Course guide
205277 - APMC++ - Application of Python/Matlab/C++ to Thermal Engineering Mechanical and Aeronautical Problems

Last modified: 11/04/2025

Unit in charge: Terrassa School of Industrial, Aerospace and Audiovisual Engineering
Teaching unit: 724 - MMT - Department of Heat Engines.

Degree:
BACHELOR'S DEGREE IN AUDIOVISUAL SYSTEMS ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN CHEMICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN ELECTRICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN INDUSTRIAL ELECTRONICS AND AUTOMATIC CONTROL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN MECHANICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN TEXTILE TECHNOLOGY AND DESIGN ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN AEROSPACE TECHNOLOGY ENGINEERING (Syllabus 2010). (Optional subject).
BACHELOR'S DEGREE IN AEROSPACE VEHICLE ENGINEERING (Syllabus 2010). (Optional subject).
BACHELOR'S DEGREE IN INDUSTRIAL DESIGN AND PRODUCT DEVELOPMENT ENGINEERING (Syllabus 2010). (Optional subject).
BACHELOR'S DEGREE IN INDUSTRIAL TECHNOLOGY ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2025 **ECTS Credits:** 3.0 **Languages:** English

LECTURER

Coordinating lecturer: Joaquim Rigola

Others:
Jordi Vera Fernández
Carles Oliet
Yolanda Calventus

PRIOR SKILLS

Thermodynamics, Heat Transfer and Fluid Dynamics

REQUIREMENTS

Labtop computer

TEACHING METHODOLOGY

The course is based in 3 different methodologies:

- Theoretical sessions
- Practical sessions
- Self-study for doing exercises and activities.

In the theory sessions, teachers will introduce the theoretical basis of numerical methods, programming and data management applied to thermal and fluid dynamics engineering problems. Some sessions might use a specific programming language (C++, MATLAB and PYTHON) to introduce the student to basic concepts, and specific aspects like object-oriented programming and post-processing. However, the student can choose the language of programming for the delivered exercises.

In the practical sessions (also in the classroom), teachers guide students in applying theoretical concepts to solve the proposed exercises, always using critical reasoning. We suggest that students solve exercises in and outside the classroom, to promote contact and use the basic tools needed to solve problems.

Students, independently, need to work on the materials provided by teachers and the outcomes of the sessions of exercises/problems, to fix and assimilate the concepts.
The teachers provide the curriculum and monitoring of activities (by ATENEA).

The different modules in which the course is divided are flexible with different options depending on the student's previous knowledge and capacities. The professors and coordinators will select the most adequate exercises to ensure that students have always acquired the expected concepts.

LEARNING OBJECTIVES OF THE SUBJECT

Learn how to solve thermodynamics, heat and mass transfer and fluid dynamics problems in different programming languages using numerical and computational methods.
Learn how to open and process files from experimental data with different programming languages.
Learn how to write codes for embedded systems to measure thermodynamic properties
Learn how to solve problems with different languages
This course also allows the student to complete the knowledge in propulsion topics and applications for the different Aeronautical grades.

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	40.00
Self study	45,0	60.00

Total learning time: 75 h

CONTENTS

Introduction to programming

Description:
The basic concepts of programming are presented. These concepts are introduced for Matlab and C++. In addition, a specific approach regarding the basic concepts of object-oriented programming is given.

Related activities:
Practical exercise 1

Full-or-part-time: 7h
Theory classes: 5h
Self study : 2h

Basic computational and numerical methods for heat transfer

Description:
The numerical resolution of the 1D transient heat transfer problem is presented using the finite volume method (FVM) and finite difference method (FDM). The importance of numerical methods and its use are presented here. At the end of this part, the student should be able to program a 1D transient code in the preferred programming language (C++, Matlab or Python)

Related activities:
Exercise 2

Full-or-part-time: 9h
Theory classes: 5h
Self study : 4h

Matlab linear solvers for sparse systems of equations

Description:
Numerical methods often require solving systems of linear equations. In this part the 1D heat transfer problem from the previous module is solved using different types of Matlab linear solvers, and the computational time and required resources are compared. In addition, students can compare their code developed in the previous part to the one developed using linear solvers.

Related activities:
Exercise 3

Full-or-part-time: 4h
Theory classes: 2h
Self study : 2h

Advanced computational and numerical methods for heat transfer

Description:
This part extends the basic concepts developed in the 1D heat transfer part to be able to solve 2D and 3D heat transfer problems on solids. Apart from the extension to more dimensions, a special focus is given to the general approach for complex problems, boundary conditions, limitations, etc. The student will be given the option to either do the proposed academic problem "4 material case", or choose a specific exercise of a practical application

Related activities:
Exercise 4

Full-or-part-time: 20h
Theory classes: 5h
Self study : 15h

Solving fluid-thermal problems

Description:
This part consists of solving fluid-thermal problems at different levels. The convection diffusion equation is discretized and basic convection schemes are presented. The student will solve the "Smith-Hutton case", corresponding to a case where a velocity field is known and the evolution of a scalar field using the convection-diffusion equation is calculated.
In addition, this module has an optional part where the Navier-Stokes equations are discretized using finite volume methods and the fractional step method. The students that choose to do this optional part will be able to program a code that can solve the "Lid Driven Cavity case" from scratch, or another type of cases related with propulsion or similar.

Related activities:
Exercise 5

Full-or-part-time: 20h
Theory classes: 5h
Self study : 15h

Scripting tools for thermal data management

Description:
Data-driven activities are gathering a very high relevance in the engineering field (post-process, training of machine-learning models, etc.). In this course they will be focused on the analysis and comparison of thermal data, coming from experiments or numerical calculations.
The explanation will be based on a very simple case (numerical to experimental comparison) to highlight easily the different steps: importing data into a table format, data management to create new parameters/properties; filtering of the data to select subgroups; fitting procedures to create models of the comparison; plotting basics. Automatic reporting and presentation tools will also be presented shortly.

Related activities:
The different techniques will be presented in class by following a simple example (Python is in principle expected as the dominant data-driven programming language; Matlab examples also available) of a experimental-vs-numerical comparison scenario. The students will have some time after the professor explanation to run the example scripts given, and to mature/consolidate the concepts.
An exercise will be proposed to practice the techniques. It will include raw data related to the Module 7 thermodynamic exercise. Importing, treatment, filtering, plotting and fitting techniques will be practiced to finalize with consolidated skills and a clear understanding of the data provided.

Full-or-part-time: 8h
Theory classes: 5h
Self study : 3h

Scripting programming applied to Thermodynamic cycles

Description:
The thermodynamic courses given at the Undergraduate level are basic and typically use property tables, while the students do the calculations by hand. This module intends to shift the level of this analysis by calculating the properties using software databases, and then allowing a massive calculation of thermal cycles by scripting (using the programming tools and data management skills already seen in previous modules).

Related activities:
The first step in this module is to learn how to integrate the calculation of thermodynamic and thermophysical properties of fluids (using an open-source database) into our programs/scripts (Python is in principle expected as the native interface of the database; Matlab examples also available).
The second step is to review some Thermodynamics basic principles, to allow the student to translate the gathered skills in programming and properties calculation into automatic/massive calculation of thermal cycles.
A thermodynamic exercise will be required in this sense, to calculate and compare the main indicators of the suggested cycles in different conditions (evaporator and condenser heat rates, compressor work, COP, etc.), draw the p-h/T-s diagrams, etc. The pre-processing of the experimental/numerical data, that will be the input for these calculations, will be prepared during the Module 6 by the students.

Full-or-part-time: 7h
Theory classes: 3h
Self study : 4h

GRADING SYSTEM

The final grade depends on the following assessment criteria:
- Practical exercises 1, 2 and 3: Different exercises within programming and thermal and fluid dynamics phenomenological problems (1D heat conduction, linear solvers, OOP programming). 40%
- Practical exercises 4 and 5: The students will choose a specific exercise within thermal and fluid dynamics applied problems for industrial and/or aeronautical engineering. 35%
- Practical exercise 6: The students will choose a specific exercise for pre-processing, post-processing and data management issues. 25%

BIBLIOGRAPHY

Basic:

- Palm, William J. Introduction to MATLAB for engineers. 3rd ed. Dubuque, IA: McGraw-Hill, 2012. ISBN 9781259012051.
- Patankar, Suhas V. Numerical heat transfer and fluid flow [on line]. Boca Ratón: Washington: Taylor & Francis; Hemisphere Pub. Co, 1980 [Consultation: 25/09/2024]. Available on: https://www-taylorfrancis-com.recursos.biblioteca.upc.edu/books/mono/10.1201/9781482234213/numerical-heat-transfer-fluid-flow-suhas-patankar. ISBN 0891165223.
- Stroustrup, Bjarne. The C++ programming language [on line]. 4th ed. Upper Saddle River, NJ: Addison-Wesley, cop. 2013 [Consultation: 25/09/2024]. Available on: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=7115378. ISBN 9780321563842.
- Guttag, John. Introduction to computation and programming using Python: with application to understanding data [on line]. 3rd edition. Cambridge, Massachusetts: The MIT Press, 2021 [Consultation: 25/09/2024]. Available on: https://search-ebscohost-com.recursos.biblioteca.upc.edu/login.aspx?direct=true&AuthType=ip,uid&db=nlebk&AN=2518027&site=ehost-live&ebv=EK&ppid=Page- -1. ISBN 9780262542364.