Hypothesis for Neural Residual PDEs

September 11, 2019

Define a dataset $\mathcal{D} = \{\phi_i, \psi_i, \mathbf{x}_{i,1:T}\}_{i=1}^N$ of boundary conditions ϕ, initial conditions ψ, and a sequence of observations $\mathbf{x}_{1:T}$. We can define an observation in two ways: first, $\mathbf{x}_t = (\mathbf{u}_t, \mathbf{v}_t, \mathbf{p}_t)$ contains two dimensions of momentum and pressure; or second, $\mathbf{x}_t = (\lambda_t^{\mathbf{u}}, \lambda_t^{\mathbf{v}}, \lambda_t^{\mathbf{p}})$, a vector of spectral coefficients for momentum and pressure functions.

Then, the function we want to learn consists of two components — a base function $f \in \mathcal{F}$ where $f_\theta : \mathcal{X} \to \mathcal{X}$ that takes an observation as input. This base function is responsible for learning the dynamics. Second, define a residual function $g_\theta : \{\phi, \psi\} \to \mathcal{F}$, tranforming boundary and initial conditions to a function. The base function is shared over all entries in the dataset — we consider the following objective:

$$\min \mathbb{E}_{\phi,\psi,\mathbf{x}_{1:T}\sim p_{\mathcal{D}}} \left[\sum_{t=1}^{T} \mathbf{x}_t - (g_\theta(\phi, \psi) + f_\theta(\mathbf{x}_{t-1})) \right] \tag{1}$$

Note that g_θ is kind of like a hypernetwork. Next question is what does f_θ look like (how does it relate to PDEs/ODEs)?

We are inspired by the spectral approach and want to learn the coefficeints using neural networks. However, unlike Chorin's method, we do not discretize in time. Define the following:

$$u(x,y,t) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \lambda_{k,l}(t) T_k(x) T_l(y) \approx \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \lambda_{k,l}(t) T_k(x) T_l(y) = u_\lambda(x,y,t) \tag{2}$$

$$v(x,y,t) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \omega_{k,l}(t) T_k(x) T_l(y) \approx \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \omega_{k,l}(t) T_k(x) T_l(y) = v_\omega(x,y,t) \tag{3}$$

$$p(x,y,t) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \gamma_{k,l}(t) T_k(x) T_l(y) \approx \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \gamma_{k,l}(t) T_k(x) T_l(y) = p_\gamma(x,y,t) \tag{4}$$

Note that differentiation is very simple in this paradigm:

$$\frac{\partial}{\partial x} u_\lambda(x,y,t) = \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \lambda_{k,l}(t) T_l(y) \frac{\partial}{\partial x} T_k(x) \tag{5}$$

$$\frac{\partial}{\partial y} u_\lambda(x,y,t) = \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \lambda_{k,l}(t) T_k(x) \frac{\partial}{\partial y} T_l(y) \tag{6}$$

So the Navier Stokes equations are a function of only $(\lambda_{k,l}(t)), (\omega_{k,l}(t)), (\gamma_{k,l}(t))$. Each of these are only functions of time, meaning we can treat them as ODEs. Then we can try the following objective,

$$\mathcal{L} = \|\mathbf{u}_\mathcal{D} - \mathbf{u}_\lambda\|_2 + \|\mathbf{v}_\mathcal{D} - \mathbf{v}_\omega\|_2 + \|\mathbf{p}_\mathcal{D} - \mathbf{p}_\gamma\|_2 \tag{7}$$

$$\mathcal{L} = \sum_{x,y,t} |u_\mathcal{D}(x,y,t) - u_\lambda(x,y,t)|^2 + |v_\mathcal{D}(x,y,t) - v_\omega(x,y,t)|^2 + |p_\mathcal{D}(x,y,t) - p_\gamma(x,y,t)|^2 \tag{8}$$

Effectively, we decompose the learning problem from a PDE to an ODE by use spectral methods to eliminate space.