Course and Examination Fact Sheet: Autumn Semester 2015

7,626: Solving Economics and Finance Problems with MATLAB

ECTS credits: 3

Overview examination/s
(binding regulations see below)
Decentral - Group examination paper (all given the same grades) (50%)
Decentral - Group examination paper with presentation (all given the same grades) (50%)

Attached courses
Timetable - Language - Lecturer
7,626.1.01 Solving Economics and Finance Problems with MATLAB, Group 1 - English - Gruber Peter
7,626.1.02 Solving Economics and Finance Problems with MATLAB, Group 2 - English - Gruber Peter

Course information

Course prerequisites
- Knowledge of bachelor level mathematics, statistics, econometrics and finance.
- Interest in quantitative methods.
- No programming experience is required.
- This course is targeted at MiQE/F, MEcon and MBF students. All other students are welcome, but may need extra time to study the prerequisites.
- Note that group 1 (08:15-10:00) is a laptop class, so you need a laptop with MATLAB installed (see below). No computer and no installation is required for group 2 (10:15-12:00).

Course content

Important:
Group 1 (Tue 08:15-10:00) is organized as a laptop class. Students are expected to bring their own computer with MATLAB installed (see below for details).
Group 2 (Tue 10:15-12:00) takes place in the PC lab. Students can use the version of MATLAB installed on the computers in the PC lab.

Many interesting problems in economics and finance can only be solved with the help of the computer, as no analytical solutions exist. This course gives an introduction to solving problems numerically with the help of MATLAB, an easy to learn, powerful and widely used programming environment.

The course has three equally important goals:
- Learn to use the most important features of the MATLAB programming environment
- Understand the differences between analytical and numerical problem solving
- Learn suitable algorithms for important problems in economics and finance.

This course embraces learning by doing as much as possible, which may result in an above-average workload. Starting from this semester, grading will be based on a portfolio system. Students build their portfolio by solving problem sets during the semester and by finishing a programming project (with presentation), in groups of two students.

After this course, students should be able to employ numerical methods for their master thesis and future research as well as to reproduce the numerical procedures of a current research paper in their field of specialization. This course is organized by methods rather than by applications, however throughout the course examples from economics and finance will be presented.

Students who wish to own a copy MATLAB (e.g. for the laptop class) can buy a student version at a discounted price of ca. CHF 100. Please make sure to include the toolboxes as stated below. Alternatively, students can work on their projects in the PC lab.

Course structure

The course is organized in 12 classes. Each class is divided into three parts to take students from theory to learning by doing:
- Every block starts with a lecture.
- This is followed by practical exercises, where students follow instructions to make their first steps with the new concepts.
After this, students solve problem sets in self-study as homework. These problem sets are usually discussed in the subsequent lecture.

The following topics will be covered:

1. Introduction to numerical methods and MATLAB
   How computers calculate: floating point numbers
   How problems are solved: an introduction to numerical algorithms
   An interactive introduction to MATLAB: variables, operators, statements, functions
   The first MATLAB program

2. Introduction to programming with MATLAB
   Control flow: choice, loops and more operators
   Functions (built-in and user defined)
   Good programming style
   Graphical output

3. Working with data, some econometrics and linear algebra
   Data input, output, manipulation, cleaning and visualization
   Manually programming the OLS estimator
   An introduction to the (open source) Financial Econometrics Toolbox
   Introduction to nonparametric estimation
   Some convenient linear algebra methods

4. Simulation techniques
   Random number generation
   Simulating time series processes
   An introduction to the Monte Carlo method

5. Nonlinear functions and algorithms
   Root finding, inverse functions
   Iterative and recursive algorithms
   Polynomials, approximation and interpolation
   General properties of iterative algorithms

6. Optimization
   Metrics
   Convex optimization
   Non-convex and stochastic optimization methods

7. Numerical integration and differentiation
   Numeric integration
   Transform methods (FFT, Fourier-cosine method)
   Numeric differentiation and precision

8. Good programming style
   Rule for good programming, how to avoid errors, how to collaborate with others
   Debugging programs

9. Advanced topics (selection, time permitting)
   Accelerating programs
   The symbolic mathematics toolbox
   An introduction to parallel computing
   An introduction to databases

The program can be adapted to accommodate special requests by students

Only the current timetable as published on Stundenplan Online does apply.

Contextual Studies are considered part of Contact Learning; thus, taking part properly implies regular attendance. It is the students’ own responsibility to ensure that there is no timetable clash between the courses they have chosen.

**Course literature**

- There is a 200-page script for the course.
- All transparencies, sample programs and relevant literature will be made available on the StudyNet.
- Students wishing to prepare for the course can have a look at the “MATLAB Getting Started GUIDE” and some training videos, available at the MATHWORKS homepage [http://www.mathworks.com/access/helpdesk/help/techdoc/](http://www.mathworks.com/access/helpdesk/help/techdoc/)
A selection of the vast choice of MATLAB books will be discussed in the first lectures.

**Additional course information**

**Minimum MATLAB configuration**
- MATLAB 2014b or later
- Optimization Toolbox
- Symbolic Math Toolbox

* Students may use older versions of MATLAB at their own risk. The main difference is the user interface and graphics engine.

**Examination information**

**Examination sub part/s**

1. **Examination sub part (1/2)**

   **Examination time and form**
   - Decentral - Group examination paper (all given the same grades) (50%)

   **Remark**
   - Problem sets in groups of 2 students

   **Examination-aid rule**
   - Term papers
     - Term papers must be written without anyone else's help and in accordance with the known quotation standards, and they must contain a declaration of authorship.
     - The documentation of sources (quotations, bibliography) has to be done throughout and consistently in accordance with the APA or MLA standards. The indications of the sources of information taken over verbatim or in paraphrase (quotations) must be integrated into the text in accordance with the precepts of the applicable quotation standard, while informative and bibliographical notes must be added as footnotes (recommendations and standards can be found, for example, in METZGER, C. (2013), Lern- und Arbeitsstrategien (11th ed., 3rd printing). Aarau: Sauerländer).
     - For any work written at the HSG, the indication of the page numbers both according to the MLA and the APA standard is never optional.
     - Where there are no page numbers in sources, precise references must be provided in a different way: titles of chapters or sections, section numbers, acts, scenes, verses, etc.
     - For papers in law, the legal standard is recommended (by way of example, cf. FORSTMOSER, P., OGOREK R. et SCHINDLER B. (2014, Juristisches Arbeiten: Eine Anleitung für Studierende (5. Auflage), Zürich: Schulthess, or the recommendations of the Law School).

   **Supplementary aids**
   - Examination languages
     - Question language: English
     - Answer language: English

2. **Examination sub part (2/2)**

   **Examination time and form**
   - Decentral - Group examination paper with presentation (all given the same grades) (50%)

   **Remark**
   - Large programming project in groups of 2-3 students

   **Examination-aid rule**
   - Term papers
     - Term papers must be written without anyone else's help and in accordance with the known quotation standards, and they must contain a declaration of authorship.
     - The documentation of sources (quotations, bibliography) has to be done throughout and consistently in accordance with the APA or MLA standards. The indications of the sources of information taken over verbatim or in paraphrase (quotations) must be integrated into the text in accordance with the precepts of the applicable quotation standard, while informative and bibliographical notes must be added as footnotes (recommendations and standards can be found, for example, in METZGER, C. (2013), Lern- und Arbeitsstrategien (11th ed., 3rd printing). Aarau: Sauerländer).
     - For any work written at the HSG, the indication of the page numbers both according to the MLA and the APA standard is never optional.
     - Where there are no page numbers in sources, precise references must be provided in a different way: titles of chapters or sections, section numbers, acts, scenes, verses, etc.
numbers, acts, scenes, verses, etc.

- For papers in law, the legal standard is recommended (by way of example, cf. FORSTMOSER, P., OGOREK R. et SCHINDLER B. (2014, Juristisches Arbeiten: Eine Anleitung für Studierende (5. Auflage), Zürich: Schulthess, or the recommendations of the Law School).

**Supplementary aids**

- **Examination languages**
  - Question language: English
  - Answer language: English

---

**Examination content**

The **problem sets** aim at putting the recently studied material into practice. They comprise the writing of one or more small programs plus possibly a short summary of a topic from the script, from a book or a similar source. The idea is that the solutions will form a portfolio of small programs that can be of later use to the student. These problem sets are to be solved in **groups of two students**. Problem sets are usually due on Sunday before the next lecture. Grading will be based on the best (n-1) problem sets to cover all special cases (sickness, military service, technical problems, forgotten to hand in...)

The **programming project** is a larger project that aims at applying several of the techniques presented in the course to one topic (e.g. option pricing, asset pricing, portfolio management, monetary policy, growth theory, econometrics and so on). Students can choose from a selection of topics or suggest their own projects. The programming project involves writing one program (possibly with a few support functions), a five-page user documentation and presenting the project. They are solved in **groups of two to three students**.

For both parts, the grading is based on the following factors:

- Technical correctness of the program code
- Mathematical/economical correctness of the solution
- Generality of the solution
- Conformity to the rules of good programming style
- Usability

For the programming project, the following additional criteria apply:

- Difficulty of the problem and depth of the solution
- Quality of the presentation
- Quality of the user documentation
- Level of understanding in the presentation and discussion

The user interface and the graphical interface design will not be graded.

**Problem sets and projects will only be graded fully if they have been handed in the correct way** (i.e. follow the instruction on naming files, structure of the solution and where to send it).

**Examination relevant literature**

This course more is about mastering a technique than reproducing facts, therefore any literature can only be indicative.

- The **script**, the **slides** and all **sample programs** as published on the StudyNet (published by November 17th at latest).
- All **original literature** published on the StudyNet, except those marked “for further reading”.
- Additional required literature will depend on the topic chosen for the programming project.
Please note
We would like to point out to you that this fact sheet has absolute priority over other information such as StudyNet, faculty members’ personal databases, information provided in lectures, etc.

When will the fact sheets become binding?

- Information about courses and examination time (central/decentral and grading form): from the start of the bidding process on 20 August 2015
- Information about decentral examinations (examination-aid rule, examination content, examination relevant literature): after the 4th semester week on 12 October 2015
- Information about central examinations (examination-aid rule, examination content, examination relevant literature): from the start of the enrolment period for the examinations on 02 November 2015

Please look at the fact sheet once more after these deadlines have expired.