**The Hong Kong University of Science and Technology**

**Course Syllabus**

**Optimization in Financial Engineering**

IEDA 4000H

3 Credits

Pre-/co-requisites: IEDA 3010 Prescriptive Analytics

**Name:** Zijie Zhou

**Email:** jerryzhou@ust.hk

**Office Hours:** Mondays 2-3pm

**Course Description**

This course introduces fundamental optimization techniques and their applications in financial engineering. Students will learn how to model and solve financial problems using linear programming, convex optimization, and robust optimization. Topics include portfolio selection, sensitivity analysis, risk management, option pricing, currency exchange, and optimal execution. The course emphasizes practical implementation with Python solvers and case studies to bridge theory and real-world applications.

Key topics covered are:

- Convex Optimization: Applications in risk-return trade-offs and Markowitz portfolio formulation.
- Robust Optimization: Worst-case risk analysis for handling market uncertainty.
- Financial Applications: Option pricing, optimal execution, bond pricing, and model predictive control.

**Intended Learning Outcomes (ILOs)**

By the end of this course, students should be able to:

1. Formulate and solve financial optimization problems – Students will be able to model real-world financial scenarios (e.g., portfolio selection, risk management) using linear and convex optimization techniques.

2. Implement optimization models computationally – Students will gain hands-on experience using Python solvers (e.g., CVXPY, SciPy) to analyze and optimize financial decision-making processes.

3. Analyze sensitivity and robustness in financial models – Students will learn to assess how changes in market conditions impact optimization results and apply robust optimization to mitigate uncertainty.

4. Apply optimization techniques to advanced financial topics – Students will explore applications in option pricing, optimal execution, bond pricing, and predictive control, connecting theory to practical financial engineering challenges.

**Assessment and Grading**

This course will be assessed using criterion-referencing and grades will not be assigned using a curve. Detailed rubrics for each assignment are provided below, outlining the criteria used for evaluation.

**Assessments:**

- There are 4 assignments. Each of them has the same weight.
- There is an in-person midterm exam. The time limit is 90 minutes. It will test whether you understand the first half of course materials or not.
- There is a final (group) project. The score is based on the quality of the report and the final presentation.
- Grade percentage: Assignments 30%; midterm exam 30%; final project 40%.

**Summary Table:**

| Assessment Task | Contribution to Overall Course grade (%) |
|---|---|
| Mid-Term | 30% |
| Assignments | 30% |
| Final project & presentation | 40% |

**Mapping of Course ILOs to Assessment Tasks**

| Assessed Task | Mapped ILOs | Explanation |
|---|---|---|
| Mid-Term | ILO1, ILO3 | This task assesses students' ability to formulate financial optimization problems using linear and convex optimization techniques (ILO1) and analyze sensitivity/robustness in financial models under market uncertainty (ILO3). The exam focuses on theoretical understanding and problem-solving without computational implementation. |
| Assignments | ILO1, ILO2, ILO3. ILO4 | These tasks evaluate students' proficiency in modeling financial scenarios (e.g., portfolio selection) with optimization techniques (ILO1), implementing models computationally using Python solvers like CVXPY (ILO2), assessing robustness against market changes (ILO3), and applying techniques to advanced topics (e.g., option pricing) |

| | | |
|---|---|---|
| | | (ILO4). Assignments bridge theory and practical implementation. |
| Final project & presentation | ILO1, ILO2, ILO3. ILO4 | The project assesses students' ability to integrate course concepts by formulating a real-world financial optimization problem (ILO1), implementing a computational solution (ILO2), analyzing model sensitivity/robustness (ILO3), and presenting applications to advanced topics (e.g., optimal execution or bond pricing) (ILO4). The presentation evaluates clarity, depth, and practical relevance. |

**Grading Rubrics**

Use the following rubrics to guide you for the assessment tasks that you submit in this course.

**Midterm Exam Rubric**

| Criteria | Excellent (A) | Good (B) | Satisfactory (C) | Marginal (D) | Fail (F) | Mapping to ILOs |
|---|---|---|---|---|---|---|
| Problem Formulation | Accurately defines financial optimization problems (e.g., Markowitz portfolio) with clear assumptions and context. | Defines problems correctly but lacks depth in assumptions or financial context. | Basic problem formulation with minor errors or oversimplifications. | Problem statement is vague or missing key elements (e.g., constraints). | Fails to formulate the problem or misinterprets requirements. | ILO1 |
| Mathematical Rigor | Derives solutions with precise mathematical reasoning (e.g., Lagrange multipliers, convexity proofs). | Correctly applies methods but with minor logical gaps or notation errors. | Uses valid techniques but struggles with derivations or justification. | Limited mathematical coherence; major errors in methodology. | No valid mathematical approach or solution. | ILO1, ILO3 |
| Sensitivity Analysis | Thoroughly analyzes how changes in | Identifies key sensitivities but lacks depth in | Basic sensitivity discussion with superficial insights. | Minimal or incorrect sensitivity analysis. | No analysis of parameter impacts. | ILO3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | parameters (e.g., risk tolerance) impact optimal solutions. | interpretation. | | | | |
| Clarity & Organization | Solutions are logically structured, notationally consistent, and professionally presented. | Clear presentation with minor organizational issues. | Disorganized or unclear in parts but readable. | Poorly structured; difficult to follow. | Incoherent or illegible work. | -- |

**Assignments Rubric**

| Criteria | Excellent (A) | Good (B) | Satisfactory (C) | Marginal (D) | Fail (F) | Mapping to ILOs |
|---|---|---|---|---|---|---|
| Theoretical Solutions | Correctly formulates problems and justifies methods with rigor (e.g., duality, KKT conditions). | Valid solutions but limited justification or minor errors. | Solutions are functionally correct but lack depth or contain errors. | Major flaws in problem setup or methodology. | No valid theoretical solution. | ILO1, ILO3 |
| Coding Implementation | Efficient Python code (CVXPY/SciPy) with robust error handling, visualization, and documentation. | Functional code with minor inefficiencies or sparse comments. | Code runs but lacks optimization or clarity (e.g., hardcoded values). | Partial or buggy implementation; no documentation. | No working code or irrelevant submission. | ILO2 |
| Analysis & Insights | Deep sensitivity analysis, robustness checks, and real-world interpretations (e.g., trade-offs). | Adequate analysis but limited exploration of edge cases. | Basic results summary without critical insights. | Superficial or incorrect conclusions. | No analysis provided. | ILO3, ILO4 |

**Final Presentation Rubric**

| Criteria | Excellent (A) | Good (B) | Satisfactory (C) | Marginal (D) | Fail (F) | Mapping to ILOs |
|---|---|---|---|---|---|---|
| Problem Relevance | Addresses a complex, real-world financial problem with clear motivation and scope. | Practical problem but narrower scope or less innovative. | Generic problem with limited novelty or applicability. | Problem is trivial or lacks financial relevance. | No identifiable problem or off-topic. | ILO1, ILO4 |
| Technical Depth | Combines advanced optimization techniques (e.g., robust MPC) with rigorous analysis. | Appropriate methods but simpler analysis (e.g., basic convex optimization). | Standard techniques with superficial analysis. | Overly simplistic or incorrect technical approach. | No valid technical content. | ILO1, ILO2, ILO3 |
| Computational Quality | Polished Python implementation with scalable design, visualizations, and reproducibility. | Functional code with minor bugs or lacking scalability. | Code meets requirements but is messy or undocumented. | Partial or non-functional code. | No code or irrelevance. | ILO2 |
| Presentation Clarity | Engaging delivery, professional slides, and clear articulation of trade-offs/insights. | Clear presentation with minor pacing or clarity issues. | Basic delivery; struggles to explain technical details. | Disorganized or unclear presentation. | No coherent presentation. | ILO4 |
| Q&A Performance | Confidently addresses questions, defends methodology, and discusses limitations. | Answers questions adequately but lacks depth. | Basic responses with gaps in understanding. | Unable to answer most technical questions. | No meaningful engagement. | ILO3, ILO4 |

**Final Grade Descriptors:**

| **Grades** | **Short Description** | **Elaboration on subject grading description** |
|---|---|---|
| A | Excellent Performance | Demonstrates mastery in formulating and solving financial optimization problems (e.g., portfolio selection, robust risk analysis) with precision. Implements Python solvers (CVXPY/SciPy) efficiently and creatively, exceeding assignment requirements. Exhibits deep sensitivity analysis and robustness |

| | | |
|---|---|---|
| | | insights, with exceptional clarity in connecting theory to real-world applications (e.g., option pricing, optimal execution). Final project showcases originality, rigor, and professional-grade solutions. |
| B | Good Performance | Shows strong understanding of optimization techniques (linear/convex) and their financial applications. Competently implements models in Python, meeting assignment criteria. Analyzes market uncertainty and sensitivity adequately but with minor gaps. Final project demonstrates solid analysis and execution, though with limited innovation or depth in advanced topics (e.g., bond pricing). |
| C | Satisfactory Performance | Achieves basic competency in modeling financial problems and using optimization tools. Python implementations are functional but lack efficiency or elegance. Struggles with robustness analysis or advanced applications. Final project addresses core requirements but lacks thoroughness or real-world relevance. |
| D | Marginal Pass | Displays threshold knowledge of optimization concepts (e.g., Markowitz portfolio) but struggles with formulations or computational implementation. Limited ability to assess sensitivity or adapt models to uncertainty. Final project meets minimal standards but contains significant errors or omissions. |
| F | Fail | Fails to model or solve basic financial optimization problems. Python implementations are incomplete or non-functional. No meaningful analysis of robustness or financial applications. Final project is inadequate or missing. Does not meet course learning outcomes. |

**Course AI Policy**

The use of Generative AI is permitted and requested to assist students with brainstorming, drafting, and writing their papers.

**Communication and Feedback**

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Feedback on assignments will include comments on strengths and areas for improvement. Students who have further questions about the feedback including marks should consult the instructor within five working days after the feedback is received.

**Required Texts and Materials**

Convex Optimization, Stephen Boyd and Lieven Vandenberghe, Cambridge University Press (Publicly available online)

Daniel P. Palomar (2025). Portfolio Optimization: Theory and Application. Cambridge University Press. [portfoliooptimizationbook.com]

Yiyong Feng and Daniel P. Palomar. A Signal Processing Perspective on Financial Engineering. Foundations and Trends in Signal Processing, Now Publishers, 2016.
[https://palomar.home.ece.ust.hk/papers/2016/Feng&Palomar-FnT2016.pdf]

**Academic Integrity**

Students are expected to adhere to the university's academic integrity policy. Students are expected to uphold HKUST's Academic Honor Code and to maintain the highest standards of academic integrity. The University has zero tolerance of academic misconduct. Please refer to Academic Integrity | HKUST – Academic Registry for the University's definition of plagiarism and ways to avoid cheating and plagiarism.