Technische Universität Braunschweig

Fakultät für Maschinenbau
Institut für Strömungsmechanik

Machine Learning in Fluid Dynamics

Dr.-Ing. Andre Weiner, Prof. Dr.-Ing. Rolf Radespiel
Technische Universität Braunschweig | Institut für Strömungsmechanik
a.weiner@tu-braunschweig.de | Telefon +49 (0) 531 391-94266

Stiftung Innovation in der Hochschullehre

README.md

Stiftung Innovation in der Hochschullehre

This lecture is funded by the foundation for innovation in higher education.

Machine learning in computational fluid dynamics

This repository contains resources accompanying the lecture machine learning in computational fluid dynamics provided by the Institute of Fluid Mechanics at TU Braunschweig. **Note that slides, notebooks, and other resources will be regularly updated throughout the term.**

Lectures

If equations in the lecture notebooks do not get rendered properly on Github, download the notebook and open it using `jupyter-lab` (refer to the first exercise session for an overview of dependencies and installation instructions).

#	topic	slides	notebook
1	Course overview and motivation	link	view
2+3	Finite-volume-based simulations in a nutshell	link	view
4+5	Introduction to machine learning	link	view
6	Surrogate modeling for discrete predictions	link	view

Concept and starting point

- Application of state-of-the-art machine learning techniques for the analysis, modelling, and optimization of fluid flow applications
- Story telling – machine learning and simulation techniques introduced in the context of real world problems and applications
- Direct knowledge transfer from latest research to engineering curriculum
- Targets mainly computational, mechanical, and aerospace engineers
- Hybrid format – recorded in-person/online lecture and exercise sessions
- Lecture material and communication in English language
- Lecture started in 2021/2022 winter term and got significantly revised and extended during the 2022/2023 winter term thanks to **ProDigi**
- Currently 15 lectures and exercises (11 revised, 2 new, 2 in preparation)

← Left: The lecture content is organized as a Github repository. The lecture material is freely available at https://github.com/AndreWeiner/ml-cfd-lecture.

Technology stack

- Slides are created with **Reveal.js** and hosted on Github → slides are device independent (laptop, tablet, phone), platform independent (Windows, Linux, MacOS, Android, iOS) and always accesible online via web browser
- **Jupyter notebooks** as lecture and exercise scripts → Jupyter notebooks are interactive documents containing text, formulas, source code, and interactive visualizations; the notebooks can be viewed online (static mode) or locally on a laptop/workstation (interactive)
- Version control with **Git** and **Github** → lecture material is version-controlled and gets updated regularly throughout term; students are required to check for the latest updates and to “pull“ if necessary
- Reproducibility of scientific data through **virtualization** → software and programming libraries are distributed as software containers
- Open-source only → the lecture material is published under a non-restrictive Creative Commons license; free simulation (OpenFOAM, Basilisk) and machine learning packages (PyTorch, ScikitLearn)

Challenges

Compatibility – issues between BigBlueButton, iPad and Linux/Ubuntu
Time – update and creation of content is demanding
Technical requirements – many students don’t have access to a Linux PC
Access – there is demand for thrid-party access but no legal basis for participation
Active engagement – most students don’t interact hands-on with material (exercises and notebooks)

Future skills

hands-on **programming** in Python and C++

Data analysis and **visualization** of real data

Machine learning in action – transfer of cutting edge research to teaching

Data literacy – state-of-the-art technology stack

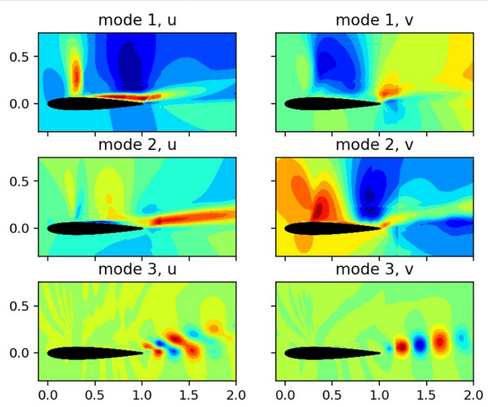
Professional online **communication**

Lecture notebooks

Modal analysis of transonic shock buffet on airfoils

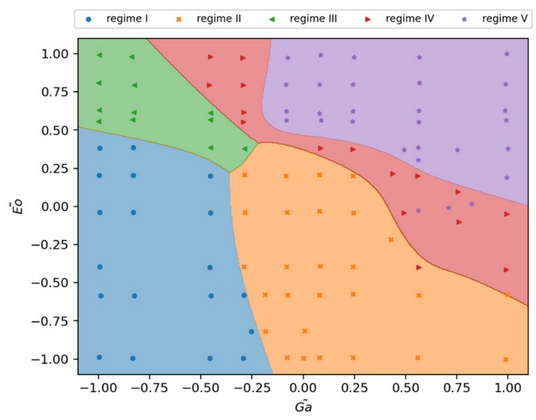
```
In [37]: fig, axarr = plt.subplots(3, 2, figsize=(6, 5), sharex=True, sharey=True)

for i in range(3):
    axarr[i, 0].tricontourf(xa, ya, U[:n_points, i], levels=14, cmap="jet")
    axarr[i, 1].tricontourf(xa, ya, U[n_points:, i], levels=14, cmap="jet")
    for ax in axarr[i, :]:
        add_naca0012_patch(ax)
        ax.set_aspect("equal", 'box')
        ax.set_xlim(-0.1, 2.0)
        ax.set_ylim(-0.3, 0.75)
    axarr[i, 0].set_title(f"mode {i+1}, u")
    axarr[i, 1].set_title(f"mode {i+1}, v")
plt.savefig(f"{output}/naca_pod_modes.png", bbox_inches="tight")
```




```
In [90]: xx, yy = pt.meshgrid(
             pt.linspace(-1.1, 1.1, 500),
             pt.linspace(-1.1, 1.1, 500)
         )
         X = pt.stack((xx.flatten(), yy.flatten())).T
         prob = pt.nn.functional.softmax(regime_model(X), dim=1)
         prediction = pt.argmax(prob, dim=1) + 0.01

         fig, ax = plt.subplots()
         cmap = ListedColormap([f"C{i}" for i in range(5)])
         ax.contourf(xx, yy, prediction.reshape(xx.shape), cmap=cmap, alpha=0.5)
         for regime, marker in zip(regimes, markers):
             ax.scatter(logData[logData["regime"] == regime].Ga, logData[logData["regime"] == regime].Eo,
                        marker=marker, s=10, label="regime {}".format(regime))
         ax.set_xlabel(r"$\tilde{Ga}$")
         ax.set_ylabel(r"$\tilde{Eo}$")
         ax.legend(loc='upper center', bbox_to_anchor=(0.5, 1.1), ncol=5, fontsize=8)
         plt.savefig(join(output, "network_multiclass_prediction.svg"), bbox_inches="tight")
```



Path stability classification of rising bubbles

Just the numbers

72 enrolled students from computational, mechanical, aerospace, electrical and other engineering diciplines

Lecture repository has **~ 85 unique visitors** per week and gets **“cloned” ~15 times** per week

~ 640 weekly visualizations of lecture material

PROMOTING DIGITAL EDUCATION THROUGH GLOBAL INTERCONNECTION