AI in the Sciences and Engineering 2024: Lecture 4

Siddhartha Mishra

Seminar for Applied Mathematics (SAM), D-MATH (and),
ETH AI Center,
ETH Zürich, Switzerland.

What you learnt so far

- What is Deep Learning ?
  - Deep Neural Networks.
  - Large labelled datasets.
  - Gradient descent training algorithms.
  - Ability to generalize.
- The next several lectures: Use of Deep Learning for solving PDEs

What are PDEs ?

The general paradigm is : given a blueprint, to find the corresponding recipe. Much of the activity of science is an application of that paradigm : given the description of some natural phenomena, to find the differential equations for processes that will produce the phenomena.

- A prize quote from Herbert Simon (Nobel prize in Economics, 1978)
- Given a scientific problem, find a Partial Differential Equation (PDE) describing it

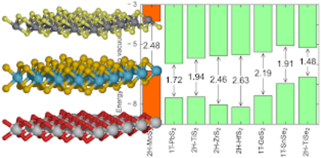
What are PDEs ?

- Generic form of PDEs:

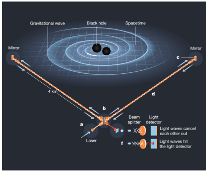
$$\mathcal{F}\left(x, t, u, \nabla u, u_t, u_{tt}, \nabla^2 u, \ldots, \ldots\right) = 0.$$

- Here $u : D \times (0, T) \mapsto \mathbb{R}^m$ for $D \subset \mathbb{R}^d$

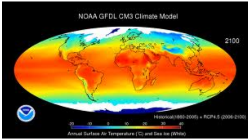
Partial Differential Equations (PDEs)



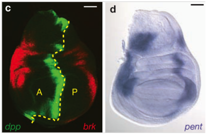
Atoms



Black Holes

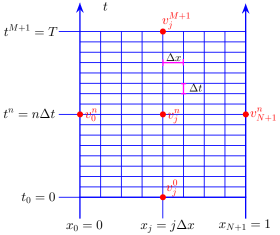


Climate

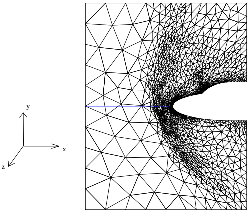


Fly wing

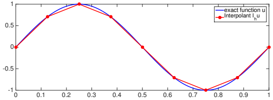
Traditional Numerical Methods



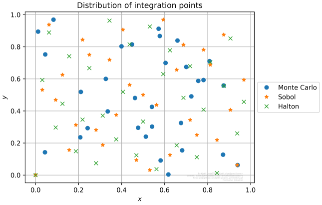
Finite Difference



Finite Volume

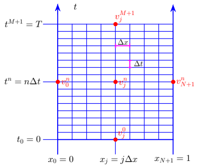


Finite Element



Collocation

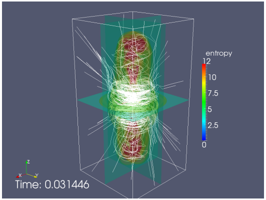
- Different flavors of Runge-Kutta for Time Integration



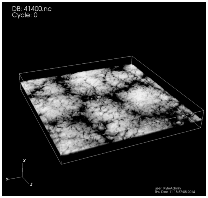
- Heat equation is $u_t = u_{xx}$, Initial conditions: $u(x,0) = \bar{u}(x)$, Boundary conditions: $u(0,t) = u(1,t) = 0$.
- $u_t(x_j, t^n) \approx \frac{v_j^{n+1} - v_j^n}{\Delta t}$, $u_{xx} \approx \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{\Delta x^2}$
- Finite Difference Scheme:

$$v_j^{n+1} = (1 - 2\lambda) v_j^n + \lambda (v_{j-1}^n + v_{j+1}^n), \quad \lambda = \frac{\Delta t}{\Delta x^2}$$

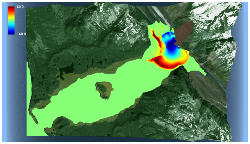
Numerical Methods are very Successful



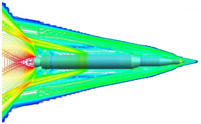
Supernovas



Clouds



Tsunamis



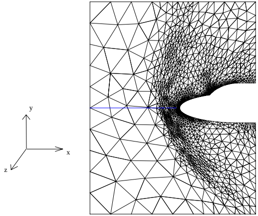
Rockets

Wheres the Caveat ?

- For stability (CFL): $\Delta t \sim \Delta x^2$.
- Error: $E = E_{\Delta x} \sim \Delta x^2$
- $\#$ (meshpoints) $\sim \Delta x^d$, $\#$ (timesteps) $\sim \Delta t \approx \Delta x^2$,
- Compute: $\mathcal{C} \sim \frac{1}{\Delta x^{d+2}}$
- Computational Complexity: $\mathcal{C} \sim \left(\frac{1}{E}\right)^{\frac{d+2}{2}}$
- Compute grows exponentially with dimension

Issues with Numerical Simulations

- High-quality Grid Generation:



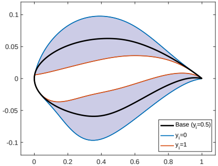
- Multiscale, MultiPhysics problems.
- High-Dimensional Problems

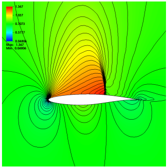
PDEs in high dimensions: $d \geq 4$

- Boltzmann Equation ($d = 7$)
- Radiative Transfer Equation ($d \geq 5$)
- Computational Finance: Black-Scholes ($d >> 1$)
- Computatational Chemistry: Schrödinger ($d >> 1$).

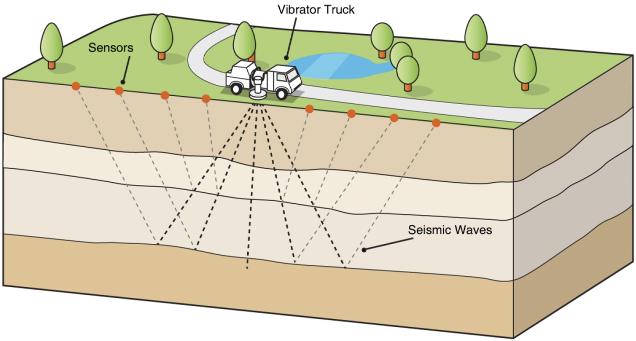
Many Query Problems: Optimal Design

- Example: Flow past airfoils (Euler or Navier-Stokes)
- Observables: Lift, Drag
- Uncertain parameters: Incident Mach Number, Angle of Attack, Pressure, Shape defects.
- Design parameters: Shape via Hicks-Henne functions.

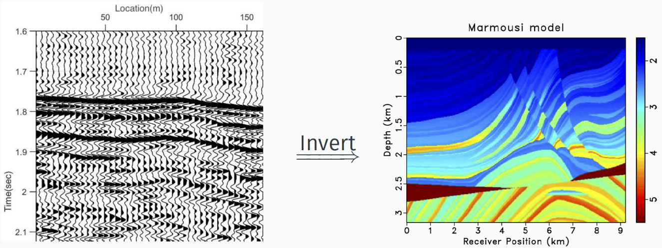




Many query Problems: Inverse Problem



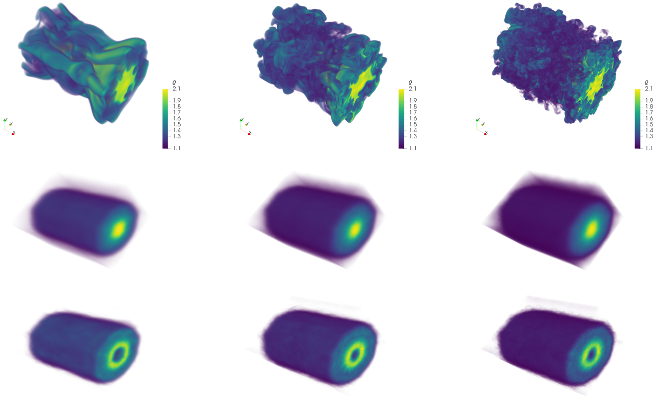
Inverse Problem for Wave Equation



Waveform measurements from receivers at the surface

Subsurface properties (i.e., wave velocity or material density)

Many Query Problems: UQ



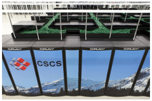
Sample

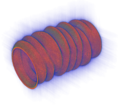
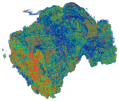
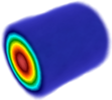
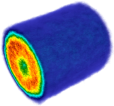
Mean

Var

256^3 512^3 1024^3

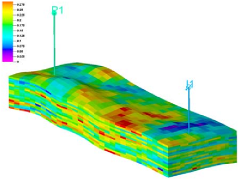
A little secret: Computational cost

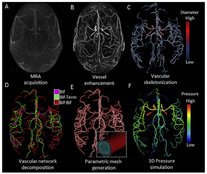


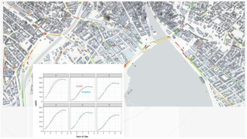





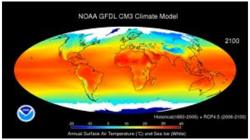
- Ensemble simulation of Navier-Stokes at 1024^3:
  - With Azeban on Piz Daint.
  - Single Sample: 94 node hours.
  - Ensemble simulation: 96256 node hours
  - Cost: Approx $560K$ CHF.

What about Missing Physics?









- The physics is incomplete/unknown and it is not just Unknown Parameters !!

Issue with traditional PDE solvers

- Computational Cost (Many Query Problems)
- Feasibility (High-dimensional Problems)
- Systems with Incomplete Physics but lots of data.
- Most of the rest of the course deals with finding alternative strategies based on deep learning.