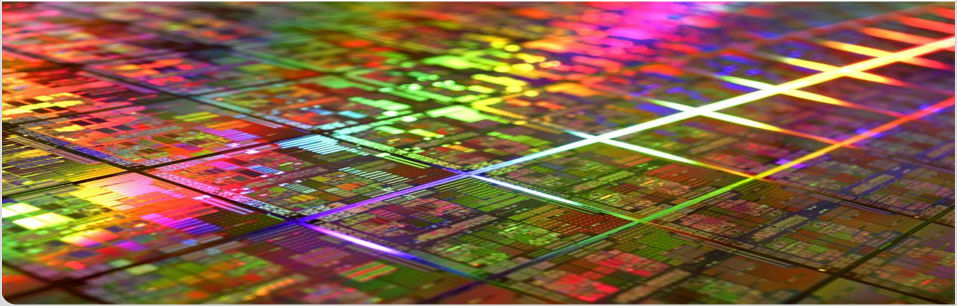
Towards Bringing Together Numerical Methods for Partial Differential Equation and Deep Neural Networks

Project presentation, Supervisor - Markus Hoffmann

Stanislav Arnaudov | 4. Juni, 2020

CHAIR FOR COMPUTER ARCHITECTURE AND PARALLEL PROCESSING



Introduction

Basic idea: Apply machine learning and image processing in computational fluid dynamics context

Introduction

Basic idea: Apply machine learning and image processing in computational fluid dynamics (CFD) context

- Machine learning through deep neural networks
- DNNs for image generation

Introduction

Basic idea: Apply machine learning and image processing in computational fluid dynamics (CFD) context

- Machine learning through deep neural networks
- DNNs for image generation

- Simulation of fluids through mathematical models
- Differential equations
- Specialized solvers for the equations

Project

- **What** we aimed to do?
- **Why** is our research being done?
- **How** were our goals achieved?

What

- Predict the flow around an object in a channel.

What

- Predict the flow around an object in a channel.

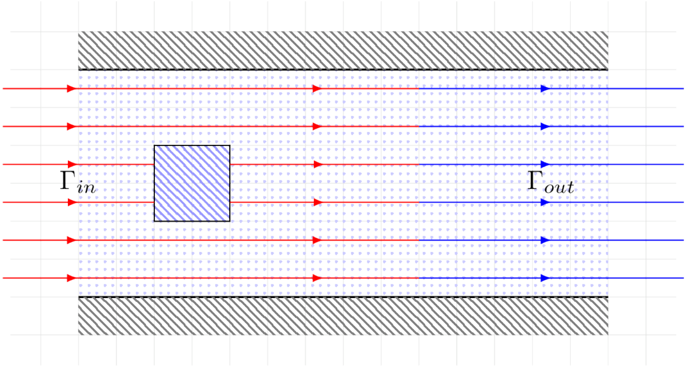
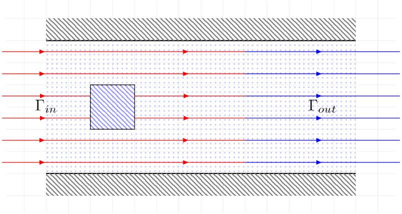


Figure: Simulation Setup

What

- Predict the flow around an object in a channel.



Parameters
for the whole simulation

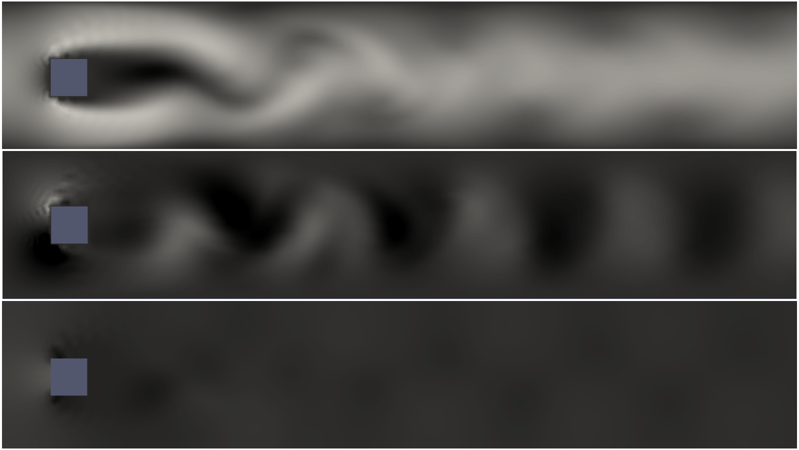
- Inflow speed of the fluid
- Viscosity and density of the fluid

Solutions
for each timestep

- Velocity field (x&y directions)
- Pressure field

What

- Predict the flow around an object in a channel.

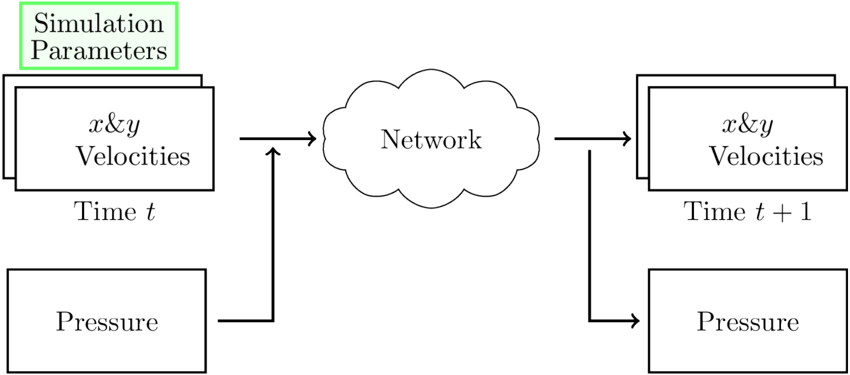


What

- Predict the flow around an object in a channel.
- Network Principle

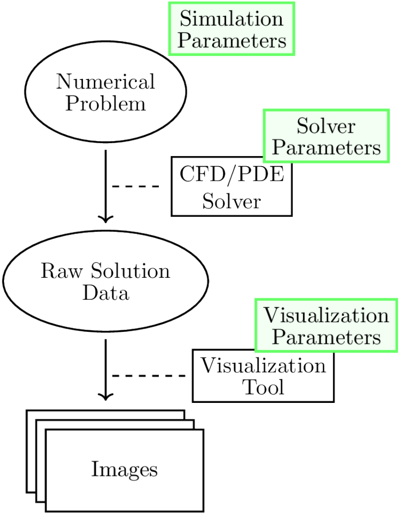
What

- Predict the flow around an object in a channel.
- Network Principle



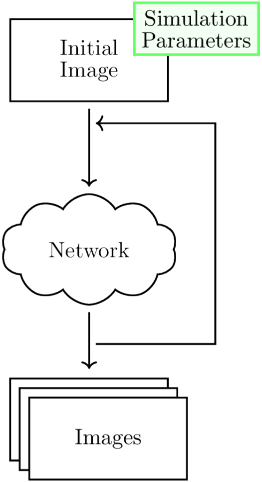
Why

Why



- Lots of parameters to tweak
- Complicated workflow
- Computationally expensive

Why



- Everything is learned during training
- DNNs are well established in image processing tasks
- Faster image generation

How

How

- Not a single holistic model
  - Constant model
  - Inflow speed model
  - Viscosity-density model

How

- Not a single holistic model
- Model variations
  - Usage of the pressure field
  - No usage of the pressure field

Data Generation

Data Generation

- Real simulation data gathering
  - Define large amount of simulations
  - Solving — HiFlow3[1]as numerical solver
  - Rendering — ParaView[2]as visualization toolkit

[1]Ayachit, Utkarsh, “The ParaView Guide: A Parallel Visualization Application, Kitware” , 2015, ISBN 978-1930934306

[2]Gawlok, S., Gerstner, P., Haupt, S., Heuveline, V., Kratzke, J., Lösel, P., Mang, K., Schmidtobreick, M., Schoch, N., Schween, N., Schwegler, J., Song, C. and Wlotzka, M., “HiFlow3 – Technical Report on Release 2.0”

Data Generation

- Real simulation data gathering
- Separate data sets for each model
  - Single simulation for the constant model
  - Multiple simulations for the other models

Data Generation

- Real simulation data gathering
- Separate data sets for each model
- Parameters for the simulations
  - Not chosen at random
  - *Reynolds number* in [90, 450]
  - Non-trivial simulations

Data Generation

- Real simulation data gathering
- Separate data sets for each model
- Parameters for the simulations

Figure: X velocity

Figure: Y velocity

Data Generation

- Real simulation data gathering
- Separate data sets for each model
- Parameters for the simulations
- Test train splits
  - 80/20 split for all data sets
  - No common Reynolds numbers between the simulations in the train- and test-sets

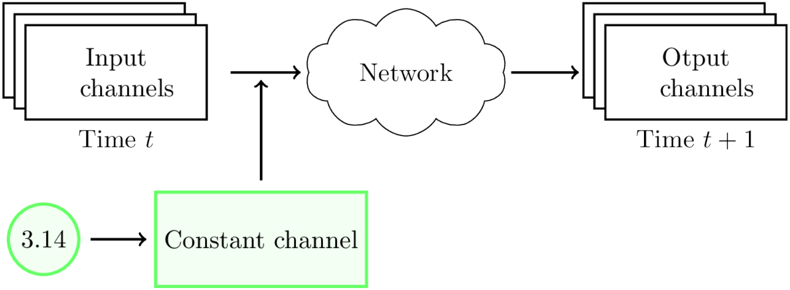
Networks Architectures

Networks Architectures

- Real numbers handling

Networks Architectures

- Real numbers handling



Networks Architectures

- Real numbers handling
- Approach based on Pix2Pix[3]
  - General image-to-image translation framework
  - Impressive results in recent years

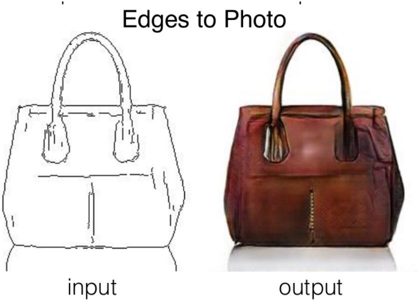


Figure: Example image from the pix2pix paper

[3]Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. "High-resolution image synthesis and semantic manipulation with conditional

Networks Architectures

- Real numbers handling
- Approach based on Pix2Pix
- Conditional GAN (cGAN)

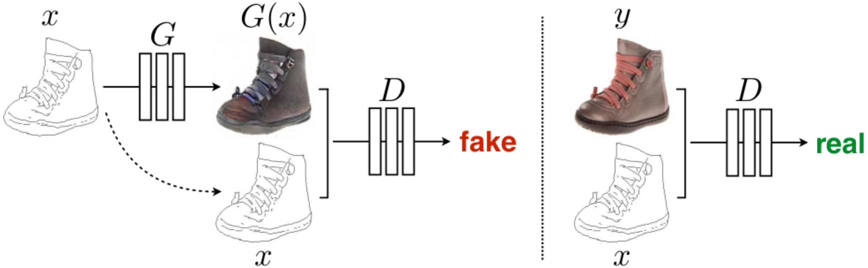


Figure: Training a cGAN

Networks Architectures

- Real numbers handling
- Approach based on Pix2Pix
- Conditional GAN (cGAN)
- Generator Architecture
  - UNet

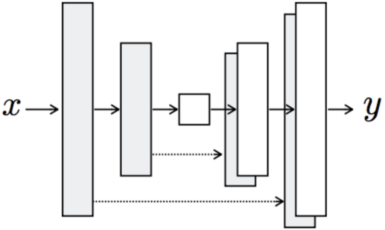


Figure: The UNet architecture

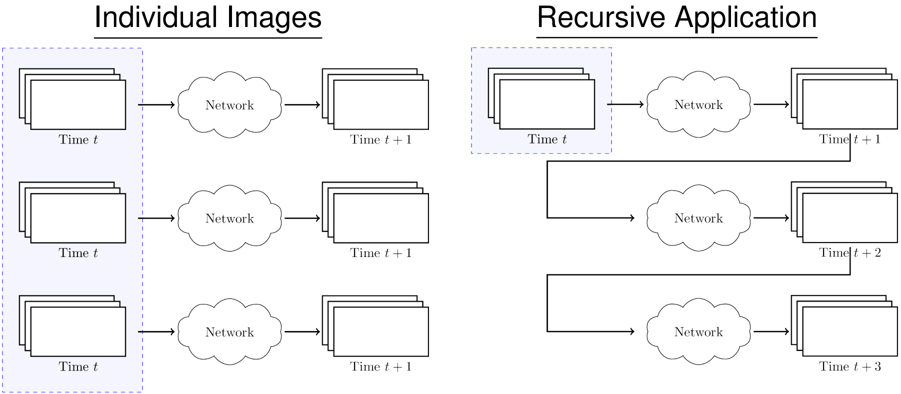
Networks Architectures

- Real numbers handling
- Approach based on Pix2Pix
- Conditional GAN (cGAN)
- Generator Architecture
- Discriminator Architecture
  - PatchGAN

Evaluation

Evaluation

- Evaluation Strategies



Evaluation

- Evaluation Strategies
- Results Views
  - Numerical point of view
  - Perceptual/Visual point of view

Evaluation

Numerical View

- Average difference
- Maximal difference

Evaluation

Numerical View

Model	Max diff.	Average diff.
Constant	14.56%	0.60%
Inflow speed	32.16%	0.55%
Viscosity-density	57.00%	13.85%

Table: All of the results are from models using the pressure field

Evaluation

Numerical View

Model	Max diff.	Average diff.
Constant	14.56%	0.60%
Inflow speed	32.16%	0.55%
Viscosity-density	57.00%	13.85%

Table: All of the results are from models using the pressure field

Evaluation

Numerical View

Figure: Visible pattern in a generated image

Evaluation

Perceptual view

- Correlation
- PSNR (Peak Signal to Noise Ration)

Evaluation

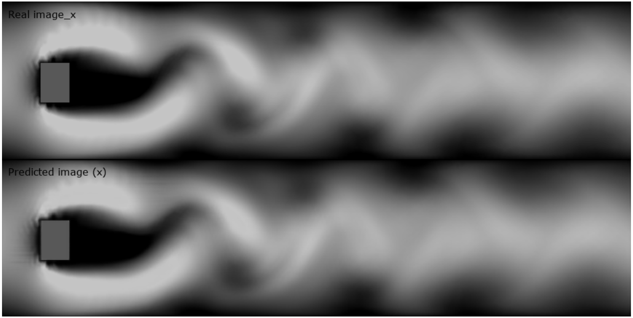
Perceptual view

- **Correlation**
- PSNR (Peak Signal to Noise Ration)

Evaluation

Perceptual view

- **Correlation**
- PSNR (Peak Signal to Noise Ration)



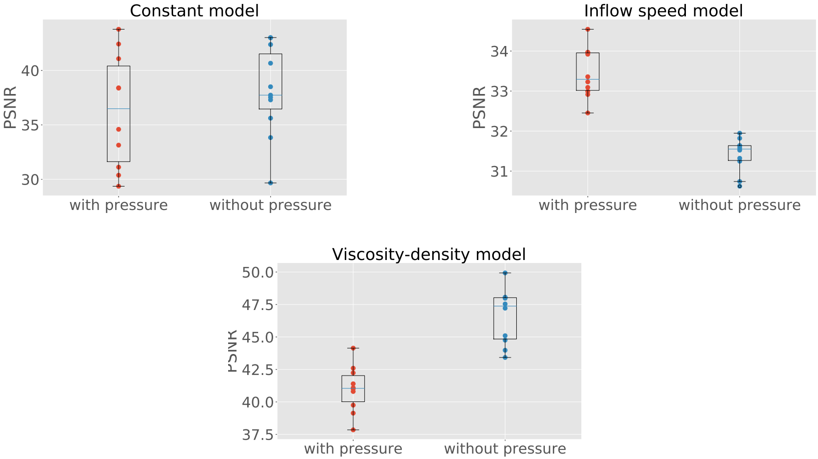
Evaluation

Perceptual view

- Correlation
- **PSNR (Peak Signal to Noise Ration)**

Evaluation

Perceptual view: *PSNR* — higher is better



Evaluation

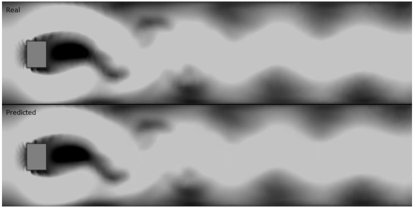


Figure: Inflow speed model

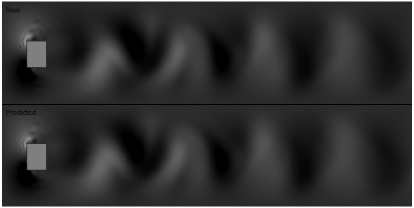


Figure: Fluid model

Evaluation

Perceptual view: *PSNR* — higher is better

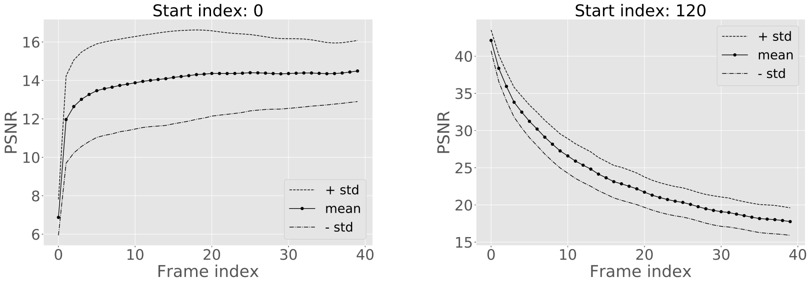


Figure: Inflow speed model recursive results

Evaluation

Perceptual view

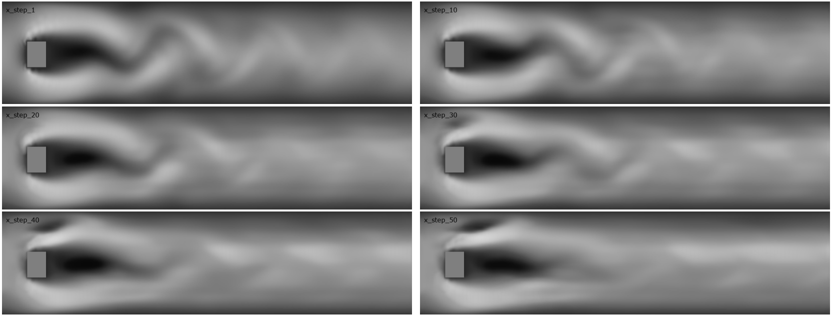


Figure: Predicted simulation from the viscosity-density model

Performance

Performance

Method	Single core	Multi-Core (12)	GPU
Our Networks	$\approx 600ms$	$\approx 100ms$	$4ms$
HiFlow3	$\approx 8000ms$	$\approx 1000ms$	–

Table: Time in milliseconds needed per simulation-step

Thank you for your attention.

Questions?