II. Numerical Algebra

A. Lectures 10-12: LLMs and Linear Algebra

Things you should know

Lecture 10 – Review of Matrix Algebra

- Whether a system of equations is linear or nonlinear
- Vector notation and index notation

Lecture 11 – Gauss Elimination

- How the Gauss Elimination algorithm works

Lecture 12 – Numpy linear algebra

- Concept of computational cost in terms of memory and CPU time

Things you should be able to do

Lecture 10 – Review of Matrix Algebra

- Calculate vector norms, dot products and matrix products by hand and using Python
- Convert a system of linear equations to matrix notation
- Solve a linear system using the Gauss elimination algorithm by hand

Lecture 11 – Gauss Elimination

- Write or debug a Python code that performs Gauss elimination (forward elimination and back substitution)

Lecture 12 – Numpy linear algebra

- Calculate the number of bytes a variable or program uses
- Calculate the asymptotic behavior and approximate running time of a code
- Use Numpy functions (`add`, `subtract`, `multiply`, `transpose`, `norm`) to manipulate matrices
- Solve a system of linear equations using Python's linear algebra package (`np.linalg.inv`, `np.lingalg.solve`)

B. Lectures 13-16: Nonlinear Algebra

Things you should know

Lecture 13 – Fixed Point Methods

- Why nonlinear equations are more difficult to solve than linear equations (no guaranteed or unique solution)
- Identify types of nonlinear equations (implicit equation, polynomials)
- The concept of a fixed-point method, i.e. iterative $x^{(k+1)} = g(x^{(k)})$
- Picard's method and Newton's method: formulas and pros/cons

Lecture 14 – NLEs with SciPy

- What is in the SciPy module

Lecture 15 – Optimization

- Difference between fixed-point methods and optimization (minimum of the squared error or residual)

Lecture 16 – Engineering Problems

- Strategies to solve systems of linear/nonlinear engineering equations (units, vector notation, good guesses)

Things you should be able to do

Lecture 13 – Fixed Point Methods

- Write a single nonlinear equation in standard/residual form
- Solve a single nonlinear equation by Picard's or Netwon's method by hand, in Excel and in Python

Lecture 14 – NLEs with SciPy

- Write a system of nonlinear equations in standard/residual form in vector notation
- Solve a single NLE or a system of NLEs using `scipy.optimize.root`

Lecture 15 – Optimization

- Solve a single NLE or a system of NLEs using Excel's Solver
- Solve a single NLE or a system of NLEs via `scipy.optimize.minimize`

Lecture 16 – Engineering Problems

- Solve a system of linear/nonlinear engineering equations (with units, etc.) using root-finding or optimization methods in Excel and Python