1 DD-GAN group

1.1 Domain Decomposition Generative Adveserial Networks (DD-GANs) for time series predictions

State of the art AI models (e.g. RNN, LSTM) can learn time series data well, but have difficulty in predicting in time beyond the training data. This project explores the use of Generative Adversarial Networks to do this. Although most often used in the field of image classification and generation, GANs have recently been used to predict in time and show great promise for this new application. This method will be developed with the aim to apply it to challenging test cases with high aspect ratio domains (i.e. long and thin) hence the use of domain decomposition. The student will use CFD data and build AI models in Python and TensorFlow or PyTorch.

Project 1 (Jon Tómasson): will first combine POD and GAN for FPC (currently some code to do this). The next stage is to apply DD to this (part of the code exists for this too). The third stage is to apply this to slug flow.
Project 2 (Zef Wolffs): Improve the dimensionality reduction step (compression) by experimenting with SVD-AE, classical convolutional AE (CAE) and an adversarial autoencoder (AAE) [1] on FPC (without and with DD). Then apply to slug flow.
Project 3 (Tianyi Zhao): Compare different methods for time prediction on the FPC test case and perhaps then slug flow:

(a) a multi-layer perceptron (f) trained to predict the next time level $\boldsymbol{\alpha}^{n+1} = f(\boldsymbol{\alpha}^n)$;
(b) a multi-layer perceptron trained to predict the next increment [3] $\boldsymbol{\alpha}^{n+1} = \Delta t\, f(\boldsymbol{\alpha}^n) + \boldsymbol{\alpha}^n$;
(c) a GAN to predict the time derivatives $\boldsymbol{\alpha}^{n+1} = \Delta t\, G(\boldsymbol{z}^n) + \boldsymbol{\alpha}^n$;

Project 4 (Stella Wang): Add a physics-informed (PI) term to the GAN's loss function [4, 5] (a) using information all the points (nodes); (b) using randomly selected points; (c) and using points based on DEIM. First FPC will be used (without and with DD). If time, apply to slug flow.

project	compression	time prediction	DD	test case
1	POD	GAN		FPC
	POD	GAN	DD	FPC
	POD	GAN	DD	slug flow
2	SVD-AE/2DCAE/AAE	—	DD	FPC
	SVD-AE/2DCAE/AAE	GAN	DD	FPC
	SVD-AE/2DCAE/AAE	—	DD	slug flow
	SVD-AE/2DCAE/AAE	GAN	DD	slug flow
3	POD	derivative or not / MLP / GAN		FPC
	POD	derivative or not / MLP / GAN	DD	FPC
	POD	derivative or not / MLP / GAN	DD	slug flow
4	POD	GAN + PI loss		FPC
	POD	GAN + PI loss	DD	FPC

1.2 Building a detailed city flow model using Domain Decomposition and AI for compression and prediction

Fluid flow past buildings has a number of typical characteristics, such as eddy structures that occur near all buildings. Similarly, the influence of neighbouring buildings on the air flow is repeated across a city. Thus, it may be possible to produce a generic model of a small (e.g. 50 m by 50 m) area and link this to similar models of surrounding regions to build up a larger scale model.

A test case will be developed in 2D in which buildings are being represented by rectangles and a simulation will be run (in Fluidity or IC-FERST) to obtain the flow around these buildings. Data from within the simulation domain will be sampled many times and compressed, by an SVD or autoencoder. The flow variables can then be represented in a reduced space defined by the basis functions (from SVD) or encoder (from the autoencoder). Another neural network (e.g. a GAN) will be trained to predict the flow around the buildings in one region, given some information about the flow in neighbouring regions. All the regions being modelled are swept through iteratively until convergence is reached. This project will be coded in python and the AI models built in PyTorch or TensorFlow.

These two projects will build on the framework used in project 1 above. Project 5 will look at compressing information associated with the buildings using a 2D standard convolutional AE (CAE) for steady flow (low Reynolds numbers) and unsteady flow (moderate Reynolds number) and compare this with POD. Project 6 will use POD for compression and develop the time prediction for steady and then unsteady flow past buildings. The domain decomposition used in this project will be slightly more advanced than for project 1.

project		tasks
5 & 6		sampling / domain decomposition approach
5	Hanna	(POD), classical 2D CAE
6	Xiangqi	(POD), adversarial AE
5	Hanna	(MLP/)GAN - time level
6	Xiangqi	(MLP/)GAN - time derivative

Hanna and Xiangqi to think about sampling the vtu files to form the snapshots / data Hanna to apply (POD) and 2D classical CAE to learn the locations of the buildings Xiangqi to apply (POD) and adversarial AE to learn the locations of the buildings Hanna to apply MLP/GAN to predict the flow patterns Xiangqi to apply MLP/GAN (training for the derivative or increment) to predict the flow patterns

References

[1] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey. Adversarial Autoencoders. arxiv.org/abs/1511.05644 (2016).

[2] D. Ulyanov, A. Vedaldi and V. Lempitsky. It Takes (Only) Two: Adversarial Generator-Encoder Networks. In: AAAI Publications, Thirty-Second AAAI Conference on Artificial Intelligence, 1250– (2018). *(Version with appendix on arXiV: arxiv.org/pdf/1704.02304.pdf.)*

[3] F. Regazzoni, L. Dedè and A. Quarteroni. Machine learning for fast and reliable solution of time-dependent differential equations. *Journal of Computational Physics, 397* (2019) doi.org/10.1016/j.jcp.2019.07.050

[4] M. Raissi, P. Perdikaris and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378: 686–707*, (2019) doi.org/10.1016/j.jcp.2018.10.045

[5] M. Raissi, Z. Wang, M. S. Triantafyllou and G. Em. Karniadakis. Deep learning of vortex-induced vibrations.*Journal of Fluid Mechanics 861:119–137* (2019) doi.org/10.1017/jfm.2018.872.