Challenges in SciML

* Limited number of real world observations to train models

* Samples have sparsity

* Almost all real world systems are <u>spatio-temporal</u>

  ↳ machine learning is really bad at this.

* Real systems are $\boxed{\text{chaotic}}$

  ↳ we will come back to this

* Real systems are <u>multiscale</u> with no scale separation

Dynamical Systems:

Almost entirely, we would work on dynamical systems that model the real world.

Generally any dynamical system can be written as:

$$\frac{d\bar{u}}{dt} = \boxed{F}(\bar{u}, p)$$

→ * non linear complicated function.

→ * may be known, unknown, or partially known

F : Nonlinear function of x, y, z usually composed of $\nabla \bar{u}$, $\nabla^2 \bar{u}$, $\bar{u} \cdot \nabla \bar{u}$, ...

$\bar{u}$: A very large vector. For example, in Earth system modeling,

$$\bar{u} = \begin{bmatrix} \text{velocity} \\ \text{temperature} \\ \text{pressure} \\ \text{humidity} \\ \vdots \end{bmatrix}$$

p : This is a parameter that determines the distribution of u.

What's the current challenge in scientific computing?

Solving $\frac{d\bar{u}}{dt} = f(\bar{u})$ [given, p]

Since, $\bar{u} \rightarrow \bar{u}(\bar{x}, t)$ is a spatio-temporal vector

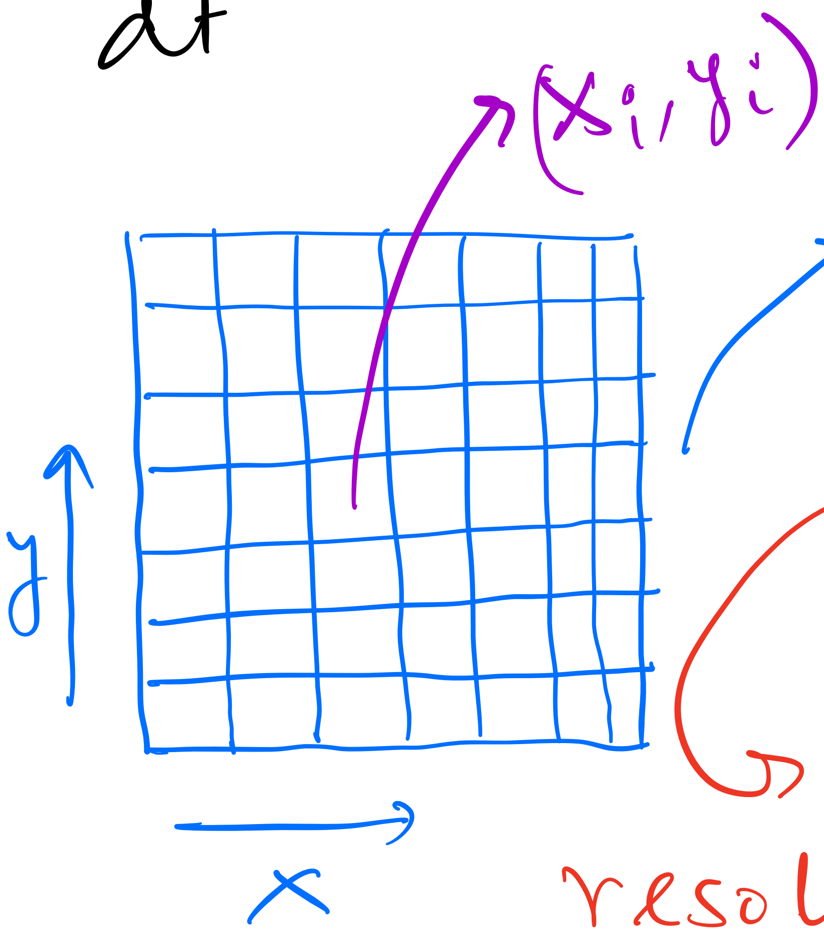
u needs to be discretized on:

* high resolution $\bar{x}$- grid.
* high resolution t - grid.

Consider a 2D-plane on which we are solving

$$\frac{d\bar{u}}{dt} = F(\bar{u})$$

Here $u \to u(x, y, t)$



We solve

$$\boxed{\frac{d\, u(x_i, y_i, t)}{dt} = F(u(x_i, y_i, t))}$$

$\to$ more the resolution, more the number of equation

Typically, a large system, requires, $10^6 - 10^8$ grid points in total for accurate spatial representation, for each variable

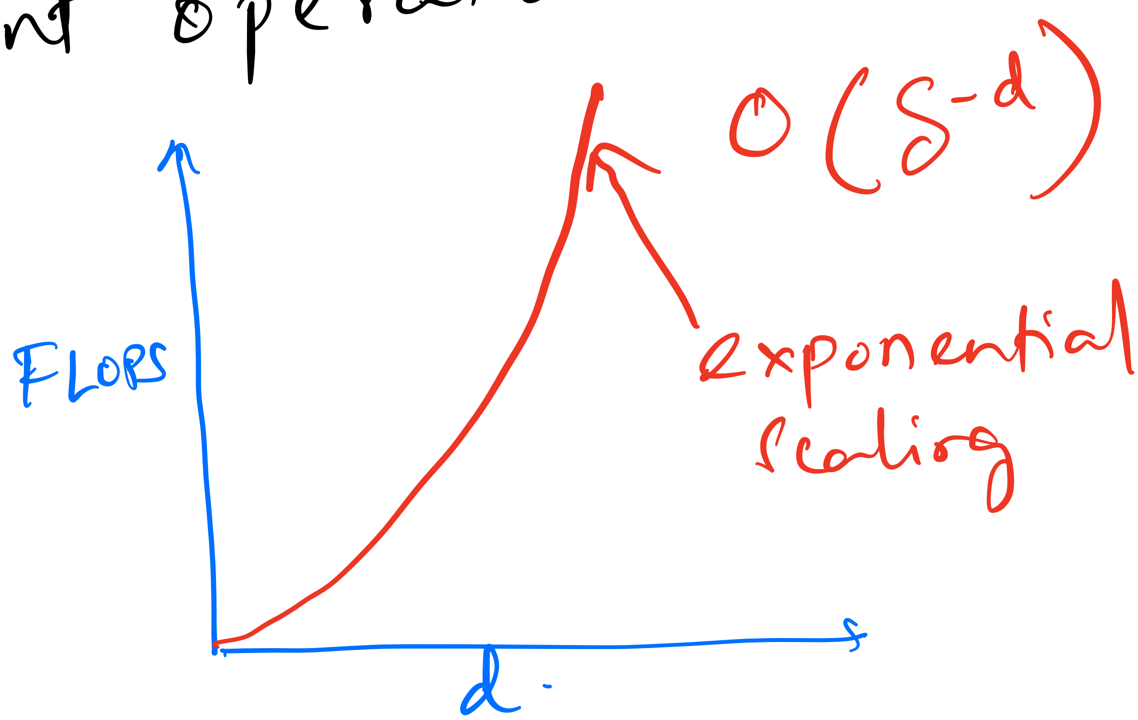
In SciML, we crudely define dimensionality of a system as:

$$\boxed{d = \text{number of variables} \times \text{resolution of each variable.}}$$

Intractability in scaling

In scientif computing, if we want to resolve a PDE with dimension d at an accuracy of $\underbrace{(1e-16)}_{\delta}$ [machine precision]

we need $O(\delta^{-d})$ floating point operations.



What kind of problems would we be dealing with?

(1) Autoregressive emulation or integration

$$\frac{d\bar{u}}{dt} = F(\bar{u})$$

Generally in numerical modeling:

$$\bar{u}(t+\Delta t) = \bar{u}(t) + \int_{t}^{t+\Delta t} F(\bar{u})\,dt$$

we come back to this

The goal of an autoregressive model is as follows:

Given historical record of

$\bar{u}(t_0)$, $\bar{u}(t_1)$ .... $\bar{u}(t_n)$

Design a model, M, with parameters, θ, $M(\theta)$, such that:

$$\bar{u}(t_i) \approx M(\theta, \bar{u}_{t_{i-1}})$$

$$\forall\, i \in \{0, 1, \ldots\ldots n\}.$$

The expectation of this model, $M(\theta)$, is to be able to predict the future:

$$\bar{u}_{t_k} = M\left(\theta^*, \bar{u}_{t_{k-1}}\right)$$

$$\forall\, k \in \{n, n+1 \ldots\}$$

$\bar{u}_{t_k}$ should match the $\bar{u}_{t_k}$ obtained from the physical model.

How do we find the optimal parameters, θ^* ?

For that, we define something called Empirical Risk. ($L(\theta)$) a.k.a. loss function.

$$\bar{u}(t_i) \approx M(\theta, \bar{u}_{t_{i-1}})$$

$$\forall\ i \in \{0, 1, \ldots n\}.$$

$$L(\theta) = \mathbb{E}_i \left[\left(\bar{u}_{t_i} - M(\theta, \bar{u}_{t_{i-1}}) \right)^2 \right]$$

$$L(\theta) = \sum_{i=0}^{n} \left(\bar{u}_{t_i} - M(\theta, \bar{u}_{t_{i-1}}) \right)^2$$

We minimize $L(\theta)$ to find optimal parameters, θ^*

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \; L(\theta),$$

↳ once we find that, we are done!

Throughout the course, we will learn how to design M.

② Estimating parameters of an unknown system (discovering physics)

Here, we have an unknown dynamical system;

$$\frac{d\bar{u}}{dt} = f(\bar{u})$$

unknown

But we have, noisy and/or sparse observations:

$$\tilde{u}(t_0),\ \tilde{u}(t_1)\ \ldots\ \tilde{u}(t_n)$$

Here we need to have an intuition about the structure of $\boxed{F}$

We assume a $\boxed{\text{very expressive}}$ form of F:

$$F = c_1 f_1 + c_2 f_2 \cdots c_k f_k$$

Important note: This is linear in c_i not in $\tilde{u}$

For example:

$$f_1 = \frac{\partial \tilde{u}}{\partial x}, \quad f_2 = \frac{\partial^2 \tilde{u}}{\partial x^2}$$

$$f_3 = \boxed{\tilde{u}\frac{\partial \tilde{u}}{\partial x}} \rightarrow \text{nonlinear}$$

Then, again, we define an empirical risk.

$$L(c) = \mathbb{E}_i \left[\left(\frac{\partial \tilde{u}_i}{\partial t} - \sum c_k f_k \right)^2 \right]$$

$$c_k^* = \operatorname*{argmin}_c L(c).$$

At the end, you "discover" the functional form of F.

Note: * The optimization of $L(c)$ is tricky, if $\tilde{u}$ is noisy and sparse.

* Sometimes, we want to find the simplest possible form of F $\rightarrow$ sparsity promoting optimization. Here, we want a sparse $\{c_k\}$ sequence.

(3) Online and Offline model error correction

Here, we assume that we know only part of the dynamical system

$$\frac{d\bar{u}}{dt} = \underbrace{H(\bar{u})}_{\text{known}} + \underbrace{G(\bar{u})}_{\text{unknown}}$$

We have noisy and sparse observations, $\bar{u}(t_0) \dots \bar{u}(t_n)$

Again, like 1, the objective is to predict the future $\bar{u}(t_{n+1})$, ... $\bar{u}(t_{n+2})$ .....

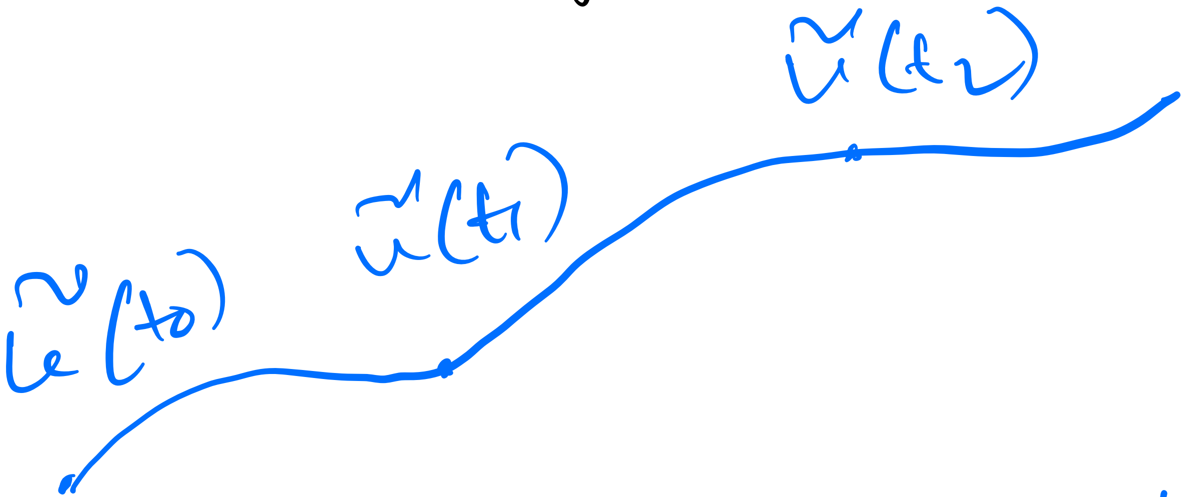
There are two approaches to doing this;

Offline mode training

or both togethere

online mode training

First step: Estimate a model for $G(n)$



-- first compute $\dfrac{\partial \tilde{u}(t)}{\partial t}$

-- 2nd compute $F(\tilde{u}(t))$

-- 3rd compute $\dfrac{\partial \tilde{u}}{\partial t} - F(\tilde{u}(t))$

Page 19

-- Now, we design a model, $M(\theta, \tilde{u}(t_i))$ in such a way that:

$$\overbrace{\boxed{\frac{\partial \tilde{u}(t_i)}{\partial t} - F(\tilde{u}(t_i))}}^{e(t_i)} \approx M(\theta, \tilde{u}(t_i))$$

$$\forall\, i \in \{0, 1, 2 \ldots n)$$

2nd step: Define the empirical risk.

$$L(\theta) = \frac{\sum_{i=0}^{n} \| e(t_i) - M(\theta, \tilde{u}(t_i)) \|}{n}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \; L(\theta)$$

Note that in prediction (inference) mode:

We solve the PDE/ODE

$$\frac{\partial u}{\partial t} = F(u) + M(\theta^*, u)$$

as you evolve this model M & F interact with each other and may go unstable.

An example is subgrid-scale models in fluid dynamics

Online training:

Step 1: Define a model $\mu(\theta)$ with random θ.

So: $$\frac{du}{dt} = F(u) + \mu(\theta, u)$$

Step 2

Write the discrete form of the equation:

$$u^{p}(t_{i+1}) = \tilde{u}(t_i) + \int_{t_i}^{t_{i+1}} F(\tilde{u})\,dt + \int_{t_i}^{t_{i+1}} \mu(\theta, \tilde{u})\,dt$$

predicted time step

Step 3: Define the empirical risk in $\bar{u}$

$$L(\theta) = \mathbb{E}_i \left[\| u^p(t_{i+1}) - \tilde{u}(t_{i+1}) \|_2 \right]$$

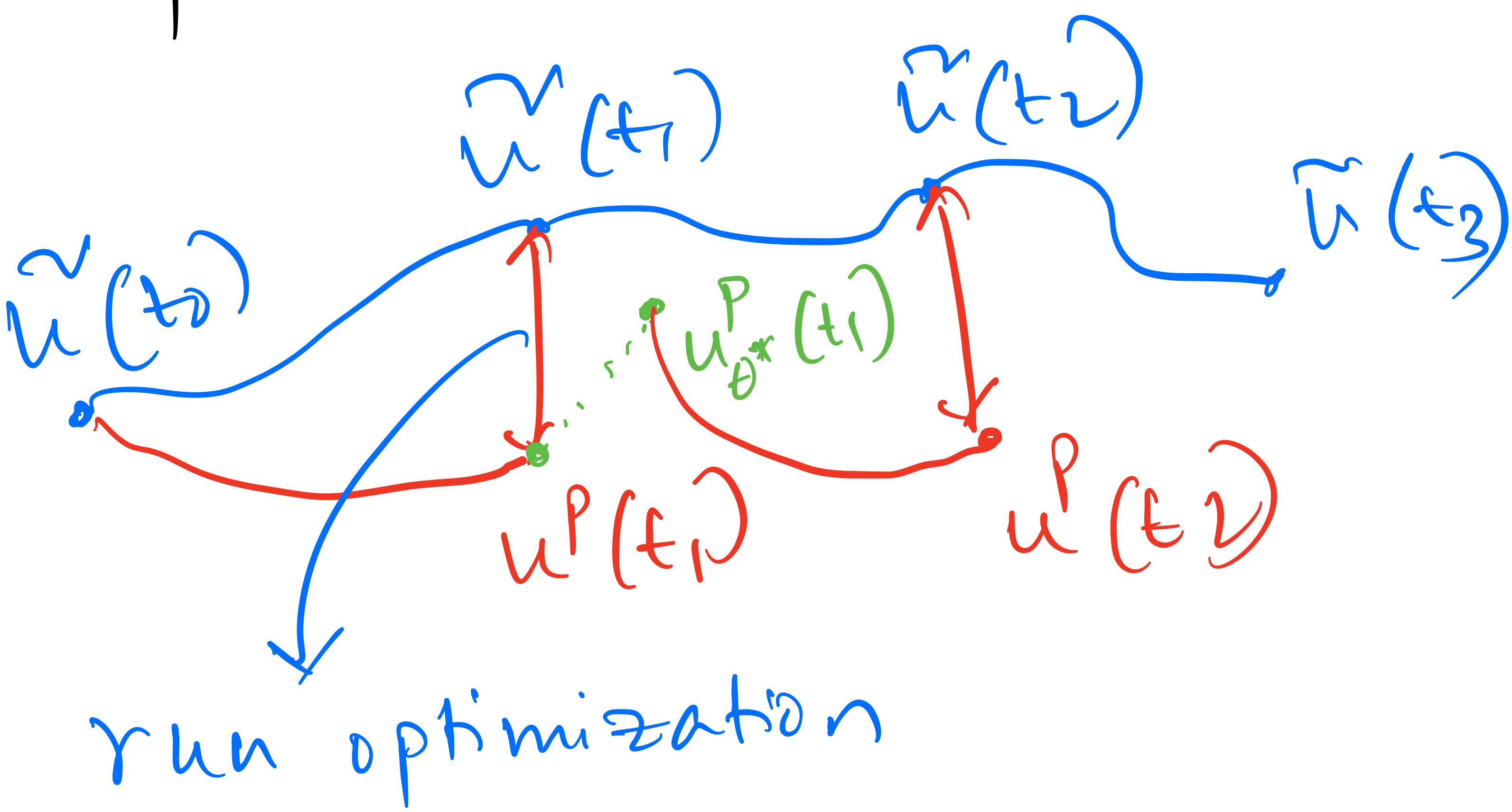
$$\theta^* = \arg\min_{\theta} L(\theta)$$

This is a complicated optimization problem are requires a differentiable solver.

Page 25

The online model is a sequential training problem

We do not have to take expectation over all 'i'



Refresher on numerical integration in time.

1st order Euler's method.

$$\frac{du}{dt} = F(u)$$

$$\boxed{u(t+\Delta t) = u(t) + F(u)\Delta t}$$

1st order simplest scheme.

4th order Runge Kutta method:

$$u(t+\Delta t) = u(t) + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right]$$

$$k_1 = F(u(t))$$

$$k_2 = F\left[u(t) + \frac{k_1}{2}\right]$$

$$k_3 = F\left[u(t) + \frac{k_2}{2}\right]$$

$$k_4 = F\left[u(t) + k_3\right]$$