University of Stuttgart
Institute of Aerodynamics
and Gas Dynamics

Andrea Beck
David Flad
C.-D. Munz

Deep Neural Networks for Data-Driven Turbulence Models

@ICERM 2019: Scientific Machine Learning

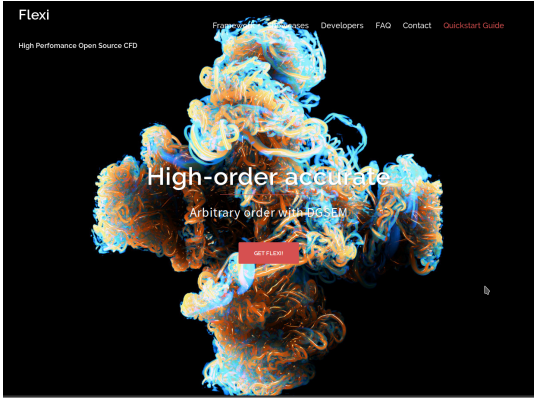
Outline

Introduction

1

Introduction

- Numerics Research Group @ IAG, University of Stuttgart, Germany
- Primary Focus: High Order Discontinuous Galerkin Methods
- OpenSource HPC solver for the compressible Navier-Stokes equations



www.flexi-project.org

DG-SEM in a nutshell

- Hyperbolic/parabolic conservation law , e.g. compressible Navier-Stokes Equations

$$U_t + \vec{\nabla} \cdot \vec{F}(U, \vec{\nabla} U) = 0$$

- Variational formulation and weak DG form per element for the equation system

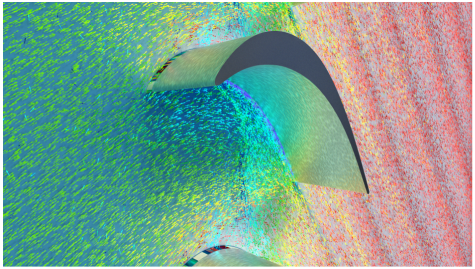
$$\langle J\, U_t, \psi \rangle_E + \left(\widetilde{f}^* \, \vec{n}_\xi, \psi \right)_{\partial E} - \left\langle \tilde{\vec{F}}, \nabla_\xi \psi \right\rangle_E = 0,$$

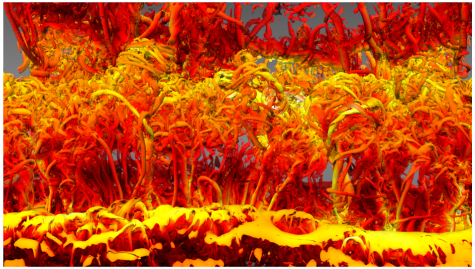
- Local tensor-product Lagrange polynomials, interpolation nodes equal to quadrature nodes
- Tensor-product structure in multi-D: line-by-line operations

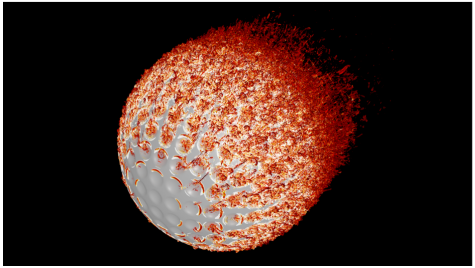
$$(U_{ij})_t + \frac{1}{J_{ij}} \left[\widetilde{f}^*(1, \eta_j) \hat{\psi}_i(1) - \widetilde{f}^*(-1, \eta_j) \hat{\psi}_i(-1) + \sum_{k=0}^{N} \hat{D}_{ik} \, \widetilde{F}_{kj} \right]$$
$$+ \underbrace{\frac{1}{J_{ij}} \left[\widetilde{g}^*(\xi_i, 1) \hat{\psi}_j(1) - \widetilde{g}^*(\xi_i, -1) \hat{\psi}_j(-1) + \sum_{k=0}^{N} \hat{D}_{jk} \, \widetilde{G}_{ik} \right]}_{\text{1D DGSEM Operator}} = 0$$

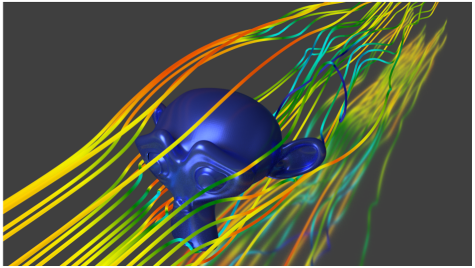
- BR1/2 lifting for viscous fluxes, Roe/LF/HLL-type inviscid fluxes, explicit in time by RK/ Legendre-Gauss or LGL-nodes

Applications: LES, moving meshes, acoustics, multiphase, UQ, particle-laden flows...









Machine Learning with Neural Networks

2

Rationale for Machine Learning

"It is very hard to write programs that solve problems like recognizing a three-dimensional object from a novel viewpoint in new lighting conditions in a cluttered scene.

- We don't know what program to write because we don't know how its done in our brain.
- Even if we had a good idea about how to do it, the program might be horrendously complicated."

Geoffrey Hinton, computer scientist and cognitive psychologist (h-index:140+)

Definitions and Concepts

An attempt at a definition:

Machine learning describes algorithms and techniques that progressively improve performance on a specific task through data without being explicitly programmed.

Learning Concepts

- Unuspervised Learning
- Supervised Learning
- Reinforcement Learning

Artificial Neural Networks

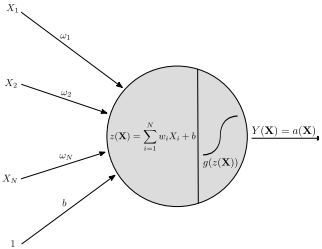
- General Function Approximators
- AlphaGo, Self-Driving Cars, Face recognition, NLP
- Incomplete Theory, models difficult to interpret
- NN design: more an art than a science

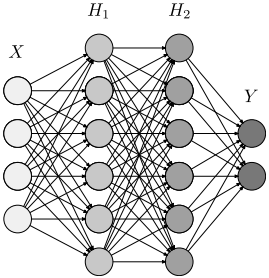
Neural Networks

- Artificial Neural Network (ANN): A non-linear mapping from inputs to ouputs: $\mathbf{M} : \hat{X} \to \hat{Y}$
- An ANN is a nesting of linear and non-linear functions arranged in a directed acyclic graph:

$$\hat{Y} \approx Y = M(\hat{X}) = \sigma_L \left(W_L \left(\sigma_{L-1} \left(W_{L-1} \left(\sigma_{L-2} \left(...W_1(\hat{X}) \right) \right) \right) \right) \right), \tag{1}$$

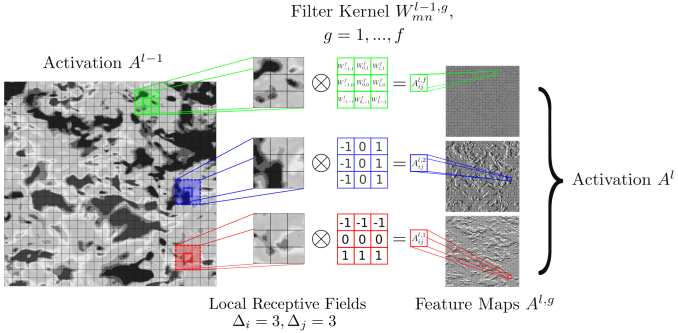
  with W being an affine mapping and σ a non-linear function
- The entries of the mapping matrices W are the parameters or weights of the network: improved by training
- Cost function C as a measure for $\left|\hat{Y} - Y\right|$, (MSE / L_2 error) convex w.r.t to Y, but not w.r.t W: $\Rightarrow$ non-convex optimization problem requires a lot of data





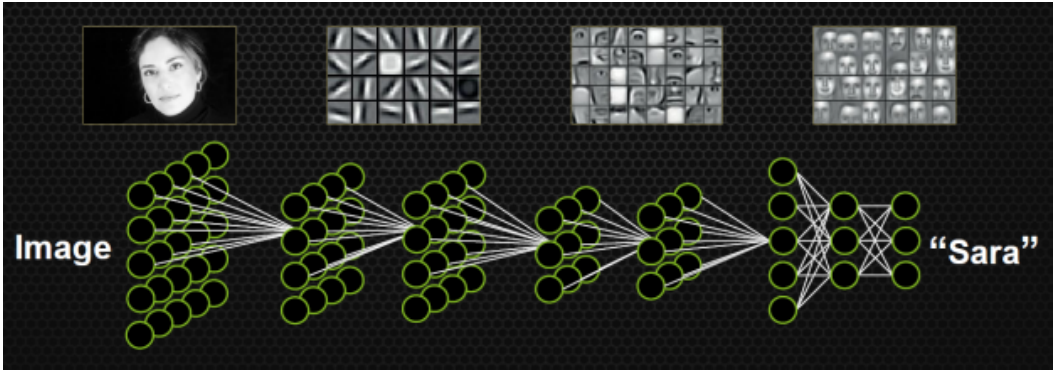
Advanced Architectures

- Convolutional Neural Networks
  - Local connectivity, multidimensional trainable filter kernels, discrete convolution, shift invariance, hierarchical representation
  - Current state of the art for multi-D data and segmentation



What does a CNN learn?

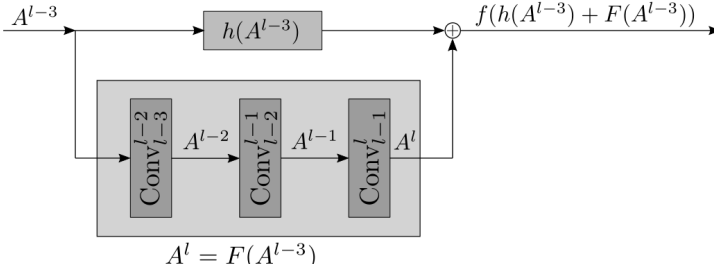
- Representation in hierarchical basis



from: H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." In ICML 2009.

Residual Neural Networks

- He et al. recognized that the prediction performance of CNNs may deteriorate with depths (not an overfitting problem)
- Introduction of skip connectors or shortcuts, most often identity mappings
- A sought mapping, e.g. $G(A^{l-3})$ is split into a linear and non-linear (residual) part
- Fast passage of the linear part through the network: hundreds of CNN layers possible
- More robust identity mapping



He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

Turbulence Models from Data

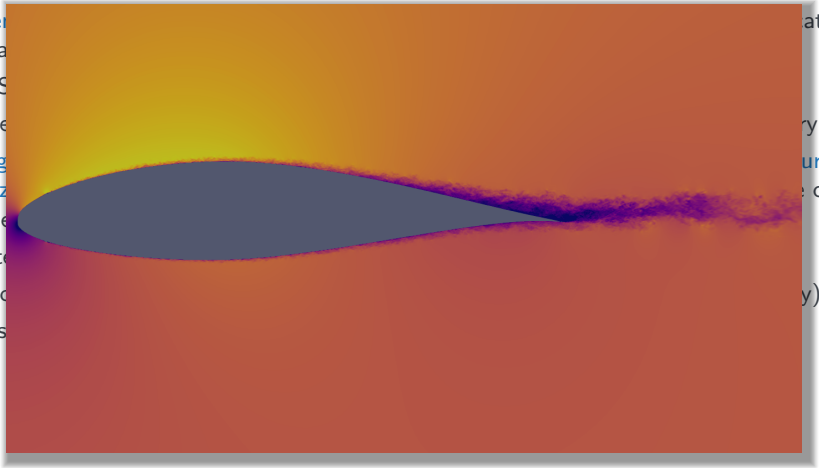
3

Turbulence in a nutshell

- Turbulent fluid motion is prevalent in naturally occurring flows and engineering applications: multiscale problem in space and time
- Navier-Stokes equations: system of non-linear PDEs (hyp. / parab.)
- Fullscale resolution (DNS) rarely feasible: Coarse scale formulation of NSE is necessary
- Filtering the NSE: Evolution equations for the coarse scale quantities, but with a closure term / regularization dependent on the filtered full scale solution $\Rightarrow$ Model depending on the coarse scale data needed!
- Two filter concepts: Averaging in time (RANS) or low-pass filter in space (LES)
- An important consequence: RANS can be discretization independent, LES is (typically) not!
- 50 years of research: Still no universal closure model

Turbulence in a nutshell

- Turbuler ations: multisca
- Navier-S
- Fullscale y
- Filtering ure term / regulariz coarse scale data nee
- Two filt
- An impo y) not!
- 50 years

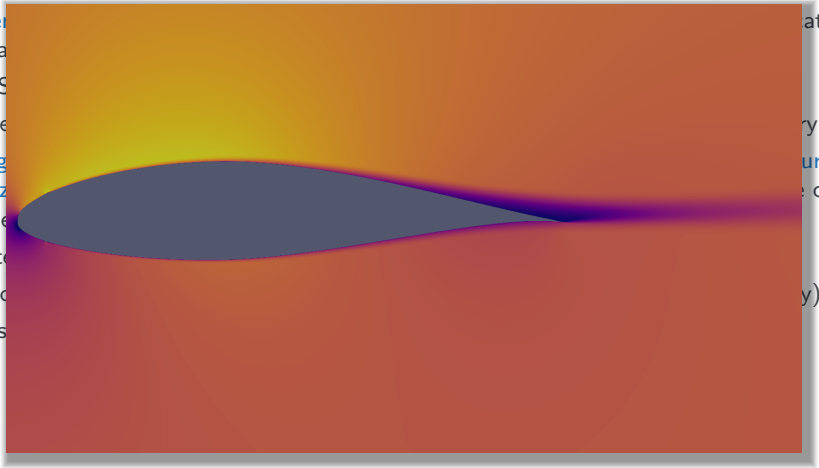


Turbulence in a nutshell

- Turbulent fluid motion is prevalent in naturally occurring flows and engineering applications: multiscale problem in space and time
- Navier-Stokes equations: system of non-linear PDEs (hyp. / parab.)
- Fullscale resolution (DNS) rarely feasible: Coarse scale formulation of NSE is necessary
- Filtering the NSE: Evolution equations for the coarse scale quantities, but with a closure term / regularization dependent on the filtered full scale solution $\Rightarrow$ Model depending on the coarse scale data needed!
- Two filter concepts: Averaging in time (RANS) or low-pass filter in space (LES)
- An important consequence: RANS can be discretization independent, LES is (typically) not!
- 50 years of research: Still no universal closure model

Turbulence in a nutshell

- Turbuler ations: multisca
- Navier-S
- Fullscale y
- Filtering ure term / regulariz coarse scale data nee
- Two filt
- An impo y) not!
- 50 years



Turbulence in a nutshell

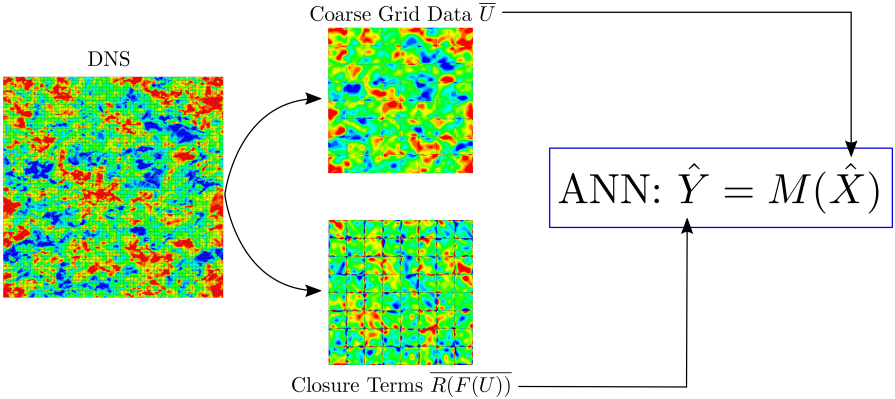
- Turbulent fluid motion is prevalent in naturally occurring flows and engineering applications: multiscale problem in space and time
- Navier-Stokes equations: system of non-linear PDEs (hyp. / parab.)
- Fullscale resolution (DNS) rarely feasible: Coarse scale formulation of NSE is necessary
- Filtering the NSE: Evolution equations for the coarse scale quantities, but with a closure term / regularization dependent on the filtered full scale solution $\Rightarrow$ Model depending on the coarse scale data needed!
- Two filter concepts: Averaging in time (RANS) or low-pass filter in space (LES)
- An important consequence: RANS can be discretization independent, LES is (typically) not!
- 50 years of research: Still no universal closure model

Idea

- Approximating an unknown, non-linear and possibly hierarchical mapping from high-dimensional input data to an output $\Rightarrow$ ANN

Idea

- Approximating an unknown, non-linear and possibly hierarchical mapping from high-dimensional input data to an output $\Rightarrow$ LES closure

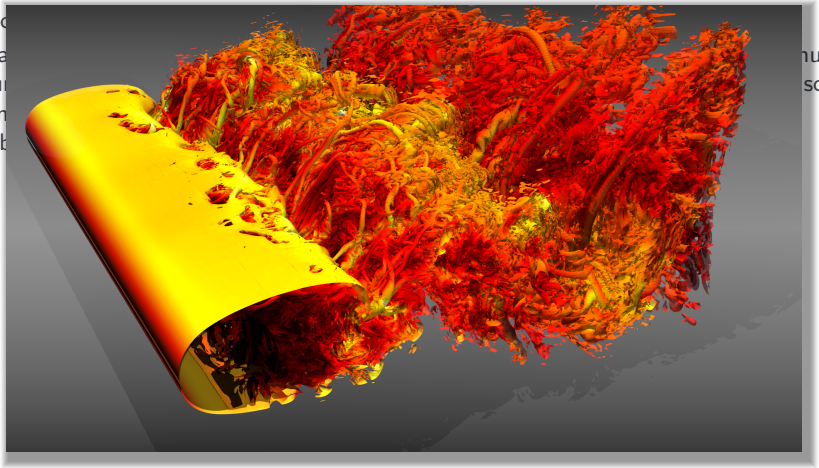


Problem Definition

- Choice of LES formulations:
  - Scale separation filter: implicit $\Leftrightarrow$ explicit, linear $\Leftrightarrow$ non-linear, discrete $\Leftrightarrow$ continuous...
  - Numerical operator: negligible $\Leftrightarrow$ part of the LES formulation, isotropic $\Leftrightarrow$ non-isotropic, commutation with filter...
  - Subgrid closure: implicit $\Leftrightarrow$ explicit, deconvolution $\Leftrightarrow$ stochastic modelling,...

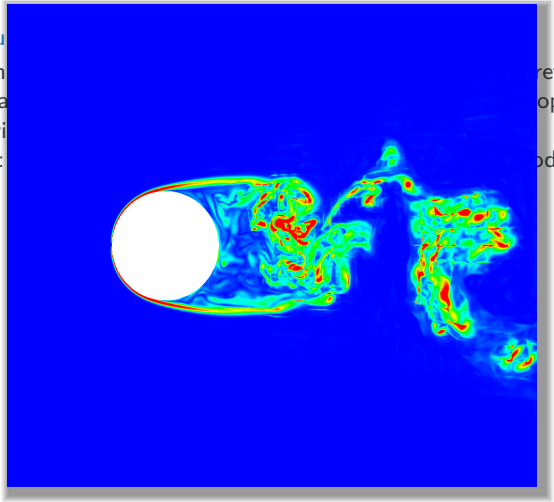
Problem Definition

- Choice o
  - Sca nuous...
  - Nu sotropic, con
  - Sub



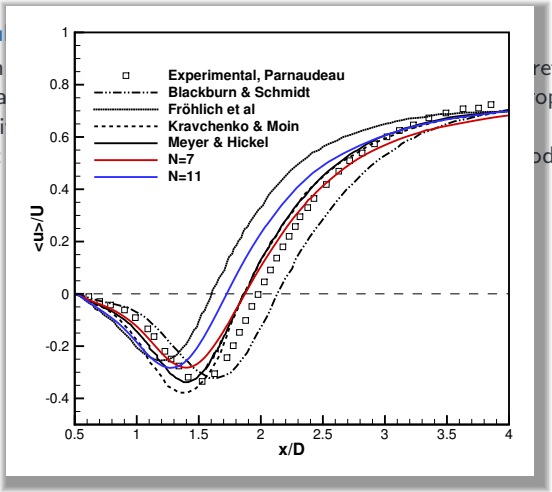
Problem Definition

- Choice of LES formu
  - Scale separation rete ⇔ continuous...
  - Numerical opera opic ⇔ non-isotropic, commutation wi
  - Subgrid closure: odelling,...



Problem Definition

- Choice of LES formu
  - Scale separation rete ⇔ continuous...
  - Numerical opera opic ⇔ non-isotropic, commutation wi
  - Subgrid closure: odelling,...



Problem Definition

- Choice of LES formulations:
  - Scale separation filter: implicit $\Leftrightarrow$ explicit, linear $\Leftrightarrow$ non-linear, discrete $\Leftrightarrow$ continuous...
  - Numerical operator: negligible $\Leftrightarrow$ part of the LES formulation, isotropic $\Leftrightarrow$ non-isotropic, commutation with filter...
  - Subgrid closure: implicit $\Leftrightarrow$ explicit, deconvolution $\Leftrightarrow$ stochastic modelling,...

- Essential for ML methods: Well-defined training data (both input and output)
- Is $\overline{U}$ known explicitly? $\Rightarrow$ For practical LES, i.e. grid-dependent LES, it is not most of the time!

Problem Definition

- Choice of LES formulations:
  - Scale separation filter: implicit $\Leftrightarrow$ explicit, linear $\Leftrightarrow$ non-linear, discrete $\Leftrightarrow$ continuous...
  - Numerical operator: negligible $\Leftrightarrow$ part of the LES formulation, isotropic $\Leftrightarrow$ non-isotropic, commutation with filter...
  - Subgrid closure: implicit $\Leftrightarrow$ explicit, deconvolution $\Leftrightarrow$ stochastic modelling,...

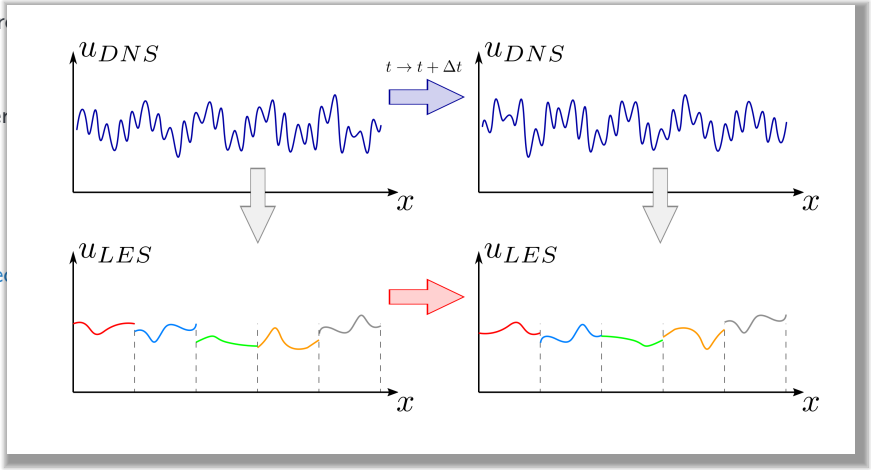
- Essential for ML methods: Well-defined training data (both input and output)
- Is $\overline{U}$ known explicitly? $\Rightarrow$ For practical LES, i.e. grid-dependent LES, it is not most of the time!

Definition: Perfect LES

- All terms must be computed on the coarse grid
- Given $\overline{U}(t_0, x) = \overline{U^{DNS}}(t_0, x) \;\; \forall\, x$, then $\overline{U}(t, x) = \overline{U^{DNS}}(t, x) \;\; \forall\, x$ and $\forall\, t > 0$

Turbulence Closure

- Filter (2)
- Imper (3)
- Perfe (4)



Turbulence Closure

- Filtered NSE:

$$\frac{\partial \overline{U}}{\partial t} + \overline{R(F(U))} = 0 \tag{2}$$

- Imperfect closure with $\hat{U} \neq \overline{U}$:

$$\frac{\partial \hat{U}}{\partial t} + \widetilde{R}(F(\hat{U})) = \underbrace{\widetilde{M}(\hat{U}, C_k)}_{\textbf{imperfect closure model}} , \tag{3}$$

- Perfect closure with $\overline{U}$

$$\frac{\partial \overline{U}}{\partial t} + \widetilde{R}(F(\overline{U})) = \underbrace{\widetilde{R}(F(\overline{U})) - \overline{R(F(U))}}_{\textbf{perfect closure model}} . \tag{4}$$

- Note $\widetilde{R}(F(\overline{U}))$ is necessarily a part of the closure, but it is known
- Perfect LES and perfect closure are not new concepts: introduced by R. Moser et al in a series of papers*, termed ideal / optimal LES

*Langford, Jacob A. & Robert D. Moser. "Optimal LES formulations for isotropic turbulence." JFM 398 (1999): 321-346.

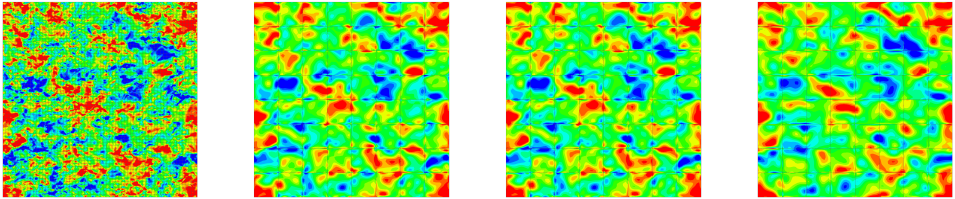
Perfect LES

$$\frac{\partial \overline{U}}{\partial t} + \overbrace{\widetilde{R}(F(\overline{U}))}^{\text{coarse grid operator}} = \underbrace{\overbrace{\widetilde{R}(F(\overline{U}))}^{\text{coarse grid operator}} - \overline{R(F(U))}}_{\textbf{perfect closure model}}.$$

- The specific operator and filter choices are not relevant for the perfect LES
- Note that the coarse grid operator is part of the closure (and cancels with the LHS)
- We choose:
  - DNS-to-LES operator $\overline{()}$: L_2 projection from DNS grid onto LES grid: We choose a discrete scale-separation filter
  - LES operator $\widetilde{()}$: 6th order DG method with split flux formulation and low dissipation Roe flux

Perfect LES

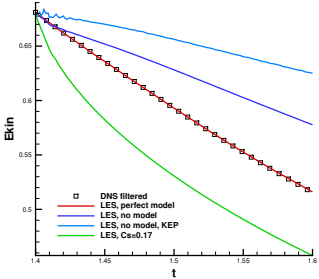
- Perfect LES runs with closure term from DNS
- Decaying homogeneous isotropic turbulence
- DNS grid: 64^3 elements, N $= 7$; LES grid: 8^3 elements, N $= 5$;

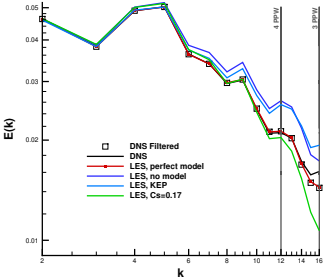


Left to right: a) DNS, b) filtered DNS, c) computed perfect LES d) LES with Smagorinsky model $C_s = 0.17$

Perfect LES

- Perfect LES runs with closure term from DNS
- Decaying homogeneous isotropic turbulence
- DNS grid: 64^3 elements, N $= 7$; LES grid: 8^3 elements, N $= 5$;





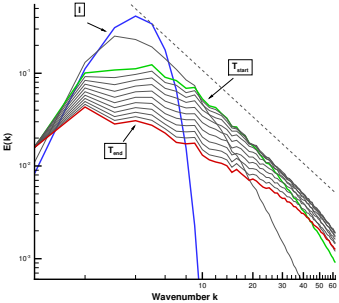
- $\Rightarrow$ Perfect LES gives well-defined target and input data for supervised with NN

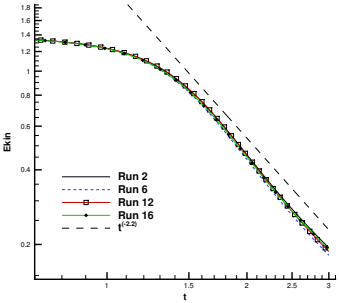
Training and Results

4

Data Acquisition: Decaying Homogeneous Isotropic Turbulence

- Ensemble of DNS runs of decaying homogeneous isotropic turbulence with initial spectrum defined by Chasnov (1995) initialized by Rogallo (1981) procedure and $Re_\lambda = 180$ at start
- Data collection in the range of exponential energy decay: 25 DHIT realizations with 134 Mio DOF each computed on CRAY XC40 (approx. 400,000 CPUh, 8200 cores)
- Compute coarse grid terms from DNS-to-LES operator

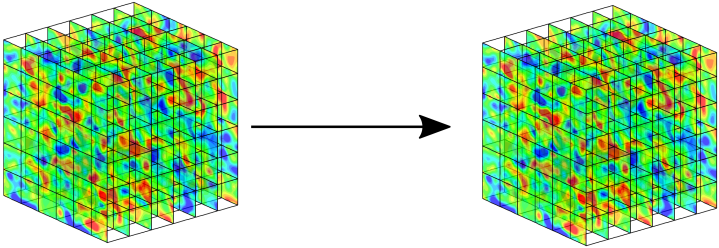




Features and Labels

- Each sample: A single LES grid cell with 6^3 solution points
- Input features: velocities and LES operator: $\overline{u_i}, \widetilde{R}(F(\overline{U}))$
- Output labels: DNS closure terms on the LES grid $\overline{R(F(U))}$

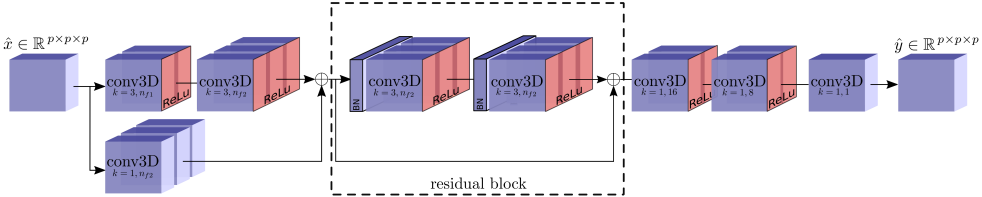
$$\hat{X} = \left\{ \hat{x} \in \mathbb{R}^{6\times p\times p\times p} \,|\, \hat{x} = (\overline{u}_{ijk}, \overline{v}_{ijk}, \overline{w}_{ijk}, \widetilde{R}(F(\overline{U^1}))_{ijk}, \widetilde{R}((F(\overline{U^2}))_{ijk}, \widetilde{R}(F(\overline{U^3}))_{ijk}), \quad \text{with } i,j,k = 0,...,p-1 \right\}$$



$$\hat{Y} = \left\{ \hat{y} \in \mathbb{R}^{3\times p\times p\times p} \,|\, \hat{y} = \overline{R(F(U))^n_{ijk}}, \quad \text{with } n = 1,...,3;\ i,j,k = 0,...,p-1 \right\}$$

Networks and Training

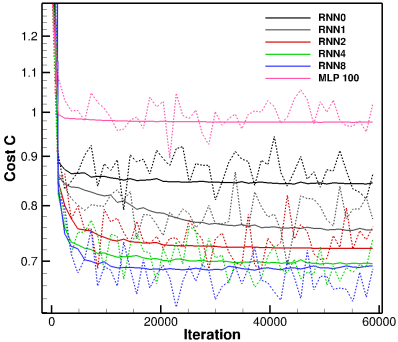
- CNNs with skip connections (RNN), batch normalization, ADAM optimizer, data augmentation
- Different network depths (no. of residual blocks)
- For comparison: MLP with 100 neurons in 1 hidden layer*
- Implementation in Python / Tensorflow, Training on K40c and P100 at HLRS
- Split in training, semi-blind validation and blind test DHIT runs



*Gamahara & Hattori. "Searching for turbulence models by artificial neural network." Physical Review Fluids 2.5 (2017)

Training Results I: Costs

- Cost function for different network depths
- RNNs outperform MLP, deeper networks learn better
- The approach is data-limited! NNs are very data-hungry!



Training Results II: Correlation

Network	a, b	$\mathcal{CC}(a,b)$	$\mathcal{CC}^{inner}(a,b)$	$\mathcal{CC}^{surf}(a,b)$
RNN0	$\overline{R(F(U))^1}, \overline{R(F(U))^1}^{ANN}$	0.347676	0.712184	0.149090
	$\overline{R(F(U))^2}, \overline{R(F(U))^2}^{ANN}$	0.319793	0.663664	0.134267
	$\overline{R(F(U))^3}, \overline{R(F(U))^3}^{ANN}$	0.326906	0.669931	0.101801
RNN4	$\overline{R(F(U))^1}, \overline{R(F(U))^1}^{ANN}$	0.470610	0.766688	0.253925
	$\overline{R(F(U))^2}, \overline{R(F(U))^2}^{ANN}$	0.450476	0.729371	0.337032
	$\overline{R(F(U))^3}, \overline{R(F(U))^3}^{ANN}$	0.449879	0.730491	0.269407

- High correlation achievable with deep networks
- For surfaces: one-sidedness of data / filter kernels

Training Results III: Feature Sensitivity

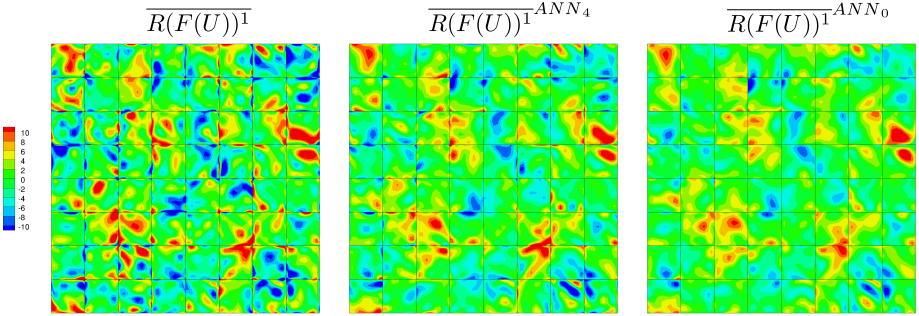
Set	Features	$\mathcal{CC}^1$	$\mathcal{CC}^2$	$\mathcal{CC}^3$
1	$u_i,\ \widetilde{R}(F(\overline{U^i})),\ i=1,2,3$	0.4706	0.4505	0.4499
2	$u_i,\ i=1,2,3$	0.3665	0.3825	0.3840
3	$\widetilde{R}(F(\overline{U^i})),\ i=1,2,3$	0.3358	0.3066	0.3031
4	$\rho, p, e, u_i,\ \widetilde{R}(F(\overline{U^i})),\ i=1,2,3$	0.4764	0.4609	0.4580
5	$u_1,\ \widetilde{R}(F(\overline{U^1}))$	0.3913		

Feature sets and resulting test correlations. $\mathcal{CC}^i$ with $i=1,2,3$ denotes the cross correlation between the targets and network outputs $\mathcal{CC}(\overline{R(F(U)^i)}, \overline{R(F(U))^i}^{ANN})$. Set 1 corresponds to the original feature choice; Set 5 corresponds to the RNN4 architecture, but with features and labels for the $u-$momentum component only.

- Both the coarse grid primitive quantities as well as the coarse grid operator contribute strongly to the learning success
- Better learning for 3D cell data than pointwise data

Training Results IV: Visualization

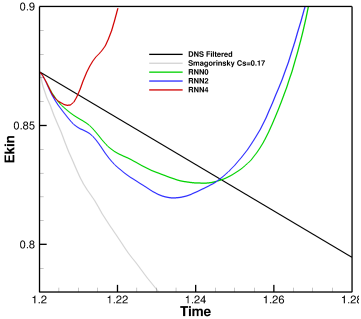
- "Blind" application of the trained network to unknown test data
- Cut-off filter: no filter inversion / approximate deconvolution

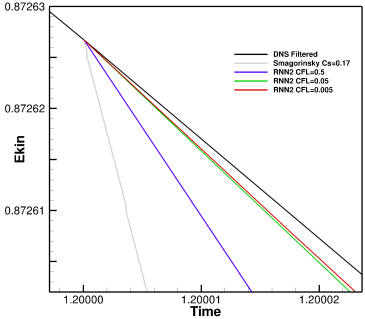


LES with NN-trained model I

$$\frac{\partial \overline{U}}{\partial t} + \widetilde{R}(F(\overline{U})) = \widetilde{R}(F(\overline{U})) \underbrace{-\overline{R(F(U))}}_{\textbf{ANN closure}}.$$

- Perfect LES is possible, but the NN-learned mappings are approximate
- No long term stability, but short term stability and dissipation



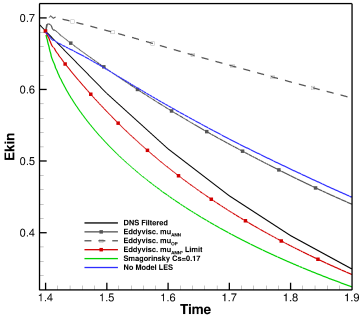


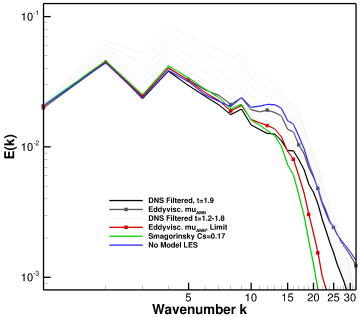
LES with NN-trained model II

$$\frac{\partial \overline{U}}{\partial t} + \widetilde{R}(F(\overline{U})) = \underbrace{\widetilde{R}(F(\overline{U})) - \overline{R(F(U))}}_{\text{data-based eddy viscosity model}} .$$

- Simplest model: Eddy viscosity approach with μ_{ANN} from

$$\widetilde{R}(F(\overline{U^i})) - \overline{R(F(U^i))} \approx \mu_{ANN}\, \widetilde{R}(F^{visc}(\overline{U^i}, \nabla \overline{U^i})) \tag{5}$$





Summary and Conclusion

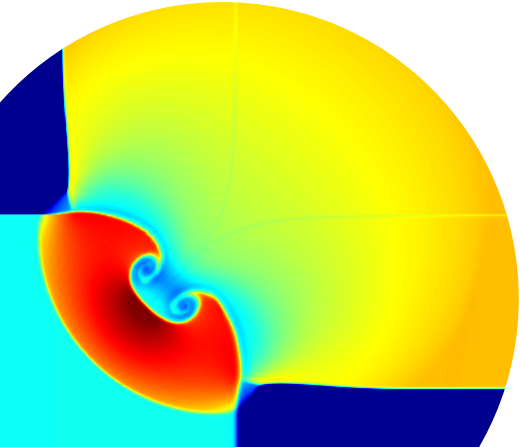
5

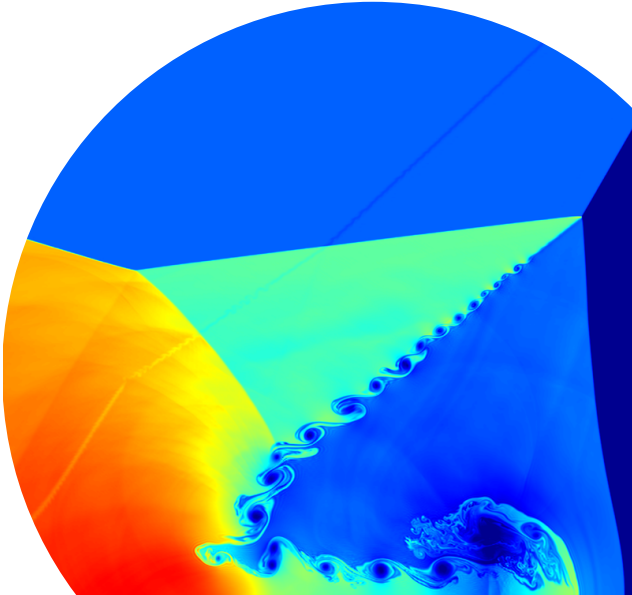
Summary

- Perfect / optimal LES framework: well-defined target quantities for learning
- Learning the exact closure terms from data is possible
- Deeper RNNs learn better
- High order methods are a natural fit to CNN: volume data
- Our process is data-limited, i.e. learning can be improved with more data
- Achievable $\mathcal{CC} \approx 45\%$, with up to $\approx 75\%$ for inner points
- Both the coarse grid velocities and the coarse grid operator contribute strongly to learning
- The resulting ANN models are dissipative
- No long term stability due to approximate model
- Simplest way to construct a stable model: Data-informed, local eddy-viscosity
- Other approaches to construct models from prediction of closure terms under investigation

flexi-project.org

Thank you for your attention!





History of ANNs

- Some important publications:
  - McCulloch-Pitts (1943): First compute a weighted sum of the inputs from other neurons plus a bias: the perceptron
  - Rosenblatt (1958): First to generate MLP from perceptrons
  - Rosenblatt (1962): Perceptron Convergence Theorem
  - Minsky and Papert (1969): Limitations of perceptrons
  - Rumelhart and Hinton (1986): Backpropagation by gradient descent
  - LeCun (1995): "LeNet", convolutional networks
  - Hinton (2006): Speed-up of backpropagation
  - Krizhevsky (2012): Convolutional networks for image classification
  - Ioffe (2015): Batch normalization
  - He et al. (2016): Residual networks
  - AlphaGo, DeepMind...

Closure Terms for LES

- For grid dependent LES: coarse grid operator is part of the closure
- Dual role of closure: cancel operator effects and model unknown term
- DNS grid: 64^3 elements, $N = 7$; LES grid: 8^3 elements, $N = 5$;

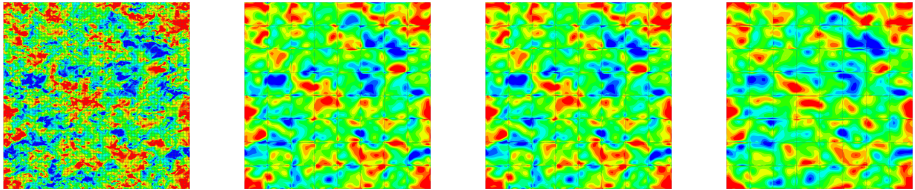


Figure: Left to right: a) DNS, b) filtered DNS, c) computed perfect LES d) LES with Smagorinsky model $C_s = 0.17$

Some thoughts on data-informed models, engineering and HPC

- Machine Learning is not a silver bullet
- First successes: ML can help build subscale models from data, not just for turbulence
- A lot of representative data is needed... maybe we already have the data? Computations, experiments...
- In this work, the computational times were: DNS: $\mathcal{O}(10^5)$ CPUh, data preparation $\mathcal{O}(10^3)$, Training the RNN: $\mathcal{O}(10^1 - 10^2)$: Is it worth it?
- Incorporating physical constraints (e.g. realizability, positivity) field of research
- Self-learning algorithms: Reinforcement learning
- "Philosophical aspects": Interpretability of the models and "who should learn what?"
- HPC: Training has to done on GPUs (easy for supervised learning, bit more complicated for reinforcement learning), but ...
- What about model deployment? GPU (native) or CPU (export model)?
- Coupling of CFD solver (Fortran) to Neural Network (python): In our case, f2py is a very cumbersome solution
- Hybrid CPU/GPU codes, or rewrite it all for the GPU?
- Data storage policy: where to compute/store the data (reproducibility)

Definitions and Concepts

An attempt at a definition:

Machine learning describes algorithms and techniques that progressively improve performance on a specific task through data without being explicitly programmed.

Learning Concepts

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

Artificial Neural Networks

- General Function Approximators
- AlphaGo, Self-Driving Cars, Face recognition, NLP
- Incomplete Theory, models difficult to interpret