Content

Course Code	Course Name	Semester	Theory	Practice	Lab	Credit	ECTS
ECON254	Numerical Methods and Analysis	4	0	0	3	3	5

Prerequisites	ECON151
Admission Requirements	ECON151

Language of Instruction	Turkish
Course Type	Compulsory
Course Level	Bachelor Degree
Objective	Most real-world economic models cannot be solved analytically and lack closed-form solutions. ECON 254 bridges the gap between economic theory and computational implementation. This course equips advanced undergraduate students with the skills to solve static and dynamic economic models using fundamental numerical methods and the Python programming language. The content progresses from basic numerical techniques (linear algebra, interpolation, root-finding) to advanced optimization (direct NLP solvers, indirect Lagrange/KKT approaches), and from there to dynamic programming and stochastic methods (backward induction, Value Function Iteration, Markov processes). Throughout the course, theoretical concepts are anchored in core economic frameworks; students learn not only to code algorithms but also to understand the underlying economic mechanisms, to perform convergence and error analysis, and to build reproducible analytical workflows. The last two weeks are devoted to group project presentations, in which student teams numerically formulate, solve, and defend a chosen economic optimization problem.
Content	The course begins with the setup of the Python scientific computing ecosystem (NumPy, SciPy, Matplotlib) and vectorized operations, followed by interpolation and function approximation. It then covers linear equation systems (Ax=b) and matrix decomposition techniques, applied to multi-sectoral economic networks such as Leontief input-output analysis. Fixed-point iteration, root-finding methods (bisection, Newton-Raphson), and nonlinear equation systems (multivariate Newton-Raphson, Jacobian) are used to solve market-clearing prices and multi-good consumer optimization. In the second half of the course, unconstrained optimization (gradient, Hessian, BFGS), constrained optimization (direct NLP solvers and the indirect Lagrange/KKT approach), finite-horizon dynamic problems (Cake-eating, Life-Cycle), and the stochastic Value Function Iteration (VFI) along with the Income Fluctuation Problem are covered. The last two weeks are reserved for project presentations in which student groups present a chosen optimization problem.
References	1. QuantEcon: \url{https://quantecon.org/} (Primary platform for Python-based economics education) 2. Iskhakov, F. \textit{Foundations of Computational Economics}. \\ \url{https://fedor.iskh.me/compecon} 3. Additional resources and tools you may find useful: - SciPy documentation: https://docs.scipy.org/doc/scipy/ - Matplotlib documentation: https://matplotlib.org/stable/ - Jupyter: https://jupyter.org/

Theory Topics

Week	Weekly Contents
1	Python Scientific Computing Environment. Application: Symbolic and numerical comparative solution of a simple supply-demand function.
2	Interpolation and Function Approximation. Application: Yield curve construction; continuous representation of demand/policy functions.
3	Linear Equation Systems and Matrix Solutions. Application: Leontief input-output analysis; intersectoral multiplier effects and shock propagation.
4	Fixed-Point Iteration. Application: Numerical equilibrium solution of the IS-LM model; solution of the Solow model.
5	Root-Finding Methods. Application: Numerical calculation of the market-clearing price; profit maximization in a monopolistic market.
6	Solving Nonlinear Equation Systems. Application: Consumer optimization in the multi-good case; cost minimization.
7	Unconstrained Optimization Methods. Application: Firm profit maximization; derivation of factor demand functions.
8	Midterm

Week	Weekly Contents
9	Constrained Optimization: Direct Methods. Application: Consumer utility maximization via direct NLP.
10	Constrained Optimization: Indirect Methods. Application: Formulation and numerical solution of the KKT system in consumer optimization.
11	Finite-Horizon Dynamic Problems. Application: Cake-eating problem.
12	Life-Cycle Model. Application: Optimal consumption-savings plan between ages 25 and 80.
13	PROJECT PRESENTATIONS I.
14	PROJECT PRESENTATIONS II.