UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB DISTRIBUTED DATA INTELLIGENCE AND TECHNOLOGIES LAB

https://www.snap4City.org

NeuroSymbolic Artificial Intelligence at Scale

Marco Fanfani, marco.fanfani@unifi.it

https://www.disit.org/

Parte: 2.2 (2025-26) Physics-Informed Neural Networks

Introduction

Physics-Informed Neural Networks (PINNs)

Solving Partial Differential Equations through Scientific Machine Learning

INTRODUCTION

Learning Objectives

- By the end of this session, you will be able to:
  - Define Scientific Machine Learning (SciML) and its role in modern engineering.
  - Understand the core architecture of Physics-Informed Neural Networks (PINNs).
  - Explain how Automatic Differentiation (AD) replaces traditional numerical differentiation.
  - Formulate a physics-based loss function using PDE residuals and boundary conditions.
  - Evaluate the trade-offs between PINNs and traditional methods like FEM and FVM.

What is Scientific Machine Learning (SciML)?

- **Definition**: SciML is a novel conceptual approach that **integrates scientific knowledge** (physical laws, symmetries, conservation principles) with **data science**.
- The Key Difference:
  - Traditional ML: Often a 'black box' that requires **massive labeled data** and **lacks interpretability**.
  - SciML: Combines **neural network** flexibility with **rigorous mathematical physics**, allowing accuracy even with small/noisy datasets.

The Core Concept: Fusing Physics and Data

- How it Works: **Embed governing equations** (e.g., Navier-Stokes, Schrödinger) **directly into the neural network's training process**.
  - **Neural Network:** Acts as a universal **function approximator**.
  - **Physics Knowledge:** Expressed as Partial Differential Equations (**PDEs**).
  - **Optimization:** Parameters are adjusted to **minimize a *Physics Loss***, i.e. a penalty for violating physical laws.

Motivation: Why do we need PINNs?

- **Limitations of Traditional Methods (CFD, FEM, FVM):**
  - **Computational Cost**: High hardware and time requirements.
  - **Meshing Effort**: Intensive manual labor for grid generation.
  - **The Inverse Problem**: Solvers struggle to 'discover' unknown parameters.
- **The PINN Advantage:**
  - **Mesh-Free**: Evaluates equations at random collocation points.
  - **Efficiency**: Instantaneous inference once trained.
  - **Robustness**: Reconstructing fields from sparse or hidden data.

Partial Differential Equations (PDEs)

- **Why PDEs Matter:**
  - Most physical laws are expressed as **Partial Differential Equations** (PDEs).
  - They describe how variables (*velocity*, *pressure*, *temperature*) change across *space* (x,y,z) and *time* (t).
  - Universal Laws: The standard tool for modeling momentum, energy balances, and conservation of mass.

Modeling Physical Phenomena

- **Core Disciplines Reliant on PDEs:**
  - **Fluid Dynamics:** Airflow over wings, blood flow in arteries.
  - **Structural Mechanics:** Stress and strain in buildings or mechanical parts.
  - **Geophysics:** Seismic wave propagation and earthquake assessment.
  - **Biomedicine:** Axonal electrophysiology and 'Brain-on-Chip' devices.

Mathematical Formulation of a PDE Problem

- To solve a physical problem, we must define:
  - **Governing Equations:** The specific PDE (e.g., Navier-Stokes, Heat Eq).
  - **Domain (Ω):** The physical space where the phenomenon occurs.
  - **Initial Conditions (IC):** The state of the system at t=0.
  - **Boundary Conditions (BC):** Constraints on edges (e.g., No-slip walls).

Transient vs. Steady-State Scenarios

- **Transient Phase:** The time-varying state before stabilization (e.g., startup).
- **Steady-State:** When the flow field becomes stable over time.
- **Industrial Example:** Flow around parts in autoclaves for material curing.
- **PINN Advantage: Excellent at capturing time-dependent evolutions where traditional solvers struggle with stability**.

The Workhorse of Engineering: Navier-Stokes

- Navier-Stokes equation is essentially Newton's Second Law (F = ma) rewritten to describe how a "blob" of fluid moves.
- Solving Navier-Stokes is the *Holy Grail* of computational physics, essential for everything from aerospace to weather forecasting.
- In a solid object, acceleration is simple. In a fluid, it's tricky because **the fluid is moving and the spot it's moving into might be moving at a different speed**.
- There are three main forces that tell the fluid where to go:
  - Pressure: Fluid gets pushed from high-pressure areas to low-pressure areas (like squeezing a tube of toothpaste).
  - Viscosity (Friction): This is the fluid's "thickness." It acts like a brake. Thick fluids (honey) have high friction and resist movement; thin fluids (water) move easily.
  - External Forces: This is usually just gravity pulling the fluid down, or perhaps a pump forcing it forward.

The Workhorse of Engineering: Navier-Stokes

- A fundamental system of **non-linear PDEs describing fluid motion**.

Velocity (vx,vy,vx) *Pressure*

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\,\nabla^2\mathbf{u} + \mathbf{f}$$

- ρ density: the "mass" part. A heavy fluid like mercury is harder to move than a light gas
- $\frac{\partial \mathbf{u}}{\partial t}$ unsteady acceleration: how the velocity at a fixed-point changes over time
- $\mathbf{u}\cdot\nabla\mathbf{u}$ convective acceleration: It represents the change in velocity because the fluid has moved to a new position where the speed is different
- $-\nabla p$ pressure gradient: Fluids naturally move from high pressure to low pressure
- $\mu\,\nabla^2\mathbf{u}$ viscous term: This represents internal friction
- $\mathbf{f}$ external forces: These are "body forces" that act on the fluid from the outside (e.g. gravity)

The Workhorse of Engineering: Navier-Stokes

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\,\nabla^2\mathbf{u} + \mathbf{f}$$

- This is the **momentum equation** (conservazione della quantità di moto)
- in N-S it is also required the **continuity equation** (conservazione della massa)

$$\nabla\cdot\mathbf{u} = 0$$

The Workhorse of Engineering: Navier-Stokes

- N-S equations cannot be solved analytically for real-world scenarios. We can only solve them for very simple, idealized cases.
- In Navier-Stokes, the velocity of the fluid at a certain point determines how that same velocity will change. This creates a feedback loop.
- Turbulence is the "unsolved" part of classical physics. When a fluid moves fast, it breaks down into eddies and vortices of all different sizes—from giant whirlpools to microscopic swirls.
- Mathematically, we don't even know if a solution **always exists** in 3D without "blowing up."

Classical CFD Approaches: FEM, FVM, and FDM

- The Standard Toolbox:
  - **Finite Element Method (FEM):** Discretizes the domain into small sub-regions (elements).
  - **Finite Volume Method (FVM):** Transforms PDEs into integral forms (standard for CFD like OpenFOAM).
  - **Finite Difference Method (FDM):** Replaces continuous derivatives with discrete grid approximations.
- *Common Goal: Transform continuous PDEs into solvable algebraic systems.*

FVM Fundamentals and Integral Transformation

1. **Spatial Discretization**: The fluid domain is divided into a finite number of control volumes (V_i), typically organized in a structured mesh.
2. **Integral Form**: Differential PDEs (like Navier-Stokes) are converted into integral representations to ensure physical conservation (mass, momentum, energy) within each cell.
   - Mathematical Example (Continuity Equation):
     - Differential: $\nabla \cdot \mathbf{u} = 0$
     - Volume Integral: $\int_{V_i} \nabla \cdot \mathbf{u}\, dV = 0$
   - Using the Divergence Theorem the volume integral is converted to a surface integral:
     - $\int_{V_i} \nabla \cdot \mathbf{u}\, dV = \oint_{\partial V_i} \mathbf{u}\mathbf{n} dA$, where $\mathbf{n}$ is the outward normal vector.

Physical Meaning: **The net mass flow in and out of the control volume must be zero**.

FVM Fundamentals and Integral Transformation

3. **Flux Approximation:** Surface integrals are discretized as a summation over the faces (f) of the volume: $\sum_f \mathbf{u}_f \mathbf{n}_f \Delta A_f = 0$ (ΔA_f is the face area)
   - **Interpolation**: Face velocities are estimated by interpolating neighbour cells considering centroids (in 2D, P of the cell and N, S, E, W of neighbours).
   - Substituting these approximations yields a spatial finite difference for each cell
4. **Equation Resolution**: All cell contributions are gathered into a large system of algebraic equations to compute the final velocity and pressure fields.

Why we use CFD

- Since we can't solve them with pen and paper (analytically), we use supercomputers to approximate them.
- We break the fluid down into millions of tiny cubes (a "mesh").
- The computer calculates the forces for each tiny cube step-by-step.
- This is how we design airplanes, cars, and weather models today, but PINNs can provide alternative ways...

The Bottleneck of Meshing

- **Manual Effort:** Significant human labor required to define and refine dense meshes.
- **Geometry Sensitivity:** Sharp corners or complex obstacles make mesh generation difficult.
- **Fixed Grids:** Geometry changes often require a complete re-generation of the mesh.

Computational Intensity and Time Constraints

- Hardware Requirements: High-fidelity simulations need expensive clusters/CPUs.
- Iterative Stepping: Solutions must be computed for every single discrete time step.
- Execution Time: A 2D scenario can take several hours even with simple physics on small domains.

Dealing with High-Dimensionality and Complexity

- **Grid Explosion:** Exponential growth of cells as spatial dimensions or resolution increase.
- **The Inverse Problem:** Classical solvers struggle to 'discover' physics from sparse data.
- **Data Integration:** Difficult to incorporate experimental measurements into the numerical scheme.

Dealing with High-Dimensionality and Complexity

Feature	Forward (Classical)	Inverse (Modern Challenge)
Input	Physics + Initial Conditions	Sparse/Messy Observation Data
Output	Future state of the fluid	Hidden parameters or "the law" itself
Analogy	Following a recipe to bake a cake.	Tasting a cake and trying to guess the secret ingredients.
Problem	Requires perfect starting data.	Struggles to fill in the gaps without AI.

Numerical Stability and Error Sources

- **Stability Limits:** Must respect the CFL number, requiring tiny time steps (e.g., 5e-4s).
- **Numerical Errors:** Prone to truncation and round-off errors during differentiation.
- **Accuracy vs. Cost:** Precision requires density, which increases memory and time usage.

Moving Towards Mesh-Free Learning

- A Shift in Paradigm:
  - **Traditional Solvers:** Grid-based (rigid connectivity, predefined discretization).
  - **Mesh-Free Paradigm:** PINNs do not require a rigid computational grid.
  - **Collocation Points:** Evaluation at scattered points throughout space-time.
  - **Geometry:** Modeling irregular domains (blood vessels, tunnels) without manual meshing.

Neural Networks as Universal Function Approximators

- PINNs rely on deep neural networks, proven universal function approximators.
- NNs can represent any continuous function on compact subsets of R^d.
- Adaptability: Unlike fixed polynomials, NNs adapt to the complexity of the solution.

Collocation Points vs. Fixed Grids

- **Traditional CFD:** Iterative calculation at fixed nodes for every time step.
- **PINN Approach:** Randomly samples (x,t) points from the domain during training.
- **Optimization:** Turns a simulation problem into a continuous optimization problem.

Inverse Problems and Data Fusion

- The Unique Advantage:
- **Inverse Discovery:** Identify unknown parameters (viscosity, reaction rates) from data.
- **Hidden Fluid Mechanics:** Reconstruct 3D fields from simple flow visualizations (smoke/dye).
- **Industry 4.0:** The cornerstone for Digital Twins and real-time monitoring.