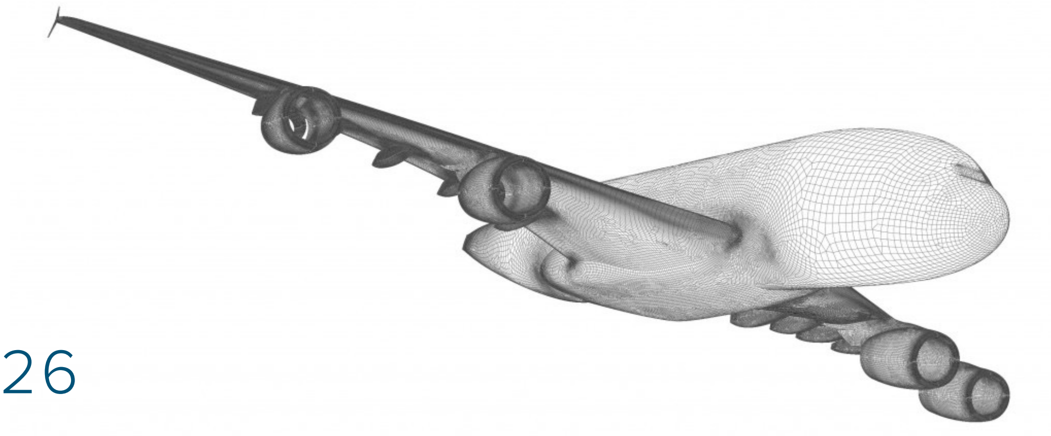

MAE 6226

Aerodynamics-Hydrodynamics

Spring 2020

Instructor: Roberto Capanna, Mechanical and Aerospace Engineering

capanna@gwu.edu | Desk Location: SEH 3343

Office hours by appointment (ask by mail).

Syllabus

Course schedule

Class meets every **Thursday 12:45–15:15 PM**

Location : **Tompkins Hall 410**. Find it on Google Maps: http://goo.gl/maps/ZiHJH

Course aims

This course aims to provide a foundation in the classical theories of aerodynamics of ideal fluids, as they apply to aerospace engineering design, competency in solution methods and understanding of their approximation power and sources of error.

Prerequisites

Formal pre-requisite at GW: **MAE 6221: Fluid Mechanics** (http://bulletin.gwu.edu/search/?P=MAE+6221)

Waivers to the pre-requisite are not granted in general. In exceptional circumstances, where the student can clearly demonstrate command of the subject, it is possible to be evaluated directly by the instructor and allowed to take the course.

Some programming skills are obligatory. **The course will use the Python programming language**, but previous experience with this language is not needed. You will learn as we go along, and we'll provide help and many resources.

This course is aimed at first-year graduate students, but senior undergraduates are also welcome. However, the pre-requisite policy is equally enforced.

You should know vector calculus of multiple variables, including tensor notation, basics of complex-variable calculus and undergraduate-level ordinary and partial differential equations. Undergraduate physics and fluid mechanics is also required background—the following concepts of fluid mechanics are assumed known:

- length scales and the continuum approximation,
- Lagrangian and Eulerian representation,
- streamlines, pathlines, streaklines,
- basic kinematics:

  translation, rotation, deformation,

  vorticity and strain tensor
- conservation laws: continuity and Navier-Stokes (derivation)
- derivation of the vorticity-transport equation

Textbook

There is no required textbook. For reference, we recommend:

J. Katz & A. Plotkin, Low-Speed Aerodynamics, Cambridge University Press (2nd ed., 2001)

J. Moran, An Introduction to Theoretical and Computational Aerodynamics (1984), Dover 2003

For a historical perspective, we recommend:

von Kármán, Aerodynamics. Selected topics in the light of their historical development, Dover Publications (2004), republished from the original 1954 book.

For a physical understanding of aerodynamics, we recommend:

D. McLean, Understanding Aerodynamics: Arguing from the real physics, Wiley & Sons (2013)

The course materials (on Jupyter notebooks) are required reading. Find them in the course GitHub repository: https://github.com/barbagroup/AeroPython. Additional readings may be provided.

Learning objectives

Students will …

1. appreciate the approximations that are made to tackle aerodynamics for engineering purposes, and understand their limitations;
2. learn the classical use of potential-flow theory in aerodynamics;
3. implement computational solutions of potential flow with panel methods;
4. read and learn to summarize and critique literature of classical aerodynamics;
5. gain understanding of how mathematical analysis of ideal flows can be used for engineering design of real systems;
6. gain experience documenting and presenting the fruits of their study.

Course online resources

We will use several online resources to share materials, work together and be connected outside of class meetings. Each one of these resources has its strengths and we combine them all to provide you with the best course experience.

If this is you: "I prefer everything in one place!" Resist the urge—think about how many apps and services you use in your daily life, and how each serves a purpose. The same applies to learning in a digital world. We use these tools because they add something unique to the course experience.

If you are confused at first: just ask!

We use the following digital tools in this course:

Blackboard — Course materials in the form of executable lecture notes using **Jupyter Notebooks** will be published on blackboard.

Jupyter Hub — We installed a server at GW Academic Technologies to provide you with Python on the cloud. You can create, save, upload and edit Jupyter Notebooks on this server, making them available to you from any computer or internet-enabled device. Get Python without installing anything!

Go to: http://go.gwu.edu/jupyter and log in using your GW single-sign-on credentials.

GitHub Classroom — https://classroom.github.com — An open-source platform that enables GitHub workflow for education. We'll use the platform to distribute and collect assignments. You'll get access to the assignments via an invitation URL. Once you have completed the assignment, you'll submit it to GitHub Classroom. (GitHub Classroom requires a GitHub account.)

Socrative — For in-class student responses and quizzes, using your own internet-enabled mobile device, or the lab computer.

Go to: http://www.socrative.com/ & join ROOM: **MAE6226**

In summary, you'll have to create a GitHub account. You'll use your GW student account to log in the JupyterHub server for cloud-based Jupyter Notebooks. You'll get a URL via email to join the Slack team. You'll receive invitations to start assignments and you'll push your solution to GitHub Classroom. There is no account needed for Socrative.

Assessment policies

Assessment will be based on (live or online) quizzes (10%), coding assignments submitted throughout the semester (20%), a midterm exam (30%) and a final project (30%). Students' written work will be online and peer assessment will sometimes inform the grade. The final project should consist of a numerical solution to an aerodynamics problem aimed at a design situation and a written report in the form of a Jupyter notebook.

Attendance to class meetings is required and any absence needs to be excused by the instructor ahead of time.

More details will be discussed at a later date.

GOOHF cards

Each student gets four "get-out-of-homework-free" cards, or GOOHF cards. You can claim one card (by mail) to get a one-day extension on any homework deadline. Once you run out of GOOHF cards, penalty for late assignments is one letter grade per day (equivalent to around 10 points off on a 100-point scale).

How to work with the course materials

This course orbits around sets of Jupyter Notebooks, created as learning objects, documents, discussion springboards, artifacts for you to engage with the material.

Each course section consists of three or four Jupyter notebooks, containing detailed notes and step-by-step code instructions.

This is how you must work with the notebooks:

Read along, like a textbook, and take handwritten notes as you do. Follow all the mathematical derivations, and look up things that you are confused with. Then execute the embedded code, play with it, change things and see what happens. Finally, write your own version of the codes, either in your own clean notebook or as a Python script.

Academic honor code & plagiarism

You must, of course, be familiar with GW's code of academic integrity. Find it at: https://studentconduct.gwu.edu/code-academic-integrity

Among the examples of academic misconduct, plagiarism is a particular source of confusion for students. The simple rule is that **you have to write your assignments yourself**. You should never copy text verbatim from other sources, unless it is a quote which is properly marked as such and properly attributed. In your work, you must always include references to any sources you used.

Please bear in mind the following extra guidance for y our project:

- You can of course discuss your work with your classmates. This course encourages collaboration. But the project should be your own original writing and code.
- You obviously will be searching for ideas and help on the Internet. Everybody does that. But your project should be your own original writing and code.
- There may be a couple of project suggestions from students that sound similar. Please think about how you will make your work original and distinct from another on the same topic.
- Do not copy any text from a source. Always write your original text. Cite all your sources!!
- Just to be clear: **even one sentence copied verbatim from a source is considered plagiarism.**
- Google is your friend. And your worst enemy. Handle with care.

You also must complete all the coding assignments individually, although collaboration is allowed and encouraged as you progress. Just **don't copy** other people's code.

Code of conduct

This course by design reflects the ethics of open-source software communities. This means that we value everyone's participation, we strive for transparency and inclusion, and we promote collaboration. We want every student to have a rewarding, fruitful learning experience. To achieve this, everyone is expected to show courtesy and respect towards each other. The following Code of Conduct (CoC)[1] is agreed upon by those taking this course:

[1] Adapted from the Software Carpentry CoC, itself evolved from those of PyCon and the Ada Initiative.

- All communication should be appropriate for a professional audience including people of many different backgrounds.
- We do not tolerate harassment in the course, in person or online. Harassment refers to offensive verbal or written comments in reference to gender, sexual orientation, disability, physical appearance, body size, race, or religion; sexual images in public spaces; deliberate intimidation, stalking, following, harassing photography or recording, sustained disruption of class meetings, inappropriate physical contact, and unwelcome sexual attention.
- Be kind to others. Do not insult or put down other participants.
- Behave professionally. Avoid jokes that could be interpreted as sexist, racist, or exclusionary. Remember that humor is a social act.

If you feel someone is violating the CoC, speak up! Say: “Hey, that’s not cool. We follow a code of conduct in this class, remember?” If the behavior persists, send a private email to the course instructor explaining the situation.

Come prepared for the first day of class

Brush up on your pre-req

The prerequisite for MAE 6226 is a fluid mechanics course. The catalog description is:

> **MAE 6221**. Fluid Mechanics. 3 Credits. Continuum, kinematics of fluids; stress and strain rate tensors; fundamental equations of viscous compressible flows. Irrotational flows; sources, sinks, doublets, and vortices. Laminar flow of viscous incompressible fluids; boundary-layer concept. Prerequisite: approval of department.

If you feel that you are rusty with the standard material onfluid mechanics, we recommend that you brush up with the following supplementary material:

- Videos on YouTube:

  Eulerian and Lagrangian descriptions: https://youtu.be/mdN8OOkx2ko

  Flow visualization: https://youtu.be/nuQyKGuXJOs

  Derivation of the continuity equation: https://youtu.be/35unQgSaT88

  Derivation of the equation of conservation of momentum: https://youtu.be/aBu4QtwgbGk
- Lecture videos on iTunes U, Fluid Mechanics course by Prof. Barba: https://itunes.apple.com/us/itunes-u/fluid-mechanics-2010-eng-me303/id452560560

Create your accounts

- Open a GitHub account.

Get a paper notebook or binder for this class! And finally …

KEEP
CALM
AND
READ THE
SYLLABUS