Received: 6 September 2022 | Revised: 23 January 2023 | Accepted: 11 February 2023

DOI: 10.1002/cae.22619

RESEARCH ARTICLE

WILEY

Jupyter Notebooks for the study of advanced topics in Fluid Mechanics

Robert Castilla[1] | **Marta Peña**[2]

[1]Department of Fluid Mechanics—CATMech, Universitat Politècnica de Catalunya, Terrassa, Spain

[2]Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

Correspondence
Marta Peña, Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain.
Email: marta.penya@upc.edu

Abstract

In recent years, Jupyter Notebooks have become a very useful free and open-source tool in teaching, as they allow you to combine text, images, mathematical expressions, links and code into a single document. This gives students an interactive document with which they can experiment and learn with the help of high-level mathematical calculus. In Fluid Mechanics, it is very common for students to deal with complex computations that take away attention from the Mechanic, especially in advanced topics such as Rheology, Turbulence, or Boundary Layer. The subject "Advanced Fluid Mechanics" is an elective one of the last year of the Bachelor's degree in Industrial and Aerospace Technology Engineering at the Terrassa School of Industrial, Aerospace and Audiovisual Engineering at the Universitat Politècnica de Catalunya. This subject has three ECTS credits and has been taught since the academic year 2020–2021 This subject complements the compulsory subject Fluid Mechanics and is developed in 6 weeks with 5 h of class each week. This work presents Fluid Mechanics modules with Jupyter Notebooks that complement the syllabus given in the compulsory subject. An elective subject is presented where subjects of Fluid Mechanics per week are studied independently, using different Python tools: symbolic calculation, modeling of experimental data, statistical analysis, numerical calculation, and so forth. The main goal is for the student to focus on mechanical concepts and actively learn to use the tools available, especially open source, to do the associated mathematical calculations.

KEYWORDS

active learning, collaborative work, fluid mechanics, interactive computing, *Jupyter Notebooks*, open source

Abbreviations: AFM, Advanced Fluid Mechanics; CFD, computational fluid dynamics; FM, Fluid Mechanics; ODE, ordinary differential equations; PDE, partial derivatives equation.

Robert Castilla and Marta Peña contributed equally to this study.

1 | INTRODUCTION

Fluid Mechanics (FM) involves mathematical tools as diverse as Differential Equations in both Ordinary and Partial Derivatives, Geometry, Vector and Tensorial Analysis, Integrals, Complex Variables, Statistics, Numerical Calculus, and the list goes on. This means that students sometimes lose sight of the mechanical concepts in the middle of a web of mathematical equations, or some students interpret the concept of solving FM problems as looking for the correct mathematical formula that fits the given data [8, 23]. This often leads to resolutions without any physical meaning.

Today, students are always using their computers, smartphones, and tablets. The high rate of device penetration globally poses new challenges and universities can continue this trend. The educational field is developing new challenges. Online education, distance learning, and other educational computing scenarios are increasingly being used. In recent years, different types of methods have been developed (artificial intelligence, mobile applications, virtual reality, game development...) that use electronic resources to improve students' understanding. These methods are known as e-learning. Several of these methods have been applied in higher education, especially in universities where education is continuously updated with new tools. The number of mobile applications used in universities is increasing, as it is shown in Refs. [15], [17], and [34]. Using new technologies in the teaching process provides a lot of benefits including saving time, having faster and more accurate evaluation tools, creating new learning environments full of visual aids such as animated media, high-quality pictures, three-dimensional (3D) models, and interactive way of learning to attract students to learn and provide better outcomes.

Teaching and learning strategies based on multiple technologies can be integrated into course content to provide the most effective learning and teaching. As computers have moved from laboratories to classrooms and homes, much research has investigated their use for educational purposes. Since computer software can be a potentially powerful tool, it is important to find out which techniques are best and subsequently employ these techniques to enhance the learning experience. In particular, computers can be used for a variety of multimedia techniques, and it is important to characterize the circumstances of their successful use. The use of electronic media has been widely recognized as an effective and efficient tool in delivering course materials, as it is shown in Refs. [16] and [18]. Considering Engineering as an applied science, simulations can be used to complement education methods, specifically in undergraduate engineering degrees, because it allows the students to work with real and complex situations in a controlled and risk-free environment. Interactive computer simulations yield dynamic and visual representations of physical phenomena and allow students to interact with a simplified version of a scientific process or system which is built upon a mathematical model. For example, Radinschi et al. [40], Botelho et al. [7], and Mateo et al. [30].

On the other hand, open and free software has gained popularity in recent years, especially for research and educational purposes. In particular, the high-level language Python has increased its acceptance due to its simplicity, large user community, and availability of powerful libraries and tools [25, 26, 33]. Jupyter [32, 36] and JupyterLab are the evolution of the IPython interactive shell [44], which is not only aimed at Python but also at other languages, such as R or Julia. There is a huge collection of Jupyter Notebooks dedicated to Data Science, mainly with the pandas library [31, 37, 47], and it is extensively used for Computing Sciences, Engineering, Science, Signal Processing,... A large collection of these notable Notebooks are available on Jupyter's Github site [27]. However, with respect to FM, there are Jupyter Notebooks about Computational Fluid Mechanics [3] and Aerodynamics [4], and also other innovative methodologies, such Project Based Learning [35] or Flipped Classroom [1, 20, 23] but, to the authors' knowledge, there is no Jupyter Notebook focused to the study of topics related to FM in general.

1.1 | Context of the study

The subject Advanced Fluid Mechanics (AFM) is an elective subject in the last year of the Bachelor's Degree in Industrial and Aerospace Engineering at the Terrassa School of Industrial, Aerospace and Audiovisual Engineering (ESEIAAT) [Universitat Politècnica de Catalunya (UPC)]. This subject has three European Credit Transfer System (ECTS) credits and has been taught since the 2020–2021 academic year. It is an elective subject that serves to complement the compulsory subject FM of 4.5 ECTS credits. The subject AFM is developed in 6 weeks with 5 h of class each week and is taught in English. The number of places offered is 20 students.

In the final year of the Bachelor's Degree, or in the Master's Degree, students must be given tools more similar to those found in the world of work. On the other hand, it is interesting to introduce students to the world of free software (Linux OS, Apache, WordPress, Mozilla, GIMP...). The international community dedicated to free software in general, and education in particular, has not stopped growing and promoting activities, as it is shown in Ref. [14].

This work presents the material developed in the subject AFM, five topics of FM that are usually explained superficially in the compulsory subject. In this subject, they are presented in more detail and with a different methodology that allows students to make autonomous learning material. Being a subject in the last year of the Bachelor's Degree, it is hoped that the students have the appropriate knowledge of FM that is taught in the compulsory subjects. In addition, they also generally have a basic knowledge of programming, including high-level languages (usually Matlab, but sometimes Python or R as well).

1.2 | Objectives and research questions

The main objective of this work is as follows:

- To provide students with an alternative study methodology, to complement the syllabus of the compulsory subject of FM, making the learning of physical concepts active instead of passive and not conditioned by the limitations of the knowledge of mathematics.

The complementary objectives that are pursued, to achieve the main objective, are as follows:

- To familiarize students with the Jupyter Notebook study, writing, documentation, and math calculation tool.
- To generate independent modules with study material with advanced topics in FM.

The research questions posed in this work are the following:

- Does the use of Jupyter Notebooks compared to the classic content presentation offer better academic results?
- Are students satisfied with the use of these new tools?

1.3 | State of the art

Nowadays, a new method used for e-learning is the knowledge pills or virtual laboratories. The pills are widely used for theoretical explanations. Knowledge pills are a tool that has been used recently. This type of resource is useful in the e-learning processes, exclusively for distance teaching or to support classroom teaching.

The main source of inspiration for this proposal is the work of Professor Lorena Barba [5, 3]. Barba and her group, from George Washington University (Washington), have for years made a strong commitment to the ecosystem represented by Jupyter Notebooks to teach, especially at distance, the methodology behind Computational Fluid Dynamics (CFD). An example of their efforts is the MOOC (Massive Online Open Course) they have published since 2017 [2]. Cardoso et al. [9] also report on the use of Jupyter Notebooks in engineering courses at the University of Coimbra, assessing that students have shown an improvement in the learning process.

More recently, Golman [21] proposes a set of Jupyter Notebooks for the numerical analysis of chemical and enzyme-catalyzed reactions in a porous pellet. Suárez-García et al. [45] present an experience of using Jupyter Notebooks as teaching material in the field of Structural Analysis. Specifically, they report the results of using Jupyter Notebooks in two lab sessions and at the end conduct a survey with 28 students about the experience. In general, they conclude that the acceptance has been positive and the Jupyter Notebooks have served for a better understanding of the theory by the students.

But if there is one publication that has had a major impact it has been that of Helen Shen in Nature [44] which has accumulated more than 200 citations in Google Scholar. Shen, however, still talks about the Ipython Notebook instead of the Jupyter Notebook, which was created precisely in 2014 from Ipython. In this article, Shen describes the benefits of creating, sharing, disseminating, and publishing documents, code, and data using Notebooks.

2 | METHODOLOGY

2.1 | Description of the Jupyter Notebooks

A Jupyter Notebook is a text file, which is saved with the .ipynb extension, and has the structure of a JSON [38] and can therefore be easily read and exchanged between users. A Jupyter Notebook file can be opened with a multitude of programs on almost all the available platforms. Basically, a Jupyter Notebook combines text, mathematical formulas, links, figures, videos, and, most importantly, executable code. The text is formatted with Markdown [22], which is a very simple way to write text with the most basic format (titles and sections, bold, italics, list, numbering, code, links, images, etc.).

The main idea behind a Jupyter Notebook is to create a computational narrative, if possible, following the 10 rules that Rule et al. [43] suggested, and which are summarized here:

1. *Tell a story to an audience.*
2. *Document the process, not just the results.*
3. *Use cell divisions to clear the steps.*
4. *Modularize the code.*
5. *Register the dependencies.*
6. *Use version control.*
7. *Build a pipe.*
8. *Share and explain your data.*
9. *Design your Notebooks to read, run, and explore.*
10. *Advocates open research.*

Jupyter Notebook is becoming a standard in many fields of Science, Technology, and, mainly, Data Science, as indicated by the "Software System Award" that the Association of Computing Machine (ACM) awarded to Fernando Pérez, creator of Jupyter Notebook, in 2017. The public repositories of GitHub hosted 2018 almost three million Notebooks.

The set of Jupyter Notebooks presented here are designed to be independent of each other, except for the first one, which serves as an introduction to the tools that are being to be used throughout the whole project. In this way, the order of the Notebooks can be changed in the classroom, some books can be skipped, or, more importantly, other Notebooks can be added.

At the present moment, the project is composed of six Jupyter Notebooks. The first one, numbered with 0 following the use with python indexed items, is an introductory module with more basic concepts about python, installation, and load of modules, and a description of the main modules used in the project: numpy, matplotlib, scipy, pandas, sympy, and statsmodels. The next Jupyter Notebooks explore exciting topics in FM as lubrication, capillarity, rheology, boundary layer, turbulence, … by using statistical tools, symbolic computation, numerical methods, function definitions, etc. The Notebooks are intended to be self-explanatory and closed. The phylosophy of making a narrative, such as telling history, has been considered in all the Notebooks.

Each Notebook has an assignment associated. This assignment can be solved individually or in groups of maximum three students, and it can be done during class hours if the students have time or as homework. The solution has to be delivered in a maximum period of 1 week.

In the next sections, the Notebooks are briefly described. The details of each Notebook can be found in the online version of the course [10].

3 | RESULTS

3.1 | Jupyter Notebooks for AFM

3.1.1 | Introduction to Python, Jupyter Notebook and main modules

In this Notebook, the fundamentals of python programming are introduced at the very beginning. After that, students learn how to install and load the main modules that are going to be used during the course. Then in several sections, each module is explained following some simple examples. Following the book by Johansson [26] sympy is used to solve the 1-D first-degree ordinary differential equation (ODE) for Newton cooling law, and the second-degree ODE for the damped harmonic oscillator. Also, more related to FM, the simple equation of motion for a free rocket is introduced.

In the next section, the pandas module for data analysis is introduced with a very easy example of subject grades and teachers. Finally, statsmodels and scipy are used to look for a model by the fitting of a scatter of two-dimensional (2-D) points into a curve.

The assignment for this Notebook is to find the emptying time of a conical vessel. It is a simple problem that is explained in the fundamental courses of FM, but with some physical simplification. Here, students are asked to solve the problem with sympy and numerical integration overcoming the mentioned simplification.

3.1.2 | Microflows. Lubrication and capillarity

Reynolds number is in this case defined as

$$\mathrm{Re} = \frac{UD}{\nu}. \tag{1}$$

When it is very small, below 1, because the scale of the flow is of the scale of microns the Navier-Stokes equations are linearized (this is known as Stokes equations) and usually can be solved analytically. The Reynolds equation [41, 48] can be derived for the case of lubrication between two close bodies, with a narrow gap, not necessarily uniform, and with a relative movement between them. Figure 1 depicts this case of lubrication between bodies.

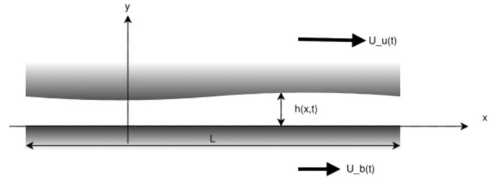

FIGURE 1 Relative movement between close bodies with relative motion, to illustrate Reynolds equation for lubrication.

Reynolds equation is derived with sympy. Students can see how to use symbolic computation to get useful ODE than can be integrated. The code to obtain the solution of Stokes equation follows

```
import sympy as sp
x,y,t,mu = sp.symbols(x,y,t,mu)
u = sp.Function(u)
p = sp.Function(p)
h = sp.Function(h)
p_ = sp.Derivative(p(x),x)

exp = p_/mu+sp.diff(u(y),y,2)

Ub,Ut = sp.symbols(U_b,U_t)
ics = {u(0):Ub,u(h(x)):Ut}

Sol = sp.dsolve(exp,u(y),ics=ics)
```

that yields

$$u(y) = -\frac{U_b y}{h(x)} + U_b + \frac{U_t y}{h(x)} + \frac{y^2 \frac{d}{dx}p(x)}{2\mu} - \frac{yh(x)\frac{d}{dx}p(x)}{2\mu}, \tag{2}$$

and, with the particular case of $U_b = 0$ and $U_t = U$,

$$u(y) = \frac{Uy}{h(x)} + \frac{y^2 \frac{d}{dx}p(x)}{2\mu} - \frac{yh(x)\frac{d}{dx}p(x)}{2\mu}. \tag{3}$$

When it is imposed that the integral of this profile cannot be function of x (continuity condition), that is easily done with `ReynoldsEq = sp.Eq(intU.diff(x),0)`, Reynolds equation is obtained

$$\frac{U\frac{d}{dx}h(x)}{2} - \frac{h^3(x)\frac{d^2}{dx^2}p(x)}{12\mu} - \frac{h^2(x)\frac{d}{dx}h(x)\frac{d}{dx}p(x)}{4\mu} = 0, \tag{4}$$

and pressure distribution is computed by integrating the velocity profile and isolating the pressure gradient.

```
intU = sp.integrate(Sol.rhs, (y,0,
h(x)))
q = sp.symbols(q)
exp_dpdx = sp.solve(intUq,p(x).diff
(x))
p_eq = sp.Eq(p_, exp_dpdx[0])
```

$$\frac{d}{dx}p(x) = \frac{6\mu(Uh(x) - 2q)}{h^3(x)}. \tag{5}$$

Finally, it is integrated

```
sp.dsolve(p_eq,p(x))
```

$$p(x) = C_1 + 6\mu\left(U\int\frac{1}{h^2(x)}dx - 2q\int\frac{1}{h^3(x)}dx\right), \tag{6}$$

where the constant C_1 will depend on the known value of the pressure on some point x.

As an example, in the Notebook the case of a tilting-pad bearing, where $h(x) = h_0\left(1 + \alpha\frac{x}{L}\right)$, with $\alpha \ll 1$, is solved. The case of a journal bearing (see Figure 2), where a load **W** displaces the shaft and produces an eccentricity e is left as the assignment for this Notebook. In this case, the distance between the surface is given by

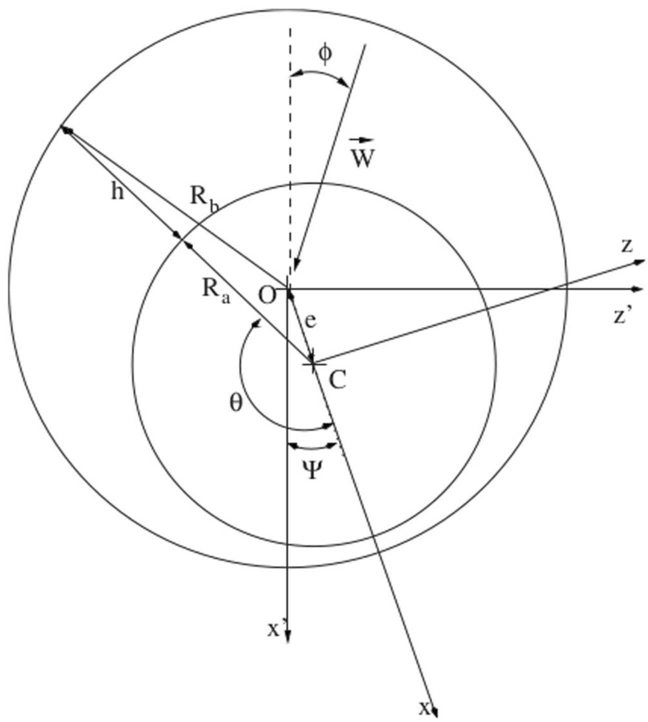

FIGURE 2 Sketch of a hydrodynamics journal bearing. Due to the load **W**, the rotating shaft, of radius R_a is not concentric to the housing, and, hence, the gap is a function of the angle θ with respect to the minimum value. This produces a pressure distribution, according to the Reynolds equation, that cancels the load.

with $h(x) = h_0(1 + \varepsilon \cos(\theta))$ where $\varepsilon = \frac{e}{h_0}$ and $h_0 = R_b - R_a$, where θ is the angular position with respect to the minimum distance, h_0, and R_a and R_b are, respectively, the shaft and the housing radia.

In this notebook, the equation of Young–Laplace for interface liquid-gas with surface tension is also presented, and it is used to find the shape of this interface.

The main objective of this Notebook is the solution of differential equations with the help of `sympy`.

3.1.3 | Rheology

Rheology is an interesting branch of FM that deserves more attention than it is usually given. It deals with the study of the deformation of materials, both solids, and fluids [24, 49]. In the present case, the Notebook focuses on fluids, and it analyzes simple models as the Ostwald-de Waele law and its analytical computation of velocity profile in a pipe, and also more complex behaviors, such as a viscoelastic unsteady flow that has to be solved with finite differences code for the case of the Maxwell model. Figure 3 shows a result of this computation, where the dimensionless velocity $u^* = \frac{4\eta}{R^2 p'} u$ is plotted against the dimensionless time $t^* = \frac{\eta}{\rho R^2} t$, where η and ρ are, respectively, the viscosity and the density of the fluid, R is the pipe radius and p' is the gradient of the pressure in the pipe, with a dimensionless parameter $\xi = \frac{\eta}{\rho R^2} \lambda = 0.2$, where λ is the relaxation time of the Maxwell model. It is suggested to students to modify the code to inspect the behavior with a more complex model, like the Voight-Kelvin one.

As an assignment for this notebook, it is proposed to use the module `scipy.curve_fit` to fit a set of experimental data, taken at the Centre for Industrial Rheology [11], that are shown in Figure 4. To get the data, and convert the points in the graphic into numbers in a file, a plot digitizer is used [42]. It is clear that the data do not correctly fit the Ostwald-de Waele law, and it is suggested to use the Carreau model, given the link to the Wikipedia page.

The main objectives of this notebook are the use of `curve_fit` module of `scipy`, that is, really useful for modelization and, also, the use of the Finite Difference Method or the solution of nonlinear partial differential equations (PDE).

3.1.4 | Turbulence

The subject of turbulence is so complex and wide that it would deserve a complete course of several tens of hours. In this Notebook, it is analyzed only from an experimental and statistical point of view, showing the main characteristics scales of a one-component of the velocity acquired in a wind tunnel, in the wake behind an airfoil. The measurement of the velocity was done with a Dantec Constant Temperature Anemometer with a hot-wire probe, and data are provided to students with a CSV file. The CSV data are composed of 17 columns, one for each y position of the probe in the normal direction to the wake and the airfoil plane, and 1000 rows, for the measurement for 1 s of the x-velocity with a frequency of 1000 Hz.

Using the `pandas` library, very popular in Data Engineering, students compute important turbulence characteristics like

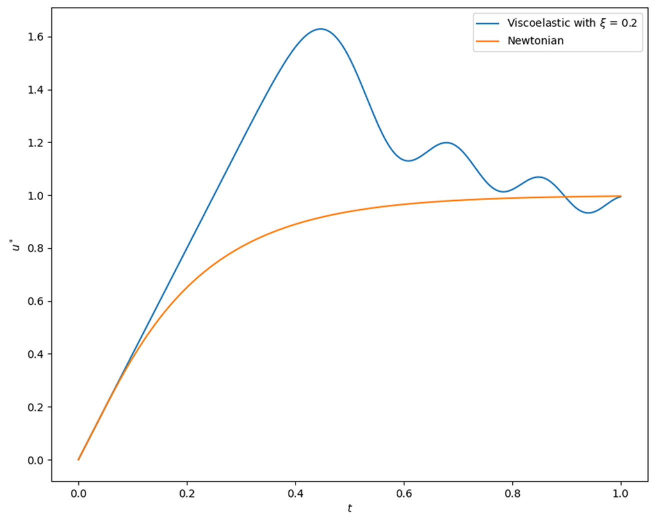

FIGURE 3 Time evolution of flow in a pipe of a viscoelastic fluid, in comparation with a Newtonian fluid. It is obtained with a small finite difference code. Velocity, time and relaxation time are dimensionless.

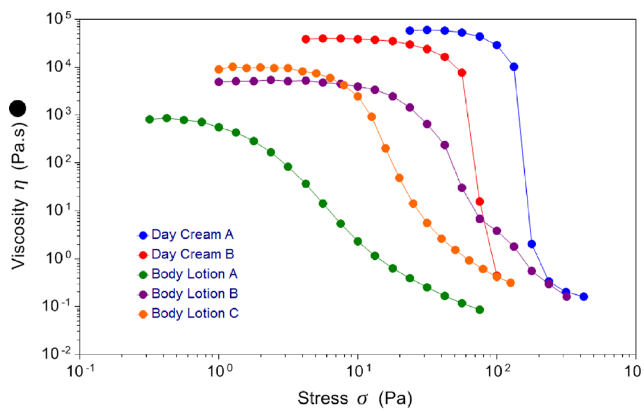

FIGURE 4 Rheological behavior of a group of creams and lotions [11].

- Turbulence intensity
- Turbulence Reynolds number
- Kolmogorov and integral time scales
- Energy dissipation rate
- Velocity auto-correlation
- Probability Density Function
- Energy Spectrum

For the assignment, quite simply, students are encouraged to find the equivalent space magnitudes by using Taylor's hypothesis [39].

3.1.5 | Boundary layer

Boundary Layer is a mandatory topic in the majority of fundamental FM textbooks, and it is also included in the syllabus of the compulsory subject of the Bachelor's Degress in our School. Nevertheless, the study of the boundary layer, even for the simplest case 2D laminar regime in a planar plate without pressure gradient, requires the solution of a nonlinear system of two PDEs. The classical Blasius theory [28, 48] uses algebraic manipulation with dimensionless magnitudes to reduce it to a third-order ODE,

$$2\frac{df}{d\eta^3} + f\frac{d^2f}{d\eta^2} = 0, \qquad (7)$$

where η stands for a dimensionless distance to the wall, and f is the dimensionless primitive of x-velocity. The vertical component of velocity is related also to f' with the continuity equation. This equation has no analytical solution, and Blasius solved it by approximation of power series [6]. In this notebook, this equation is solved as a boundary value problem, where the boundary conditions are the following:

$$f = \frac{df}{d\eta} = 0 \text{for} \eta = 0, \qquad (8)$$

$$\frac{df}{d\eta} \rightarrow 1 \text{for} \eta \rightarrow \infty. \qquad (9)$$

This ODE is numerically solved with the `solve_bvp` method of the `scipy.ode` package, and the solution is compared with the parabolic approximation. Finally, the von Kármán momentum equation is introduced and studied, as well as the turbulent boundary layer theory, along with the well-known log law.

As an assignment for this notebook, it is proposed to find a power law approximation by fitting the points obtained with the log law and use the von Kármań equation to find the growth law for a turbulent boundary layer.

3.1.6 | Compressible flow

The theory of one-dimensional compressible flow, even without friction, requires the solution of complex algebraical nonlinear equations. In this Notebook, a set of functions to relate the thermodynamical, physical, and geometrical variables are coded, following rule 4 described in Section 2.1. Some of them are left to students for practicing. Also, as a very important point, the proper documentation of the functions is required, to help a future user or, more frequently, the developer himself or herself.

The topics covered in this Notebook are

- Isentropic flow
- Normal shock wave
- Oblique shock wave
- Prandtl-Meyer expansion fan

As an example, Figure 5 shows the ratios of variables across a normal shock wave, as computed by the functions developed in the Notebook.

As an assignment for this last Notebook, it is suggested to write a function, properly documented, to compute the deflection angle in an oblique shock wave.

3.2 | Students' assessment and surveys results

Two indicators have been considered to evaluate the proposal. The first is the academic performance of the students, compared to the performance of the students themselves in the previous compulsory subjects. The second indicator is the results of the surveys, both the

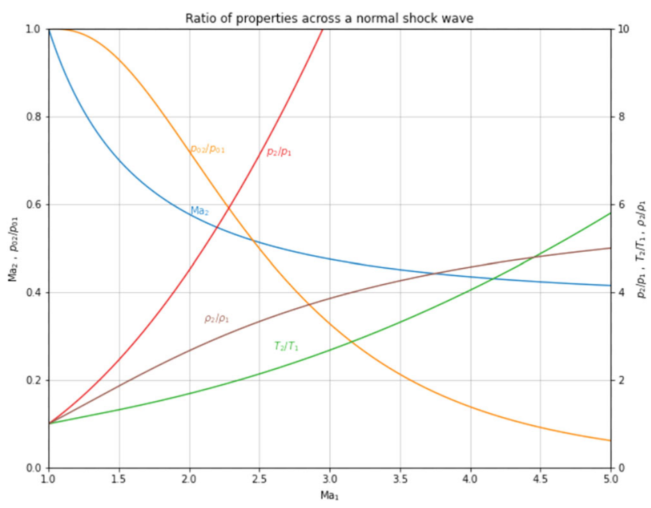

FIGURE 5 Curves of variables ratios in a normal shock wave.

officials of the [UPC] and the SEEQ survey used by the authors. The sampling of the marks was carried out on 18 students from the 2 academic years analyzed, 2020–21 and 2021–22. Students who are not noticed in any of the two subjects (because they have not been presented) and foreign (Erasmus) students have been eliminated. That is, only students who have completed both subjects in the same Bachelor's Degree have been considered. The first 12 students were from the 2020–21 academic year and the last 6 from 2021 to 22 academic year. The small number of students last year is due to the fact that more than half were foreigners. The results are shown in Figure 6.

Regarding the students' opinions, two different surveys have been considered. First, there is the official [University] survey, which consists of three main questions:

1. I found the contents of the subject interesting.
2. Overall I am satisfied with this subject.
3. The course materials are easily accessible and useful for learning.

The results of these three questions are compared to three compulsory subjects, [Subject 1], [Subject 2], and [Subject 3] of the Bachelor's Degree in the area of FM, averaging the three semesters in which the surveys were conducted, with the result obtained in the subject AFM in the only semester in which survey results are available (the survey was also conducted this academic year 2021–2022, but unfortunately the results depend on the [University] and are not be available yet). See Table 1.

Secondly, a SEEQ (Student Evaluation of Educational Quality) [19, 29] survey has been carried out in two editions of the subject presented in this report, and the compulsory subject FM of the Bachelor's Degree of Industrial Technologies. It should be noted that the lecturer was the same in both subjects; Not all the SEEQ questions, which are more than 40, are presented here, but only the five that have been considered most relevant in relation to the subject and the methodology. The questions and the results are presented in Figure 7. In both surveys, the student's answers are given on a scale of 5 values: 1: Strongly disagree, 2: Disagree, 3: Neutral, 4: Agree, and 5: Strongly agree.

4 | DISCUSSION AND CONCLUSIONS

4.1 | Jupyter Notebooks discussion

The following are the quality principles enunciated by Chickering and Gamson [13] for a teaching innovation action. The relationship with the present proposal is commented out with each of the principles.

1. *Stimulate contact between teachers and students.* Although it is not the strong point of the proposal, as the material could be used without the intervention of the teacher, it is tried to enhance this stimulation through more or less simple exercises that often require communication between teacher and student.
2. *Stimulate cooperation between students.* The assignments that are delivered and evaluated are carried out in groups of three. It is tried in the classroom, in addition, to avoiding the competitiveness between students and groups. The goal is to learn as much as possible, not to get the highest grade. In this way, collaboration between groups is encouraged, in

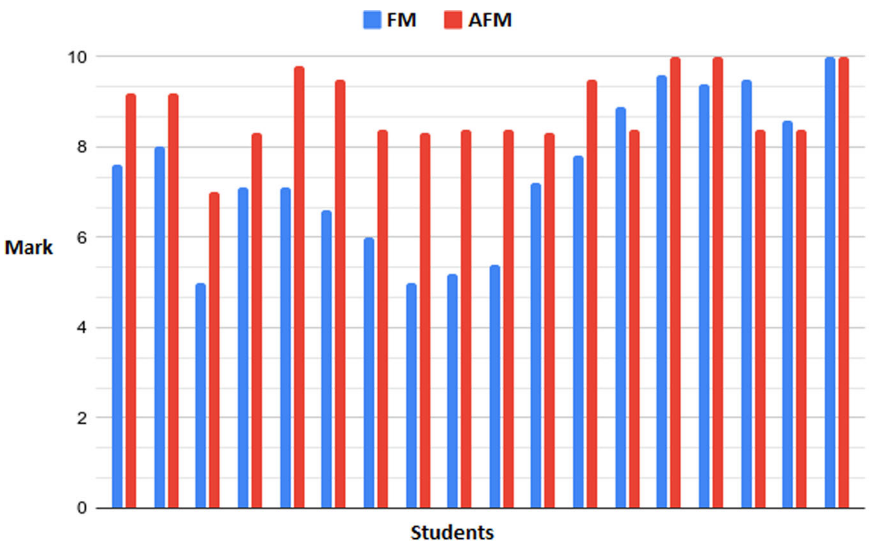

FIGURE 6 Marks from the 18 students in the compulsory subject of Fluid Mechanics (FM) and the elective subject of Advanced Fluid Mechanics (AFM).

TABLE 1 Results of the Universitat Politècnica de Catalunya surveys with the three compulsory subjects of Fluid Mechanics and the elective subject of Advanced Fluid Mechanics (AFM).

Subject	Year	Question 1	Question 2	Question 3	Sample
Subject 1	2021/22-1	3.75	3.63	3.63	8/25
	2020/21-2	4.33	3.95	4.22	55/113
	2019/20-1	4.06	3.44	4.06	16/29
Subject 2	2021/22-1	4.24	3.57	2.95	21/64
	2020/21-2	3.63	2.46	2	54/66
	2019/20-1	3.5	2.79	2.79	24/62
Subject 3	2021/22-1	3.82	4.29	4.53	17/55
	2020/21-2	4.15	4.25	4.45	20/55
	2019/20-1	4.03	3.71	3.68	31/75
Average		3.95	3.57	3.59	
AFM		4.8	4.8	4.8	5/15

classroom and in the digital classroom, through forums where all deliveries are shared.

3. *Stimulate active learning*. This is the most important point of this proposal. Students take almost no notes but are encouraged to explore, play, and change notebooks. They are encouraged to change the input and interpret the results. They are asked to look for alternative ways to solve the same problem. Throughout all Notebooks, many of the tools used are unexplained, and links to Wikipedia, GitHub repositories, and so forth are provided for information.
4. *Provide timely feedback*. Each week a topic that is independent of the rest is worked on. At the end of the week, they have an "Assignment" that they have to hand in the following week. Once delivered, the Notebooks are posted on a Moodle forum with access for all students. In this way, students can analyze each of the deliveries, and give a comment and even a mark. Feedback is often given to each other and, more importantly, they see different ways of solving the same problem. Obviously, the teacher puts the final mark on the delivery, but it does not go far from what they put.
5. *Spend time on the most relevant tasks*. As explained in Section 1, one of the motivations for this proposal is that students often do not learn FM because they are stuck with mathematical concepts and calculations. The aim of this proposal is for students to focus on the concepts of FM by learning to do mathematical calculations, and visualize them, with appropriate tools.
6. *Communicate high expectations to students*. Students end up creating their own Notebooks. At first, they seem skeptical, but after the first 2 weeks they see that it is not that complicated and they end up doing really interesting work.
7. *Respect the different talents and forms of learning*. In fact, the course could be done completely autonomously. It is designed so that students can meet small challenges gradually and at their own pace. In the classroom, it is common for students to finish work very quickly and either play with Notebooks, start working on assignments, or collaborate with other groups.

The fact that all the material is in a repository in a well-documented way makes it easy for any teacher to take charge of the subject at any time, just by making a fork of the repository or working in a branch which can then be combined with the main trunk (merge). The Jupyter Notebooks presented in this work, by definition, are fully documented and accessible to anyone. Not only can the course be transferred to any department, school, or university professor but, being developed in English and available in a public repository, it can be used by anyone in the world. To use and contribute to the proposal, it is advisable to know and use Git and GitHub, reading previously any of the books by Chacon and Atraub [12] or by Tsitoara [46].

This repository is constantly being updated with large contributions from students. More modules can be generated with topics that may be of interest to certain students, such as:

- Aerodynamics (experimental data processing - error propagation, reliability, modeling...).
- Pipe flow (development of a tool for calculating pressure losses in transport pipes).
- Potential flow (calculation with complex variable).

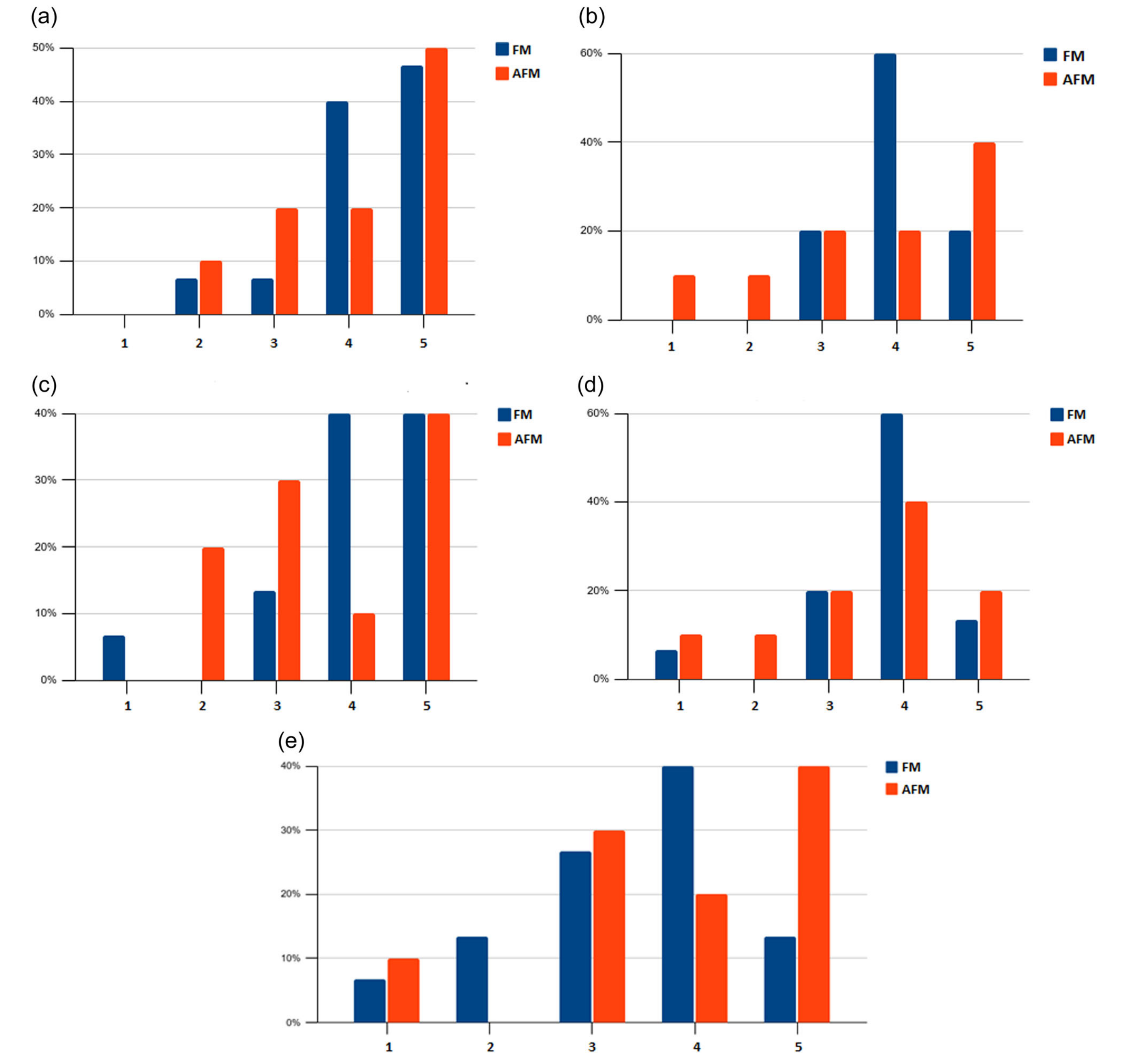

FIGURE 7 Results of the five most relevant questions of the SEEQ survey of the compulsory subject of Fluid Mechanics (FM) and the elective subject of Advanced Fluid Mechanics (AFM) (a) Results of question 1: The course has seemed to me intellectually encouraging and stimulating, (b) Results of question 2: I have learned things I consider valuable, (c) Results of question 3: My interest in the subject has increased as a result of this course, (d) Results of question 4: I have learned and understood the contents of this course, (e) Results of question 5: This course is better than most I have done at this university.

- CFD (with Fenics and Dolfin, or writing directly the code).
- etc.

Of course, the methodology described in this work can be extended to other subjects. It is very common to be used in computer science, but it can also be used in other branches of Engineering, or in Physics, Mathematics, Chemistry, and so forth.

Jupyter Notebooks is becoming a standard for the research and teaching community in recent years. It allows for combining text, images, mathematical expressions, and codes in one document that can be opened and executed in several applications and platforms. However, except for the particular case of CFD, its use for FM does not seem to be extended. This work proposes a FM course structured with Jupyter Notebooks that are presented as independent modules that encourage active and collaborative learning. The main objective is to introduce students to work on scientific and technological tools with Python, such as symbolic calculation, data analysis, ODE and PDE solution, and numerical computation, using it with advanced topics on FM, such as lubrication, rheology, compressible flow, …

The Jupyter Notebooks are distributed in a public repository in Github so that not only the students but also any academic member can contribute and extend the course.

TABLE 2 Average of assessments of each subject according to the five questions of the SEEQ survey.

	Fluid Mechanics	Advanced Fluid Mechanics
Question 1	3.3	3.1
Question 2	3.0	2.7
Question 3	3.2	2.7
Question 4	2.7	2.5
Question 5	2.4	2.8

4.2 | Students' assessment and surveys discussion

There is, in general, a significant increase in marks, as can be seen in Figure 6. In part, this increase can be attributed to the fact that it is an elective subject of a higher course, with fewer students, and therefore easier to obtain better performance. However, the fact that there are students who have dropped the mark, especially in the last year, suggests that, on average, the increase in the mark is largely due to the change in methodology.

On the one hand, there is a significant increase in the assessment by students, as can be seen in Table 1, from an average assessment of less than 4 out of 5 to 4.8 in all three questions. Nevertheless, it should be considered that the sample is much lower in the case of AFM than in the other three compulsory subjects, and therefore the result is much less reliable. A larger sample should be available in the coming years to draw more definitive conclusions.

Table 2 shows that the average score is slightly lower in the case of the subject proposed in this work in the first four questions, but higher than the last ("This course is better than most I have done at this university") where AFM is a little better valued.

From Figure 7a–e it is obvious how the opinion between "Agree" and "Strongly agree" is exchanged between the subjects FM and AFM. While the "Agree" option (score 4) is generally more common in FM, the situation changes with "Strongly agree", which is more common in AFM. However, this frequency reversal is not high enough to exceed the average.

The results of the official surveys of the [University] reflect an increase in the assessment of the subject. However, this impression could be false due to the sample difference between the two options. It would be useful to wait another academic year or two for more samples in the case of AFM.

From the results of the SEEQ survey, it can be concluded that the new methodology of using Jupyter Notebooks does not displease the students, although it does not excite them either. The opinion of the students is quite unbalanced: either they like it a lot or they do not like it at all. The opinion is different in the compulsory subject of FM, where most students give a medium-high rating.

Finally, it may be useful to reflect here the comments left on the SEEQ questionnaire by three students:

"The course and the dynamic methodology make a subject such as Advanced Fluid Mechanics more enjoyable and much more affordable to learn. Added to that, the course has concepts and works as an introduction to python which is also a big plus."

"Very good class, loved discovering the approach of notebooks in code. Also, the subjects were interesting and the class was dynamic, and the workload was heavy but necessary to understand all the fluid mechanics concepts."

"The course materials were helpful for learning about the subject. Very well-structured code and illustrations were possible. Course works were structured so that it would be rewarding to do them with group."

AUTHOR CONTRIBUTIONS

Conceptualization: Robert Castilla; *Methodology*: Robert Castilla; *Software*, Robert Castilla; *Validation*: Robert Castilla; *Formal analysis*: Robert Castilla and Marta Peña; *Investigation*: Robert Castilla and Marta Peña; *Writing–original draft preparation*: Robert Castilla and Marta Peña; *Writing–review and editing*: Robert Castilla and Marta Peña; *Supervision*: Marta Peña. All authors have read and agreed to the published version of the manuscript.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data and material are available on request from the authors upon reasonable request. Jupyter Notebooks presented in this work are publicly available in the Github repository URL https://github.com/rclUPC/AdvancedFluidMechanics.

ETHICS STATEMENT

Not applicable.

ORCID

Robert Castilla https://orcid.org/0000-0002-3848-2004
Marta Peña https://orcid.org/0000-0003-3889-8584

REFERENCES

1. O. M. Ayala, O. Popescu, and V. M. Jovanovic, Flipped classroom as blended learning in a fluid mechanics course in engineering technology, 2017.
2. L. A. Barba, GW's MAE6286. Practical Numerical Methods with Python. https://openedx.seas.gwu.edu/courses/course-v1:MAE%2BMAE6286%2B2017, 2017.
3. L. A. Barba and G. F. Forsyth, *CFD Python: the 12 steps to Navier-Stokes equations*, J. Open Source Educ. **2** (2018), no. 16, 21.
4. L. A. Barba and O. Mesnard, *Aero Python: classical aerodynamics of potential flow using Python*, J. Open Source Educ. **2** (2019), no. 16, 45.
5. L. A. Barba, L. J. Barker, D. S. Blank, J. Brown, A. B. Downey, T. George, L. J. Heagy, K. T. Mandli, J. K. Moore, D. Lippert, K. E. Niemeyer, R. R. Watkins, R. H. West, E. Wickes, C. Willing, and M. Zingale, Teaching and learning with Jupyter. Recuperado, 2017. https://jupyter4edu.github.io/jupyter-edu-book
6. H. Blasius, The boundary layers in fluids with little friction, **56**, Zeitschrift fuer Mathematik und Physik, 1950.
7. W. T. Botelho, M. d.G. B. Marietto, J. C. d.M. Ferreira, and E. P. Pimentel, *Kolb's experiential learning theory and Belhot's learning cycle guiding the use of computer simulation in engineering education: a pedagogical proposal to shift toward an experiential pedagogy*, Comput. Appl. Eng. Educ. **24** (2016), no. 1, 79–88.
8. J. K. Burgher, D. Finkel, B. J. Van Wie, and O. Adesope, *Implementing and assessing interactive physical models in the fluid mechanics classroom*, Res. Eng. Educ. **32** (2016), no. 6, 2501–2516.
9. A. Cardoso, J. Leitão, and C. Teixeira, *Using the Jupyter Notebook as a tool to support the teaching and learning processes in engineering courses*, The Challenges of the Digital Transformation in Education, 2019, pp. 227–236. https://doi.org/10.1007/978-3-030-11935-5_22
10. R. Castilla, UPC-205237 Advanced Fluid Mechanics. https://rclupc.github.io/AdvancedFluidMechanics/README.html, 2022.
11. Centre for Industrial Rheology, Emulsion stability: strong and stable or weak and feeble. Accessed June 15 2022. https://www.rheologylab.com/articles/emulsion-stability/, 2022.
12. S. Chacon and B. Straub, Progit, Springer Nature, New York, 2014.
13. A. W. Chickering and Z. F. Gamson, *Seven principles for good practice in undergraduate education*, AAHE Bull. **3** (1987), 7.
14. P. Chmielecki, Linux myth. Open source software in information society, Econ. Soc. Dev. Book Proc., New York, 2014, p. 197.
15. E. Corral Abad, M. J. Gómez García, R. Ruiz Blázquez, C. Castejon, and J. C. García-Prada, *Effects of an android app on mechanical engineering students*, Comput. Appl. Eng. Educ. **26** (2018), no. 4, 1050–1057.
16. E. Corral Abad, M. J. Gomez Garcia, E. Diez-Jimenez, P. M. Moreno-Marcos, and C. Castejon Sisamon, *Improving the learning of engineering students with interactive teaching applications*, Comput. Appl. Eng. Educ. **29** (2021), no. 6, 1665–1674.
17. K. A. Darabkh, F. H. Alturk, and S. Z. Sweidan, *VRCDEA-TCS: 3D virtual reality cooperative drawing educational application with textual chatting system*, Comput. Appl. Eng. Educ. **26** (2018), no. 5, 1677–1698.
18. B. Deliktas, *Computer technology for enhancing teaching and learning modules of engineering mechanics*, Comput. Appl. Eng. Educ. **19** (2011), no. 3, 421–432.
19. R. M. Espada and J. A. Masa, *Impact of digital storytelling in students satisfaction, measured through SEEQ questionnaire*, E-learning and Smart Learning Environment for the Preparation of New Generation Specialists, **49**.
20. P. J. Gamez-Montero, M. Peña, and N. Olmedo-Torre, *Flipped learning and threshold concepts in the turbomachinery section of fluid engineering course*, Comput. Appl. Eng. Educ. **29** (2021), no. 4, 795–809.
21. B. Golman, *A set of Jupyter notebooks for the analysis of transport phenomena and reaction in porous catalyst pellet*, Comput. Appl. Eng. Educ. **27** (2019), 531–542.
22. J. Gruber, Introducing markdown. https://daringfireball.net/2004/03/introducing_markdown
23. R. R. Gutierrez, F. Escusa, J. A. Lyon, A. J. Magana, J. H. Cabrera, R. Pehovaz, O. Link, G. Rivillas-Ospina, G. J. Acuña, J. M. Kuroiwa, M. X. Guzman, and F. G. Latosinski, *Combining hands-on and virtual experiments for enhancing fluid mechanics teaching: a design-based research study*, Comput. Appl. Eng. Educ. **30** (2022), no. 6, 1701–1724.
24. F. Irgens, Rheology and non-newtonian fluids, 1st ed. Springer International Publishing, 2014.
25. D. Irving, K. Hertweck, L. Johnston, J. Ostblom, C. Wickham, and G. Wilson, Research software engineering with python: building software that makes research possible, Chapman and Hall/CRC, New York, 2021.
26. R. Johansson, Numerical Python, Apress, New York, 2019.
27. Jupyter Community, Jupyter wiki. Accessed June 15 2022. https://github.com/jupyter/jupyter/wiki, 2022.
28. P. K. Kundu, I. M. Cohen, and D. R. Dowling, Fluid mechanics, Academic Press, Oxford, U.K., 2012.
29. H. W. Marsh, *SEEQ: a reliable, valid, and useful instrument for collecting students' evaluations of university teaching*, Br. J. Educ. Psychol. **52** (1982), no. 1, 77–95.
30. T. J. Mateo Sanguino and J. M. Andújar Márquez, *Simulation tool for teaching and learning 3D kinematics workspaces of serial robotic arms with up to 5-DOF*, Comput. Appl. Eng. Educ. **20** (2012), no. 4, 750–761.
31. W. McKinney, Python for data analysis: data wrangling with Pandas, NumPy, and IPython, O'Reilly Media, Inc., Sebastopol, CA, USA, 2012.
32. V. F. Ochkov, A. S. Moscow, and A. I. Tikhonov, *Jupyter notebook, JupyterLab; integrated environment for STEM*

education, 2022 VI Int. Conf. Inform. Technol. Eng. Educ. (Inforino), 2022.
33. T. E. Oliphant, *Python for scientific computing*, Comput. Sci. Eng. **9** (2007), no. 3, 10–20.
34. A. Ozdemir and K. F. Balbal, *Fuzzy logic based performance analysis of educational mobile game for engineering students*, Comput. Appl. Eng. Educ. **28** (2020), no. 6, 1536–1548.
35. M. Pérez-Sánchez and P. A. López-Jiménez, *Continuous project-based learning in fluid mechanics and hydraulic engineering subjects for different degrees*, Fluids. **5** (2020), no. 2, 95.
36. J. M. Perkel, *By Jupyter, it all makes sense*, Nature. **563** (2018), no. 7729, 145–146.
37. J. M. Perkel, *Why Jupyter is data scientists' computational notebook of choice*, Nature. **563** (2018), no. 7732, 145–147.
38. F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, *Foundations of JSON schema*, Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 263–273.
39. S. B. Pope, Turbulent flows, Cambridge University Press, Cambridge, 2000.
40. I. Radinschi, V. Fratiman, V. Ciocan, and M.-M. Cazacu, *Interactive computer simulations for standing waves*, Comput. Appl. Eng. Educ. **25** (2017), no. 3, 521–529.
41. O. Reynolds, *On the theory of lubrication and its application to Mr. Beauchamp Tower's experiments, including an experimental determination of the viscosity of olive oil*, Phil. Trans. Roy. Soc. **1** (1885), 157.
42. A. Rohatgi, Webplotdigitizer: Version 4.5. https://automeris.io/WebPlotDigitizer, 2021.
43. A. Rule, A. Birmingham, C. Zuniga, I. Altintas, S.-C. Huang, R. Knight, N. Moshiri, M. H. Nguyen, S. B. Rosenthal, F. Pérez, and P. W. Rose, *Ten simple rules for writing and sharing computational analyses in Jupyter notebooks*, PLoS Comput. Biol **15** (2019), no. 7, e1007007.
44. H. Shen, *Interactive notebooks: sharing the code*, Nature. **515** (2014), 151–152.
45. A. Suárez-García, E. Arce-Fariña, M. Á. Hernández, and M. Fernández-Gavilanes, *Teaching structural analysis theory with Jupyter notebooks*, Comput. Appl. Eng. Educ. **29** (2021), 1257–1266.
46. M. Tsitoara, Beginning Git and GitHub, Springer, New York, 2020.
47. J. VanderPlas, Python data science handbook: essential tools for working with data, O'Reilly Media, Inc., Sebastopol, CA, USA, 2016.
48. F. M. White and J. Majdalani, Viscous fluid flow, vol. **3**, McGraw-Hill New York, 2006.
49. H. Yamaguchi, Engineering fluid mechanics, vol. **85**, Springer Science & Business Media, Dordrecht, The Netherlands, 2008.

AUTHOR BIOGRAPHIES

Robert Castilla received a Degree in Physics from the Universitat Autònoma de Barcelona (UAB) and a PhD Degree in Applied Physics from the Universitat Politècnica de Catalunya · BarcelonaTech (UPC). He is an associate professor at the Department of Fluid Mechanics and he is a management member of the Centre for Advanced Technologies in Mechanics (CATMech) at the UPC. His research interest is Computational Fluid Dynamics with complex industrial flows: complex dynamic geometries, turbulent flows, rheological fluids, multiphase flows, compressible flows, and bioengineering fluids.

Marta Peña received a Bachelor's degree in Industrial Engineering and a PhD degree in Mathematics from the Universitat Politècnica de Catalunya. BarcelonaTech (UPC). She is an associate professor at the Department of Mathematics and is a member of the Institute of Industrial and Control Engineering at the UPC. She has been deputy director (teacher training manager) of the Institute of Education Sciences at the UPC from 2016 to 2019 and she is deputy director (head of bachelor's degrees and responsible for social commitment) of the Barcelona School of Industrial Engineering (ETSEIB) since 2021. Her research interests are applied mathematics and innovation in higher education, including social aspects of engineering education and student engagement. She is responsible for the SOC-STEM (Social Impact of STEM disciplines) research group.

How to cite this article: R. Castilla and M. Peña, *Jupyter Notebooks for the study of advanced topics in Fluid Mechanics*, Comput. Appl. Eng. Educ. 2023;31:1001–1013. https://doi.org/10.1002/cae.22619