PhD Open Days

Physics-Informed LSTM Networks for Transient 3D Incompressible Navier-Stokes: A Beltrami-Flow Benchmark

PhD in Electrical and Computer Engineering

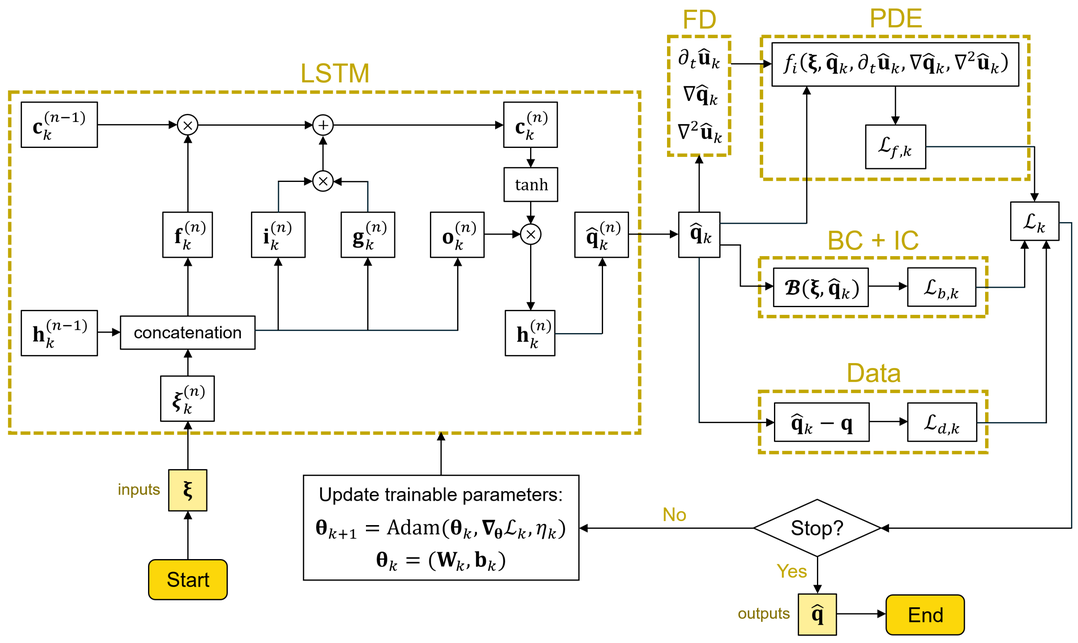
Duarte Miguel Fradinho Fernandes Malveiro (duarte.f.malveiro@tecnico.ulisboa.pt)

Summary

Physics-Informed Long Short-Term Memory (PI-LSTM) networks are employed to predict the Beltrami Flow solutions. The models are trained on 100 different training sets, using a Nvidia Tesla V100S 32 GB GPU. At the end of six different training times, the trained models are evaluated on an unseen test set.

Problem Formulation and Implementation

An LSTM (Long Short-Term Memory) is a recurrent neural network designed to model sequential data. Its recurrent nature and gated structures allow information to be retained and updated across time steps [1]. Combined with physics-informed training, a PI-LSTM can model time-dependent physical phenomena.



PI-LSTM scheme to find predictions for the Beltrami flow, illustrating the sequence performed at each iteration k.

- The velocity field $\hat{\mathbf{u}} = (\hat{u}, \hat{v}, \hat{w})$ and pressure $\hat{p}$, $\hat{\mathbf{q}} = (\hat{\mathbf{u}}, \hat{p})$, are predicted using an **LSTM** network, given space $\mathbf{x} = (x, y, z)$ and time t inputs, $\boldsymbol{\xi} = (\mathbf{x}, t)$, inside a domain Ω and at its border Γ.
- Space and time derivatives are computed through finite differences (**FD**).
- The goal is to minimize the total loss function: $\mathcal{L} = \mathcal{L}_f + \mathcal{L}_b + \mathcal{L}_d$.

Residual loss

$$\mathcal{L}_f = \frac{1}{N_f}\sum_{i=1}^{4}\sum_{j=1}^{N_f}\lambda_i^{(f)}\left(f_i\left(\boldsymbol{\xi}_f^{(j)}\right)\right)^2$$

BC and IC loss

$$\mathcal{L}_b = \frac{1}{N_b}\sum_{i=1}^{3}\sum_{j=1}^{\mathrm{N}_b}\lambda_i^{(b)}\left(u_i\left(\boldsymbol{\xi}_b^{(j)}\right) - \hat{u}_i\left(\boldsymbol{\xi}_b^{(j)}\right)\right)^2,\ b \equiv \text{BC or IC}$$

Data loss

$$\mathcal{L}_d = \frac{1}{N_d}\sum_{i=1}^{4}\sum_{j=1}^{N_d}\lambda_i^{(\mathrm{d})}\left(q_i\left(\boldsymbol{\xi}_d^{(j)}\right) - \hat{q}_i\left(\boldsymbol{\xi}_d^{(j)}\right)\right)^2$$

Residuals of i-th Navier-Stokes (NS) equations (**PDE**) at each point j, $f_i\left(\boldsymbol{\xi}_f^{(j)}\right)$:

$$f_i\left(\boldsymbol{\xi}_f^{(j)}\right) = \begin{cases} \nabla\cdot\hat{\mathbf{u}}, & i = 1 \\ \dfrac{\partial\hat{\mathbf{u}}}{\partial t} + (\hat{\mathbf{u}}\cdot\nabla)\hat{\mathbf{u}} + \nabla\hat{p} - \dfrac{1}{\mathrm{Re}}\nabla^2\hat{\mathbf{u}}, & i = \{2,3,4\} \end{cases}$$

Analytical solutions for the Beltrami flow [2], a particular case of the NS equations (the non-linear term $(\hat{\mathbf{u}}\cdot\nabla)\hat{\mathbf{u}}$ becomes linear), appear in the boundary condition (BC initial condition (BC) and data losses, as the reference velocities u_i, being $\mathbf{u} = (u, v, w) \equiv (u_1, u_2, u_3)$. The Reynolds number is $\mathrm{Re} = 1$.

Adaptive Loss Weights

Total Loss Function: $\mathcal{L} = \mathcal{L}_1(\boldsymbol{\theta}) + \sum_{l=2}^{N_\mathcal{L}}\lambda_i\tilde{\mathcal{L}}_i(\boldsymbol{\theta})$

First weight estimation and final weights:
(performed every δ_1 training iterations)

$$\begin{cases} \tilde{\lambda}_{i,k+1} = \dfrac{\overline{\left|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{1,k+1}\right|}}{\overline{\left|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{i,k+1}\right|}} \\ \lambda_{i,k+1} = (1-\alpha)\lambda_{i,\mathrm{k}} + \alpha\tilde{\lambda}_{i,k+1} \end{cases}$$

Results

- 3 recurrent layers; 100 features per layer in the hidden state h
- Learning rate: $\eta = \eta_0 \mathrm{C}^k$; $\mathrm{C} = 0.1^{1/\delta_2}$ ($\eta_0 = 0.01$; $\delta_2 = 10000$ iterations)
- Adaptive Loss Weights: $\alpha = 0.5$; $\delta_1 = 100$ iterations
- Space domain, time domain and time step: $\Omega_\mathrm{s} = [-1, 1]^3$; $\Omega_\mathrm{t} = [0, 1]$; $dt = 0.01$
- 100 different training sets (per time step): 200 randomly chosen points in the interior of Ω_s and 50 points per boundary face
- One test set (per time step): 64^3 equally spaced points, uniformly distributed throughout the space domain

Accuracy

100-run average L2 relative error at the component $q = \{u, v, w, p\}$, $\overline{L_{2,q}}(t)$

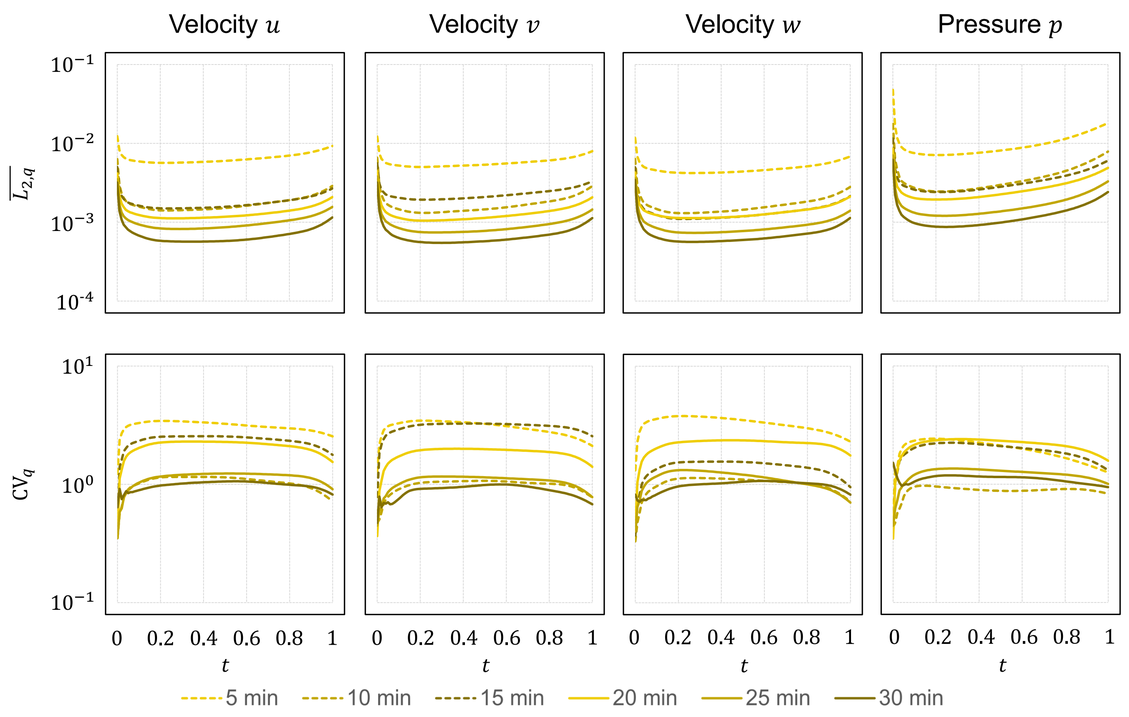
Consistency

Coefficient of variation, $\mathrm{CV}_q(t) = \frac{\sigma_q(t)}{\overline{L_{2,q}}(t)}$

Evaluated in the test set, at the end of six training time checkpoints (t_c): 5, 10, 15, 20, 25 and 30 minutes.

Standard deviation of $L_{2,q}(t, r)$, run r, in relation to its N-run average:

$$\sigma_q(t) = \sqrt{\frac{1}{N}\sum_{\mathrm{r}=1}^{N}(L_{2,q}(t,r) - \overline{L_{2,q}}(t))^2}$$



- The velocity field is well predicted even with just 10 minutes of training, with $\overline{L_{2,q}} \approx 10^{-3}$. With $t_c = 30$, $\overline{L_{2,q}} = 5.5\times10^{-4}$ to 7×10^{-4} in most time steps.
- Larger training times generally lead to higher prediction accuracy. However, such trend is not always verified in consistency.

References:

[1] Sak, H., Senior, A., & Beaufays, F. (2014). *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition*. arXiv. https://arxiv.org/abs/1402.1128

[2] C. Ross Ethier, D. A. Steinman. Exact fully 3D Navier–Stokes solutions for benchmarking. *Int. Journal for Numerical Methods in Fluids*, pp. 369-375, 1994.

TÉCNICO UNIVERSIDADE DE LISBOA

Luís Miguel Silveira, Ruxandra Barbulescu, Nelson Marques

PhD Programme in Electrical and Computer Engineering

phdopendays.tecnico.ulisboa.pt