for computing individual eigenvalues
- (8) QR Methods - QR decomposition and QR algorithm for simultaneous computation of all eigenvalues; convergence properties; practical implementation in Python
- (9) Application of Eigenvalue to Solution of Higher Order Equations - Relationship between polynomial roots and matrix eigenvalues; companion matrix method; finding all solutions of polynomial equations using eigenvalue algorithms
- (10) Application of Eigenvalue: Principal Component Analysis (PCA) - Theoretical foundations of PCA; dimensionality reduction; variance maximization; practical applications in data analysis, feature extraction, and image processing
- (11) Final Presentation - Each student presents a computational project involving eigenvalue problems or their applications, demonstrating understanding of both theory and implementation

[Course requirements]

None

[Evaluation methods and policy]

Attendance, assignment and presentation will count.

[Textbooks]

Detailed lecture notes and handouts will be provided for each session.

[References, etc.]

(References, etc.)

Introduced during class

[Study outside of class (preparation and review)]

Assignments will be given as necessary.

[Other information (office hours, etc.)]

This course is offered every other year. Students are encouraged to start planning their final presentation topics by mid-semester.

[Essential courses]