Syllabus

Course Description

Course Title	Physics Informed Neural Networks
Course Code	71082
Course Title Additional	
Scientific-Disciplinary Sector	IINF-05/A
Language	English
Degree Course	PhD Programme in Computer Science
Other Degree Courses (Loaned)	
Lecturers	Dr. Alessandro Bombini, Alessandro.Bombini@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/53352
Teaching Assistant	
Semester	All semesters
Course Year/s	2025-2026
CP	2
Teaching Hours	20
Lab Hours	0
Individual Study Hours	50
Planned Office Hours	0
Contents Summary	The course introduces the concept of Physics Informed Deep Neural Networks (PINN), discuss its implementation from scratch in PyTorch and using advanced ad-hoc developed open-source libraries such as Nvidia PhysicsNemo for addressing real-world problems in various fields (engineering, physics, petroleum reservoir). We discuss recent topics such as Mixture-of-Models, Neural Operators, Physics-Informed Kolmogorov-Arnold Networks and Physics-Informed Computer Vision.
Course Topics	Short Topic list • General Introduction to the Course: PDEs, Functional Analysis,

	Monte Carlo Integration • An Introduction to numerical resolution of PDEs • Finite Difference Methods to solve PDEs with Python • Introduction to PINNs: forward problems, inverse problems and parametric PINNs • Solving Heat equation with PINN in PyTorch (lightning) • Advanced PINNs methods - Learning strategies, Architectures, Losses, and other approaches • Introduction to PhysicsNemo-SYM to solve PDEs with PINNs • Advanced methods for PINNs in PhysicsNemo • PIKANs and Neural Operators • Solving Darcy Flow with DeepONets and FNOs in PhysicsNemo Frontal Lectures: Lecture 1: General Introduction to the Course (2h30m) Lecture 2: Introduction to Numerical Resolution of PDEs (1h30m) Lecture 3: Introduction to Physics Informed Neural Networks (1h30m) Lecture 4: Advanced PINNs (3h00m) Lecture 5: Neural Operators (3h00m) Hands On: 1. FDM with Python 2. Burgers PINN 3. PhysicsNemo-SYM-intro 4. Advanced PhysicsNemo 5. Darcy Flow with Neural Operators
Keywords	Deep Learning; Physics-Informed Neural Networks
Recommended Prerequisites	Basics of Python; Real Analysis; Numerical Methods; Machine Learning
Propaedeutic Courses	Basics of Python; Real Analysis; Numerical Methods; Machine Learning
Teaching Format	Each lecture will consist of a frontal lecture (using presentation materials) and an hands-on section (using Google Colab, Jupyter Lab)
Mandatory Attendance	Attendance is not compulsory, but non-attending students have to contact the lecturers at the start of the course to agree on the modalities of the independent study.

Specific Educational Objectives and Learning Outcomes	The goal of the course is to introduce the concept of Physics Informed Deep Neural Networks (PINN), discuss its implementation from scratch in PyTorch and using advanced ad-hoc developed open-source libraries such as nvidia-modulus for addressing real-world problems in various fields (engineering, physics, petroleum reservoir). We discuss recent topics such as Mixture-of-Models, Neural Operators, Physics-Informed Kolmogorov-Arnold Networks and Physics-Informed Computer Vision. Knowledge and understanding • D1.1 – Ability to analyse and solve complex problems in computational science by integrating physics-informed neural networks with advanced numerical methods. • D1.2 – Ability to read, understand, and critically evaluate state-of-the-art scientific literature on PINNs, Neural Operators, and Physics-Informed Computer Vision. Applying knowledge and understanding • D2.1 – Ability to design and implement PINNs from scratch, demonstrating mastery of both theoretical and practical aspects. • D2.2 – Ability to apply innovative architectures (e.g. Mixture-of-Models, Kolmogorov-Arnold Networks) to extract knowledge from complex, high-dimensional physical systems. Making judgements • D3.1 – Ability to autonomously select and integrate specialist documentation, libraries, and datasets to advance research in physics-informed AI. • D3.2 – Ability to work with broad autonomy in multidisciplinary projects, taking responsibility for the design and validation of computational experiments. Communication skills • D4.1 – Ability to present PINN-based research results clearly and effectively to both specialist and non-specialist audiences, including through scientific publications. Learning skills • D5.1 – Ability to independently extend knowledge in emerging areas of physics-informed machine learning, keeping pace with rapid developments in AI and computational science.
Specific Educational Objectives and Learning	

Outcomes (additional info.)	
Assessment	Option a: Discussion of a research work on the topic, selected by the student and accepted by the instructor; it must be presented orally with a presentation and with a Git repo offering the students implementation of the code Option b: Resolution of a small research problem discussed jointly with the instructor; presented either orally with a brief presentation or a written essay, and a git repo.
Evaluation Criteria	The exam is pass/fail and no marks are awarded. Relevant for the assessment are the following: clarity of exposition, ability to summarize, evaluate, and establish relationships between topics, ability to present scientific notions, ability to evaluate research results by others.
Required Readings	All the required reading material including slides and lecture notes will be provided during the course and will be available in electronic format. Materials for hands-on sessions will be made available on the course github repository.
Supplementary Readings	The lecture slides for the frontal lecture are made available: • Lecture 1: General Introduction to the Course (2h30m) • Lecture 2: Introduction to Numerical Resolution of PDEs (1h30m) • Lecture 3: Introduction to Physics Informed Neural Networks (1h30m) • Lecture 4: Advanced PINNs (3h00m) • Lecture 5: Neural Operators (3h00) The Lecture Notes will be made available as a .pdf, possibly published CC0 on ArXiv, and will be divided in 5 chapters, following the course structure (i.e., the lectures) Hands On GitHub URL: https://github.com/androbomb/PINN_Course_2026

The github repo is currently organised as:

- 1. FDM with Python
- 2. Burgers PINN
- 3. PhysicsNemo-SYM-intro
- 4. Advanced PhysicsNemo
- 5. Darcy Flow with Neural Operators

In the main page there is a `Readme.md` containing the information to set up the environments/apptainer containers needed to run the Notebooks.

Lecturer Web page:

https://androbomb.github.io

In the teaching section are available the materials for previous editions of the course, held at the University of Florence

Extended topics:

Chapter 1: Lecture 1: General Introduction to the course

1.1 A brief introduction: why should I care?

1.2 A brief introduction: what is a Partial Differential Equation?

1.2.1 A first example: the Heat equation in 1+1 dimensions

1.2.2 The effect of Robin conditions: the loose end violin string

1.3 Recap of Functional Analysis

1.3.1 From functions to functionals

1.3.2 The Cybenko's Theorem

1.4 Recap of Montecarlo Integration methods

1.4.1 Relevance for Machine Learning

Chapter 2: Lecture 2: An introduction to numerical resolution of differential equations

2.1 Finite Difference Methods (FDM)

2.1.1 A first application of finite differences: Sobol Edge detection algorithm

2.1.2 Using FDM to solve Poisson Equation: the Jacobi method

	2.1.3 An application in Computer Vision: image inpainting with Laplace Equation 2.1.4 Solving Burgers equation with FDM 2.1.5 A heuristic connection to CNN and Graphs: Stencil representation & Heat equation on Graphs 2.2 Finite Element Methods (FEM) 2.2.1 Ritz vs Galerkin method 2.3 Finite Volume Methods (FVM) 2.4 Meshless methods; Kansa's approach 2.4.1 The Kansa approach: RBF for numerical PDE resolution Chapter 3: Lecture 3: Introduction to Physics Informed Neural Networks 3.1 Forward Problems: Vanilla PINNs 3.1.1 An historical note 3.1.2 A brief comment on vanilla PINNs 3.1.3 A key comment on vanilla PINNs: strenghts... and limits 3.2 Inverse & Parametric Problems 3.2.1 PINNs for inverse problems 3.2.2 PINNs for parametric problems 3.3 A brief introduction to the Learning Theory of PINNs 3.3.1 The PINN Learning Problem as Risk Minimization 3.3.2 Approximation Error 3.3.3 Estimation (Quadrature) Error 3.3.4 Optimization Error 3.3.5 Total Error Decomposition 3.3.6 Training Dynamics of PINNs: connection to Optimization Dynamics and Information Flow Chapter 4: Lecture 4: Advanced PINNs methods - Learning strategies, Architectures, Losses, and other approaches 4.1 Dynamic Hyperparameter Optimisation

	4.1.1 GradNorm 4.1.2 Homoscedastic Task Uncertainty 4.1.3 Learning Rate Annealing 4.1.4 SoftAdapt 4.1.5 ReLoBRaLo 4.1.6 ConFIG 4.1.7 Neural Tangent Kernel-based Adaptive Weighting 4.2 Advanced Schemes 4.2.1 Sobolev training 4.2.2 Quasi-random sampling 4.2.3 Importance sampling 4.2.4 Approximate distance functions and R-functions for exact boundary conditions 4.2.5 Curriculum learning 4.3 Architectures 4.3.1 Adaptive Activations and Weight Factorisation 4.3.2 Deep Galerkin Method, Deep Ritz Method, and Variational PINNs 4.3.3 Spectral Bias and Fourier Embeddings 4.3.4 Sinusoidal Representation Networks 4.3.5 Transformers in PINNs 4.3.6 Mixture-of-Experts 4.4 Optimisation Schemes 4.4.1 Optimisers: second order over first order 4.4.2 Loss functions, residuals, and geometry 4.5 Recap-ish: An Expert's Guide to Training Physics-informed Neural Networks Chapter 5: Neural Operators 5.1 Learning Operators, Part I: Deep Operator Networks

	5.1.1 Why learning operators? 5.1.2 The Universal Approximation Theorem for functionals (and operators) 5.1.3 Deep Operator Networks 5.1.4 Physics-Informed DeepONets 5.2 Using a different representation theorem for functions: Kolmogorov-Arnold Network 5.2.1 Kolmogorov-Arnold Network 5.2.2 Alternative KAN implementations 5.2.3 KAN for PDEs: PIKAN and DeepOKan 5.3 Learning Operators, Part II: Neural Operators - formal theory 5.4 Learning Operators, Part III: Neural Operators - Architectures 5.4.1 Fourier Neural Operator 5.4.2 Adaptive Fourier Neural Operator 5.4.3 Physics Informed Neural Operator 5.4.4 Graph Neural Operator 5.5 Learning Operators, Part IV: Neural Operators - Applications 5.5.1 Neural operators for Weather forecasting: FourCastNet 5.5.2 Neural operators for Reservoir Simulation 5.5.3 Neural operators for Foundational Models --- Condensed Bibliography: Maziar Raissi, Paris Perdikaris, George Em Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv 1711.10561 Maziar Raissi, Paris Perdikaris, George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential

	equations. J. Comp. Phys. 378 pp. 686-707 DOI: 10.1016/j.jcp.2018.10.045 Toscano, Juan Diego et al. “From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning.” (2024). arXiv:2410.13228 Chayan B., Kien N., Clinton F., and Karniadakis G.. 2024. Physics-Informed Computer Vision: A Review and Perspectives. ACM Comput. Surv. (August 2024). https://doi.org/10.1145/3689037 Cuomo, S., Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next. Journal of Scientific Computing, 92. ArXiV 2201.05624 The full bilbiography will be furnished with the lecture notes.
Further Information	Python, PyTorch, Nvidia PhysicsNemo 2504, JupyterLab/Hub
Sustainable Development Goals (SDGs)	Quality education