Lessons learnt in learning models from data

preserving structure in closure models for fluid flows

Benjamin Sanderse, Syver Agdestein, Nikolaj Mücke, Marius Kurz, Toby van Gastelen

NUM-SCARS
CWI Amsterdam, January 23, 2026

What is structure?

Properties of a continuous mathematical-physical model that we want to mimic in a discrete sense

- **Conservation laws (primary and secondary)**
- **Symmetries**
- Invariance and equivariance
- Symplecticity
- Variational principle
- Positivity, total variation, maximum principle
- Lake at rest, pressure equilibrium
- ODE and DAE structure

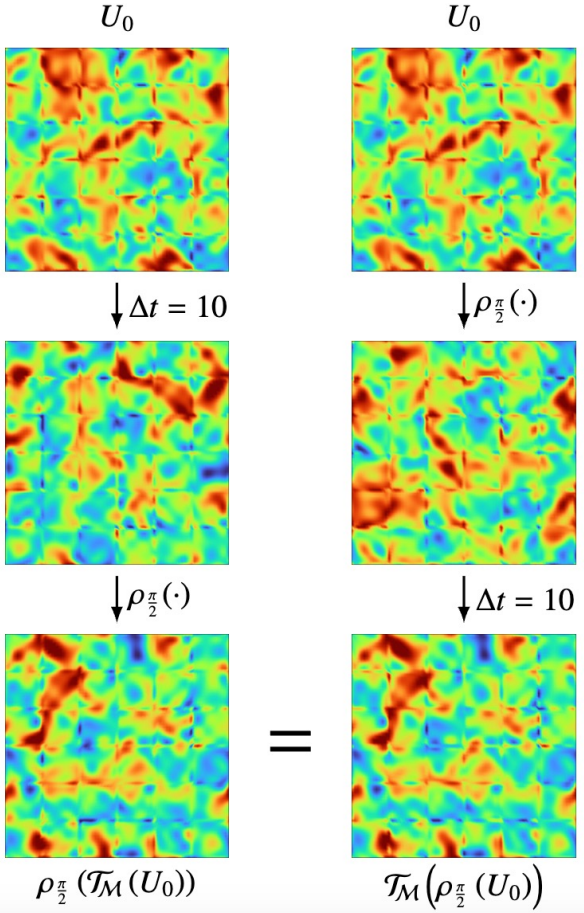
Symmetries of the NS equations

PDE symmetries (Lie groups):

- Time translation
- **Rotation**
- Reflection
- Generalized Galilean transformation
- Pressure translation
- Scaling transformation

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) + \nu \mathcal{D}(\boldsymbol{u}) - \mathcal{G}(p)$$

$$\mathcal{M}(\boldsymbol{u}) = 0$$



Symmetries in the NS equations

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) + \nu \mathcal{D}(\boldsymbol{u}) - \mathcal{G}(p)$$
$$\mathcal{M}(\boldsymbol{u}) = 0$$

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \int \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}\Omega$$
$$K(\boldsymbol{u}) := \frac{1}{2} \|\boldsymbol{u}\|^2$$

Symmetries of individual terms:

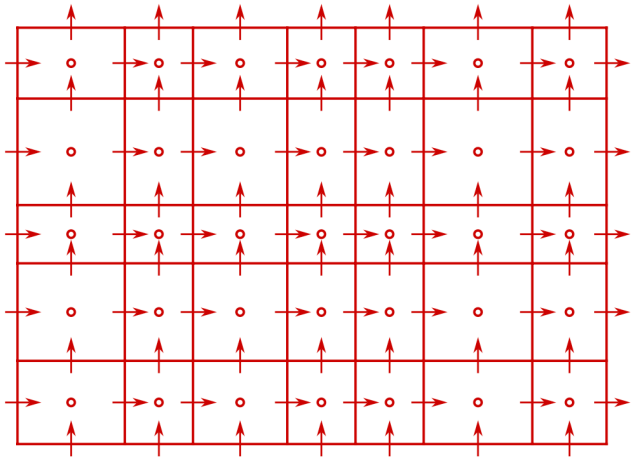
- Skew-symmetry
- Symmetry
- Div-grad

$$\langle \mathcal{C}(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{w} \rangle = -\langle \boldsymbol{v}, \mathcal{C}(\boldsymbol{u}, \boldsymbol{w}) \rangle$$
$$\langle \mathcal{D}(\boldsymbol{u}), \boldsymbol{v} \rangle = \langle \boldsymbol{u}, \mathcal{D}(\boldsymbol{v}) \rangle$$
$$\langle \mathcal{M}(\boldsymbol{u}), p \rangle = -\langle \boldsymbol{u}, \mathcal{G}(p) \rangle$$

Energy equality: $$\frac{\mathrm{d}K}{\mathrm{d}t} = -\nu \|\nabla \boldsymbol{u}\|^2$$

Symmetry-preserving discretization of NS

- Symmetries to be kept during **discretization** (in space and time)
- Discrete energy equality -> **stability independent of mesh or time step**
- Generalization to compressible flows via **entropy functions** $s(\boldsymbol{u})$

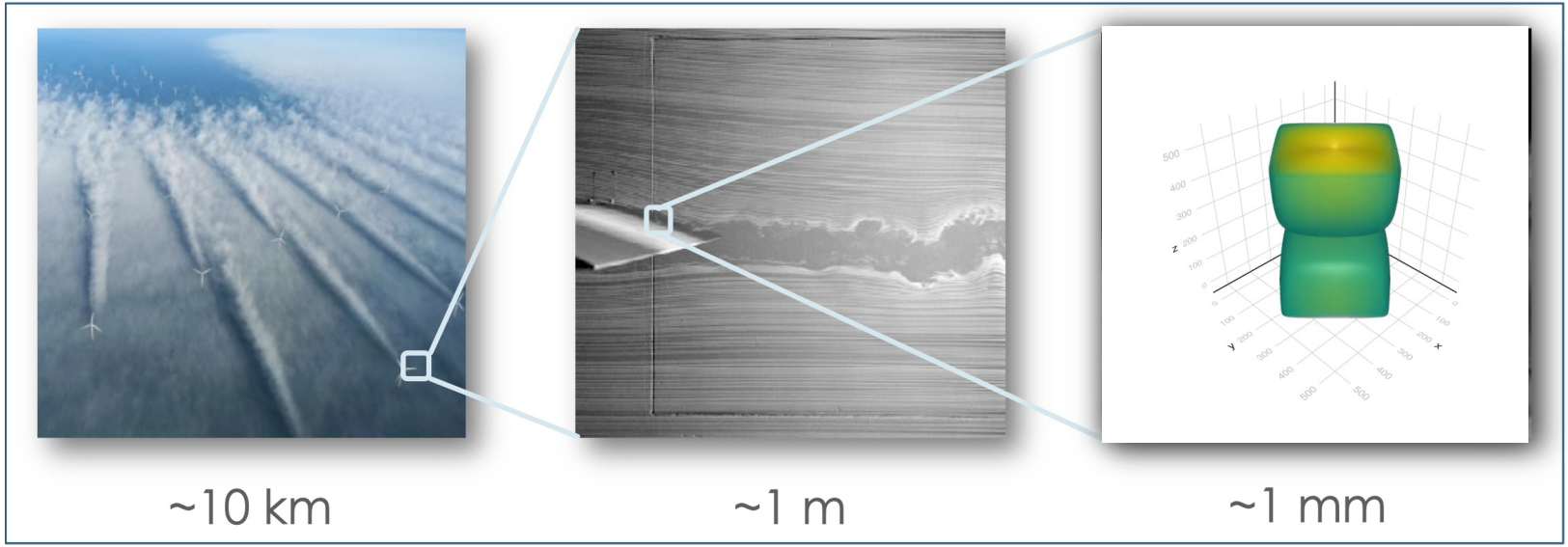


$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial f(\boldsymbol{u})}{\partial x} = 0$$

$$\frac{d\mathcal{S}[\boldsymbol{u}]}{dt} \leq 0$$

$$\mathcal{S}[\boldsymbol{u}] := \int_{\Omega} s(\boldsymbol{u}) dx$$

Simulating all scales is unfeasible



Urgent need for reduced models

- Projection-based approaches
  - Reduced-order models (ROMs)
  - Learn a basis based on DNS snapshots
  - Project equations on linear/non-linear manifold
  - => **Stability / accuracy problem**

- Filtering-based approaches
  - Large-eddy simulation
  - Remove small scales with filter
  - Learn effect of small-scale physics (closure problem)
  - => **Stability / accuracy problem**

Foundations of Data Science

doi:10.3934/fods.2024043

SCIENTIFIC MACHINE LEARNING FOR CLOSURE MODELS IN MULTISCALE PROBLEMS: A REVIEW

Benjamin Sanderse[*1], Panos Stinis[2], Romit Maulik[3] and Shady E. Ahmed[2]

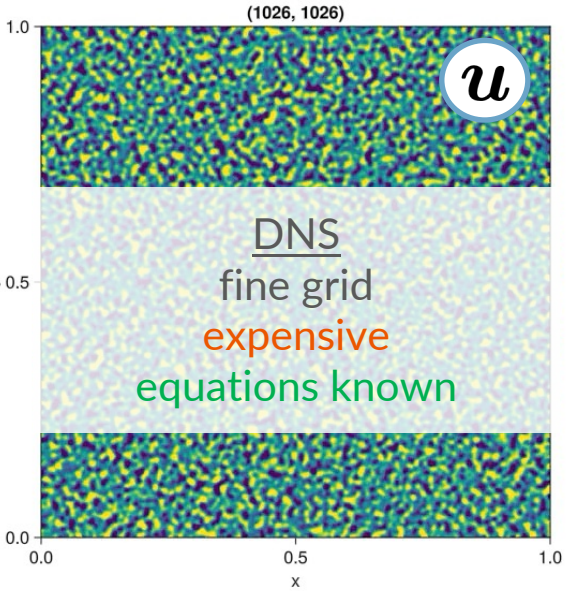
[1]Scientific Computing Group, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, The Netherlands

[2]Advanced Computing, Mathematics and Data Division, Pacific Northwest National Laboratory, Richland, 99352, WA, USA

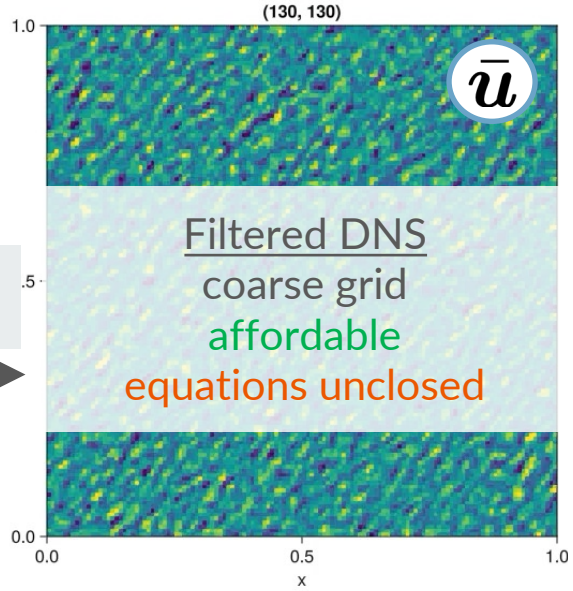
[3]College of Information Sciences and Technology, The Pennsylvania State University, Westgate Building, University Park, 16802, PA, USA

Abstract. Closure problems are omnipresent when simulating multiscale systems, where some quantities and processes cannot be fully prescribed despite their effects on the simulation's accuracy. Recently, scientific machine learning approaches have been proposed as a way to tackle the closure problem, combining traditional (physics-based) modeling with data-driven (machine-learned) techniques, typically through enriching differential equations with neural networks. This paper reviews the different reduced model forms, distinguished by the degree to which they include known physics, and the different objectives of a priori and a posteriori learning. The importance of adhering to physical laws (such as symmetries and conservation laws) in choosing the reduced model form and choosing the learning method is discussed. The effect of spatial and temporal discretization and recent trends toward discretization-invariant models are reviewed. In addition, we make the connections between closure problems and several other research disciplines: inverse problems, Mori-Zwanzig theory, and multi-fidelity methods. In conclusion, much progress has been made with scientific machine learning approaches for solving closure problems, but many challenges remain. In particular, the generalizability and interpretability of learned models is a major issue that needs to be addressed further.

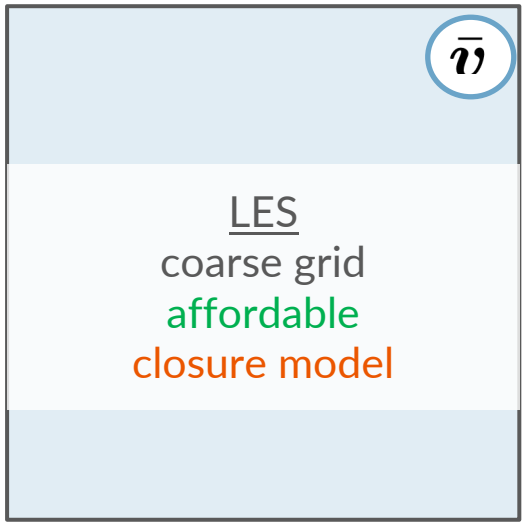
Filtering the Navier-Stokes equations



Filter
Δ



$\approx$



$$\mathcal{F}(\boldsymbol{u}) = \mathbf{0}$$

$$\overline{\mathcal{F}(\boldsymbol{u})} = \mathbf{0}$$

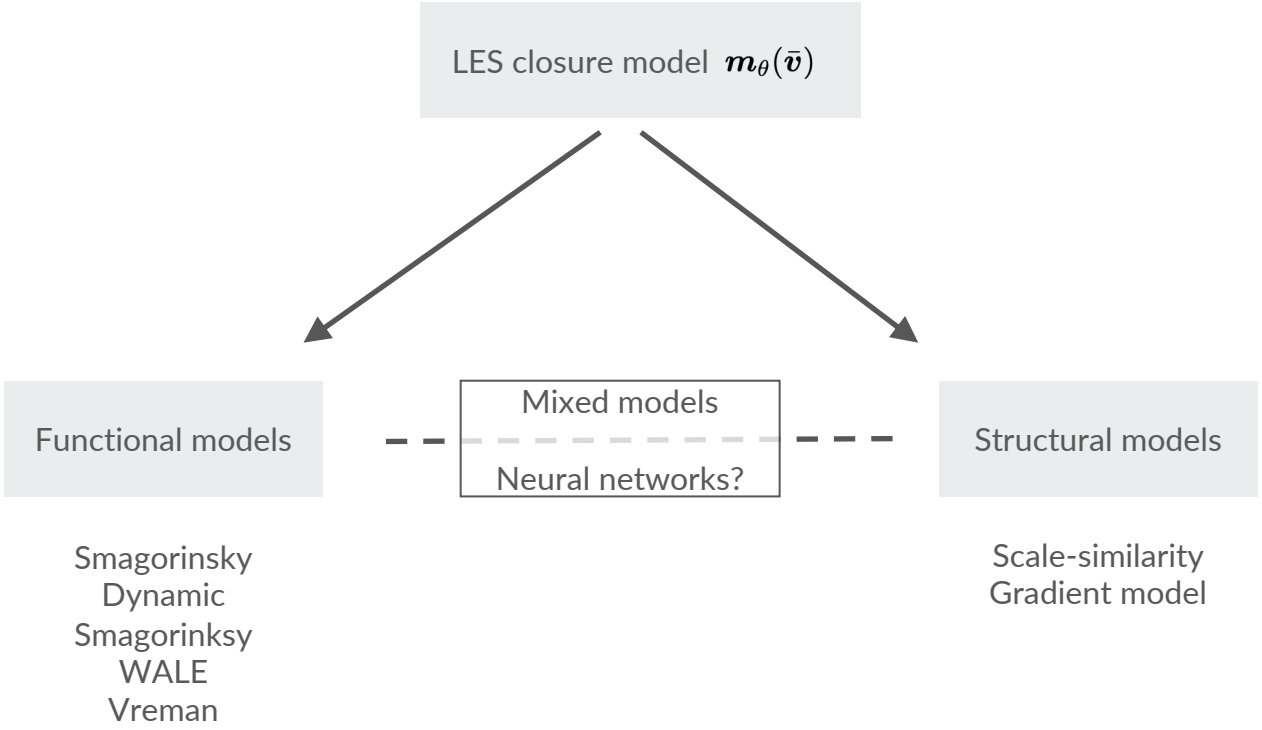
$$\mathcal{F}(\bar{\boldsymbol{v}}) = \boldsymbol{m}_\theta(\bar{\boldsymbol{v}})$$

$$\mathcal{F}(\boldsymbol{u}) := \frac{\partial \boldsymbol{u}}{\partial t} + \mathcal{C}(\boldsymbol{u}, \boldsymbol{u}) - \nu\mathcal{D}(\boldsymbol{u}) + \mathcal{G}(p)$$

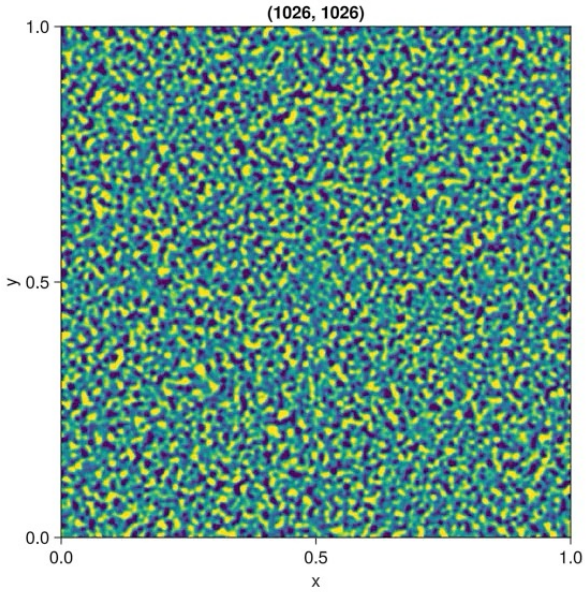
$$\mathcal{F}(\bar{\boldsymbol{u}}) \neq \mathbf{0}$$

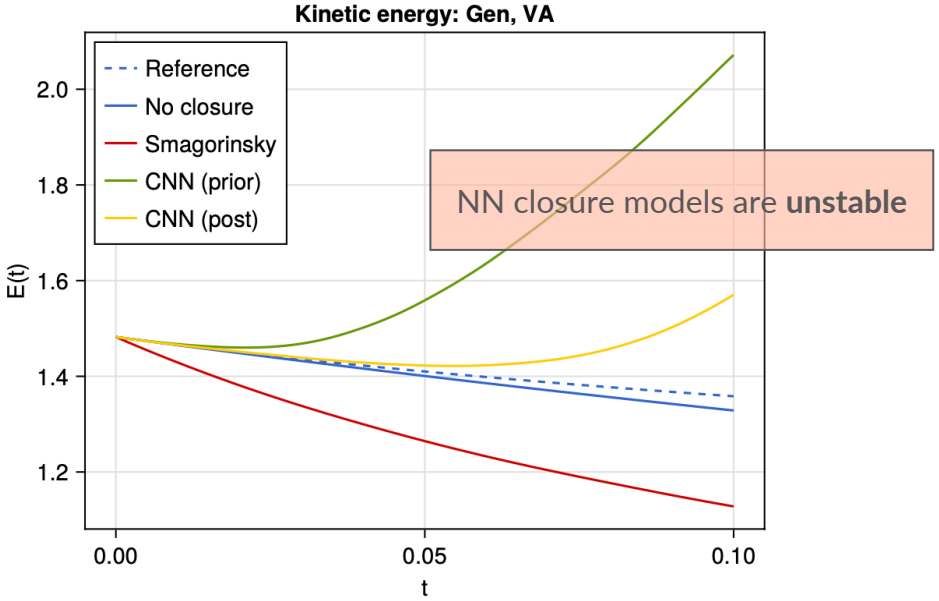
$$\begin{aligned}\boldsymbol{m}_\theta(\bar{\boldsymbol{u}}) &\approx \overline{\mathcal{F}(\boldsymbol{u})} - \mathcal{F}(\bar{\boldsymbol{u}}) \\ &= \overline{\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})} - \nabla \cdot (\bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{u}})\end{aligned}$$

Filtering the Navier-Stokes equations



Neural closure models suffer from instability





DNS: $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u})$

LES: $\frac{\mathrm{d}\bar{\mathbf{v}}}{\mathrm{d}t} = \mathbf{f}(\bar{\mathbf{v}}) + \mathbf{m}_\theta(\bar{\mathbf{v}})$

Accuracy and stability with structure preservation

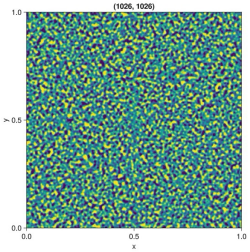
1. Change equation form: energy-stable closure
2. Change filter: discretization-consistency
3. Change NN architecture: preserve symmetries

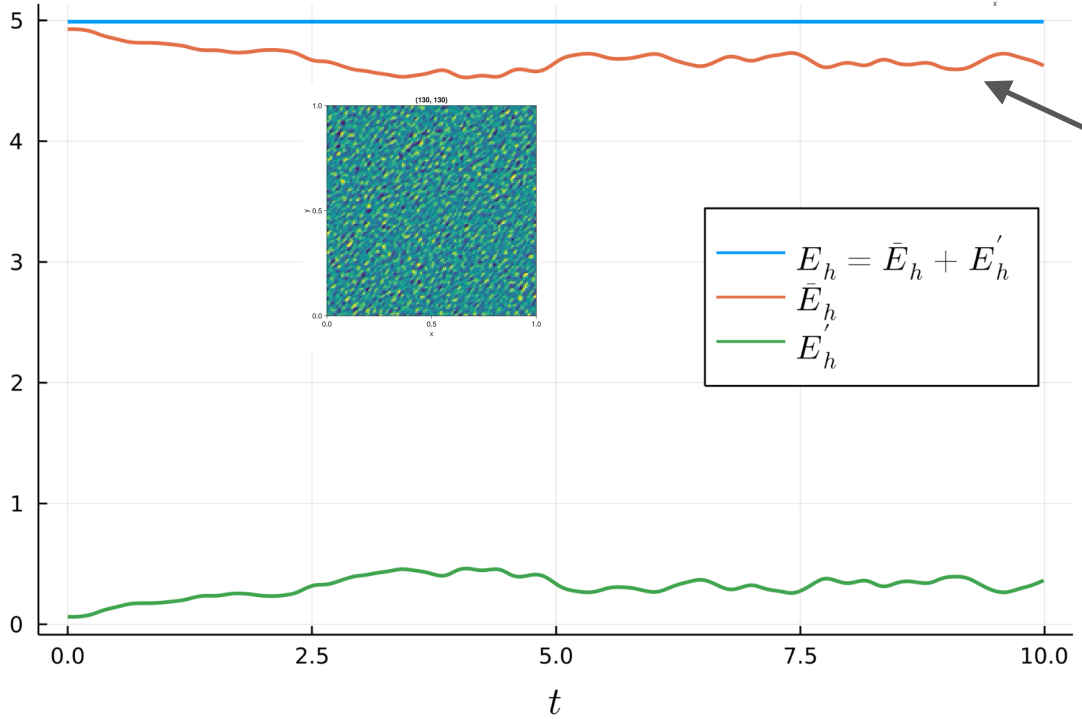
Hard constraints

4. Change loss function: a-posteriori learning (trajectory fitting)
5. Regularize through noise: stochastic differential equations

Soft constraints

1. Energy-stable closure

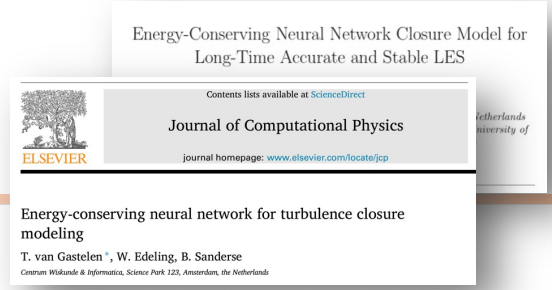


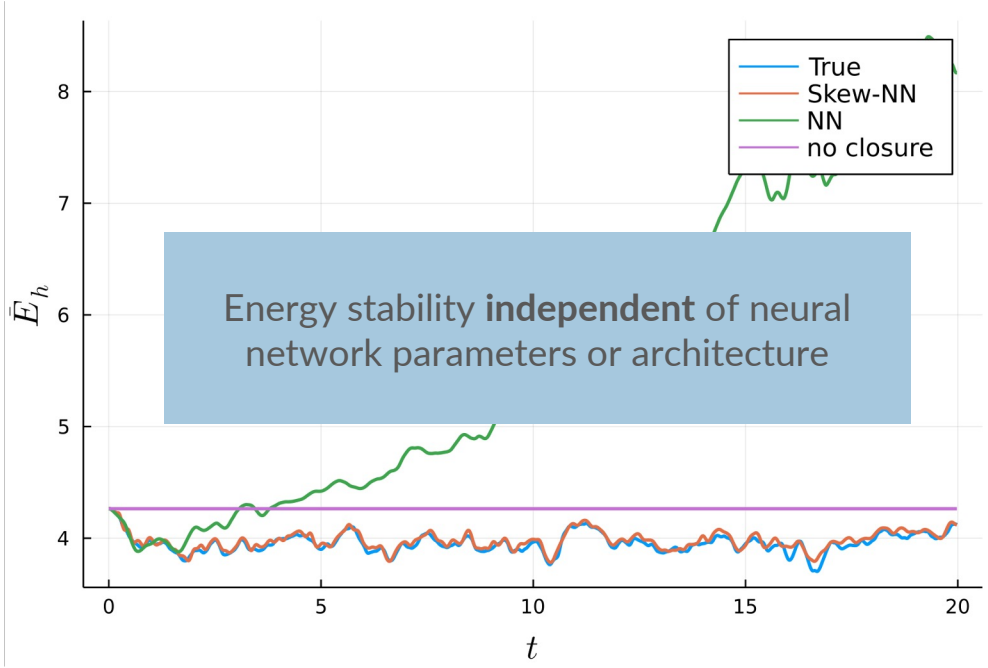


Energy of large scales is **not conserved**

$$\frac{\mathrm{d}E_h}{\mathrm{d}t} = \frac{\mathrm{d}\bar{E}_h(\bar{\mathbf{u}})}{\mathrm{d}t} + \frac{\mathrm{d}E'_h(\mathbf{u}')}{\mathrm{d}t} = 0$$

1. Energy-stable closure





Energy stability **independent** of neural network parameters or architecture

Extended neural closure model

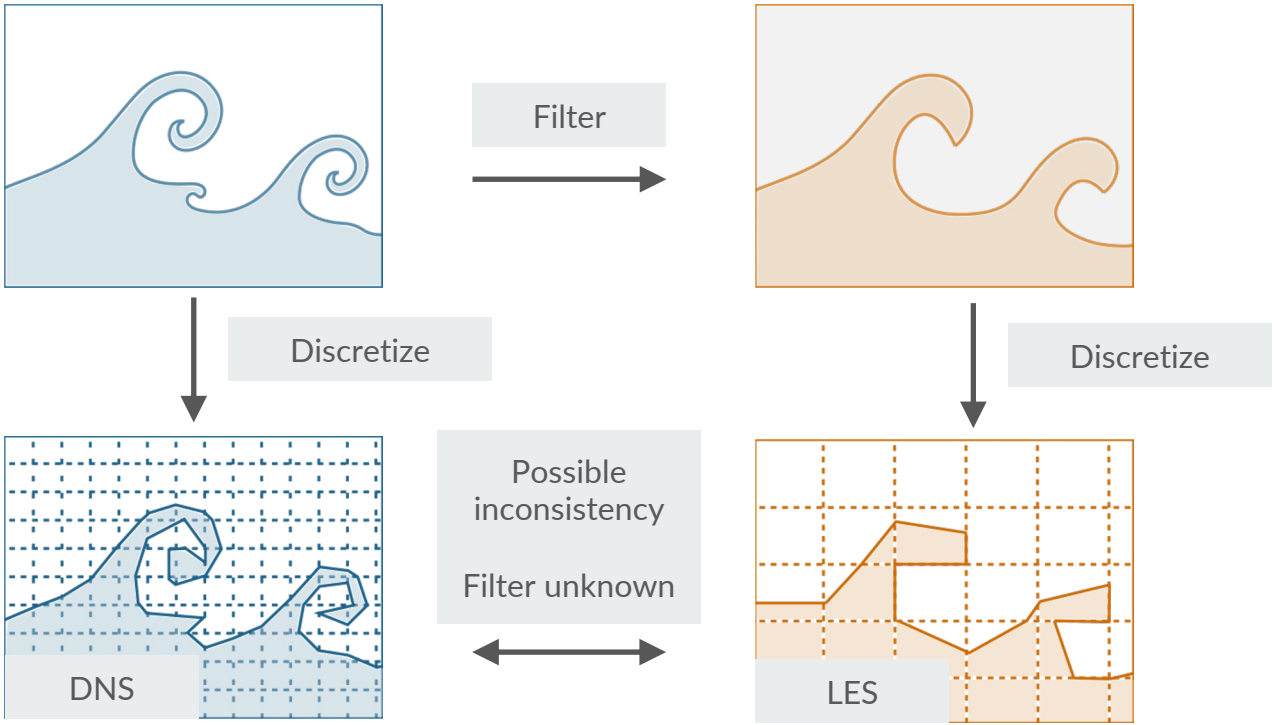
$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\bar{\mathbf{v}}\\ \mathbf{s}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\bar{\mathbf{v}})\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{m}_\theta^v(\bar{\mathbf{v}},\mathbf{s})\\ \mathbf{m}_\theta^s(\bar{\mathbf{v}},\mathbf{s})\end{bmatrix}$$

$$\begin{bmatrix}\mathbf{m}_\theta^v(\bar{\mathbf{v}},\mathbf{s})\\ \mathbf{m}_\theta^v(\bar{\mathbf{v}},\mathbf{s})\end{bmatrix} = \mathcal{K}_\theta(\bar{\mathbf{v}},\mathbf{s})\begin{bmatrix}\bar{\mathbf{v}}\\ \mathbf{s}\end{bmatrix}$$

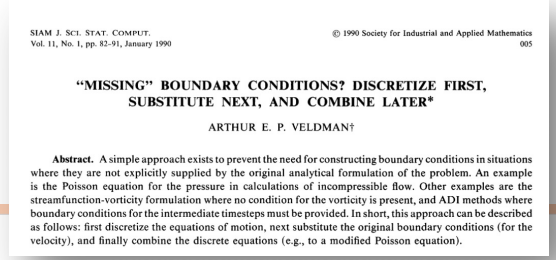
Skew-symmetric "matrix" $\mathcal{K}$ leads to exact energy conservation:

$$\frac{\mathrm{d}\bar{E}_h(\bar{\mathbf{v}})}{\mathrm{d}t} + \frac{\mathrm{d}\frac{1}{2}\|\mathbf{s}\|^2}{\mathrm{d}t} = 0$$

2. Discretization-consistent filters



2. Consistent filters via “discretize first”

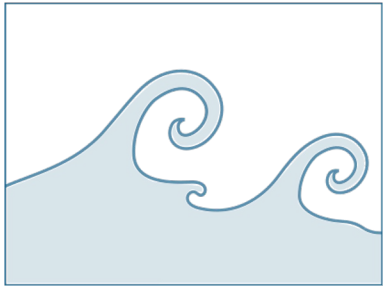


SIAM J. SCI. STAT. COMPUT.
Vol. 11, No. 1, pp. 82–91, January 1990

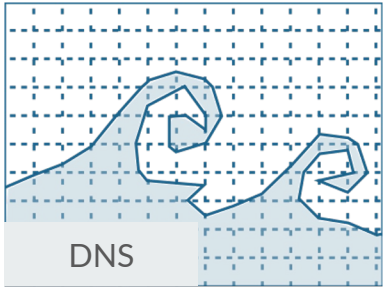
“MISSING” BOUNDARY CONDITIONS? DISCRETIZE FIRST, SUBSTITUTE NEXT, AND COMBINE LATER*

ARTHUR E. P. VELDMAN†

Abstract. A simple approach exists to prevent the need for constructing boundary conditions in situations where they are not explicitly supplied by the original analytical formulation of the problem. An example is the Poisson equation for the pressure in calculations of incompressible flow. Other examples are the streamfunction-vorticity formulation where no condition for the vorticity is present, and ADI methods where boundary conditions for the intermediate timesteps must be provided. In short, this approach can be described as follows: first discretize the equations of motion, next substitute the original boundary conditions (for the velocity), and finally combine the discrete equations (e.g., to a modified Poisson equation).

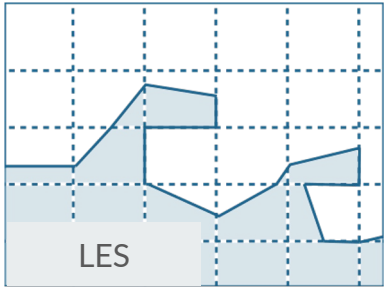


Discretize



DNS

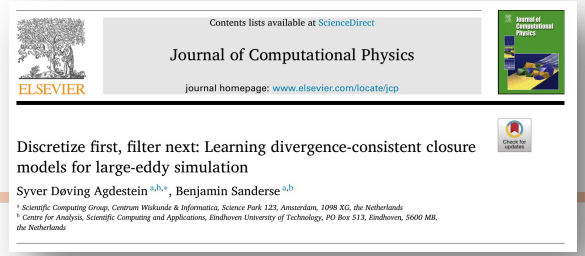
Filter



LES

- **Consistency** between training data and prediction environment
- Neural networks work on discrete data
- Incorporates discretization errors
- Easier to handle boundary conditions

2. Consistent filters via “discretize first”



$$\nabla \cdot \boldsymbol{u} = 0$$

Discretize: FVM, staggered grid
Exact divergence-freeness

$$\mathbf{D}\mathbf{u} = 0$$

$$\bar{\mathbf{u}} = \boldsymbol{\Phi}\mathbf{u}$$

$$\bar{\mathbf{D}}\bar{\mathbf{u}} = 0$$

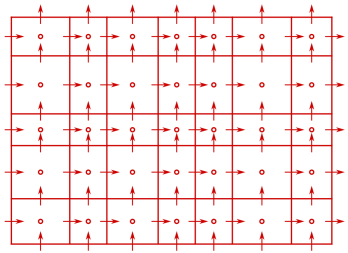
Face-averaging filter
(divergence-preserving)

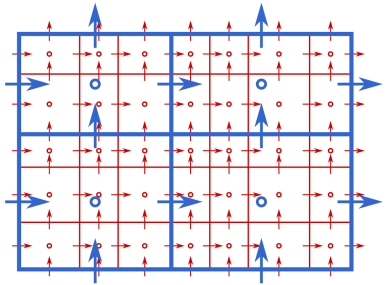
Closure model is divergence-free!

$$\bar{\mathbf{D}}\bar{\mathbf{v}} = 0$$

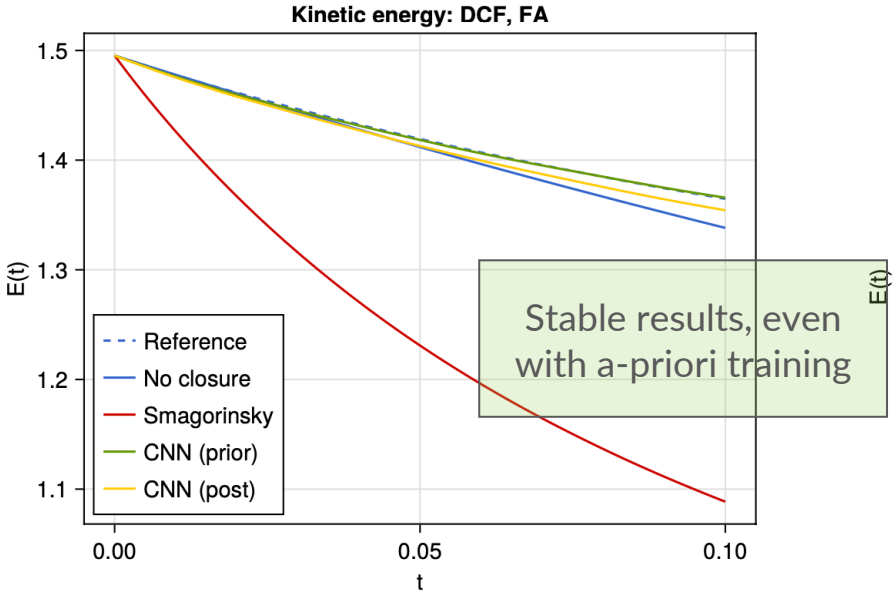
$$\frac{\mathrm{d}\bar{\mathbf{v}}}{\mathrm{d}t} = \bar{\mathbf{F}}(\bar{\mathbf{v}}) + \mathbf{m}_\theta(\bar{\mathbf{v}}) - \bar{\mathbf{G}}\bar{\mathbf{q}}$$

Energy
consistency

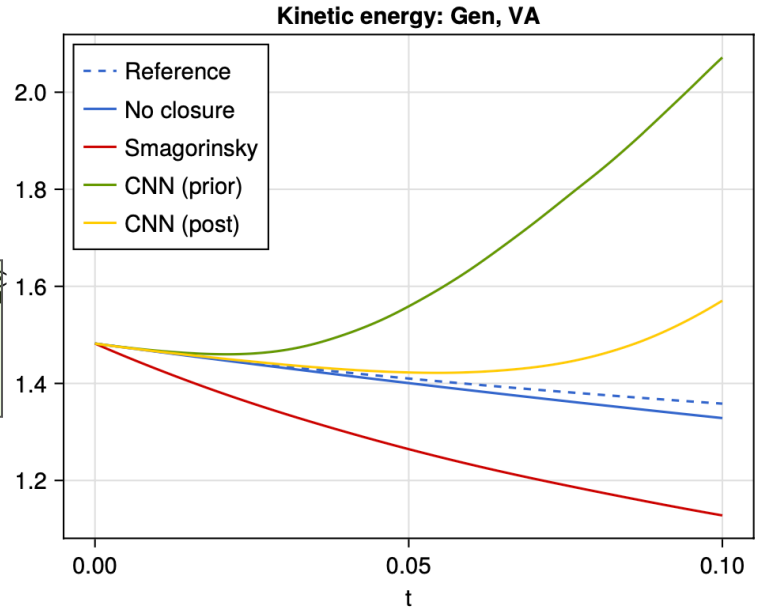




2. Discretization-consistent filters



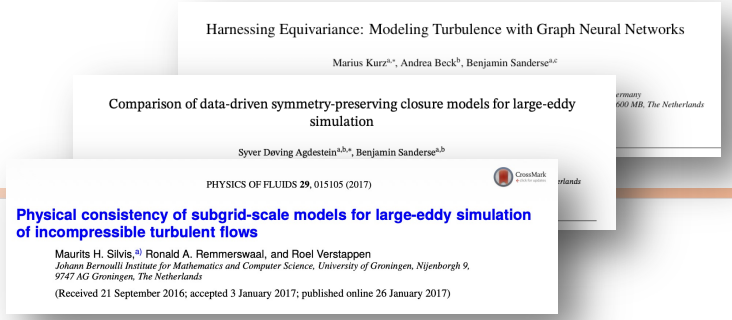
Stable results, even with a-priori training



Face-averaging filter

Standard (volume-averaging) filter

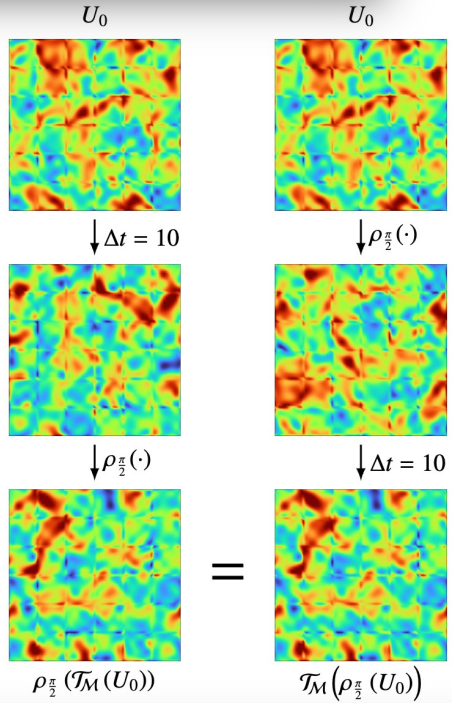
3. Change the neural network



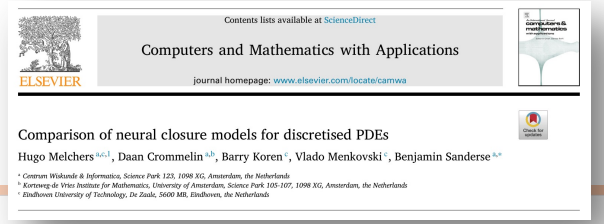
- **Symmetries** can be incorporated into NNs:
  - Group-equivariant CNNs
  - **Tensor-basis NNs**
  - Graph NNs

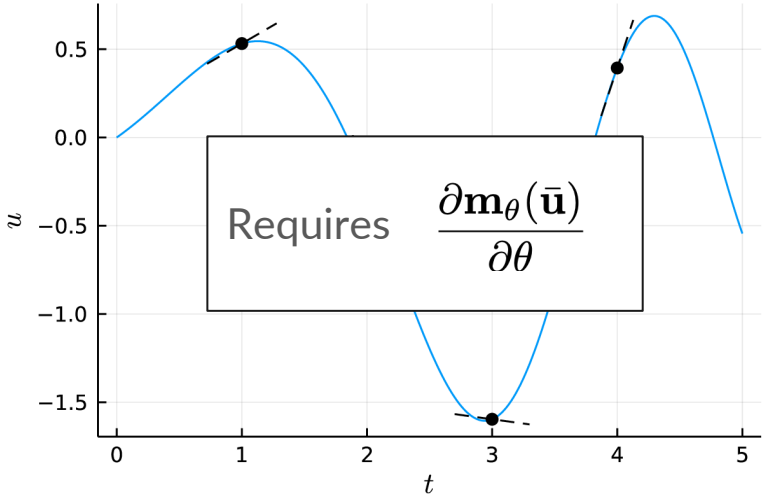
$$\tau^{\mathrm{TB}} = \Delta^2 \sum_k \alpha_k(\lambda)\mathbf{T}^{(k)}$$

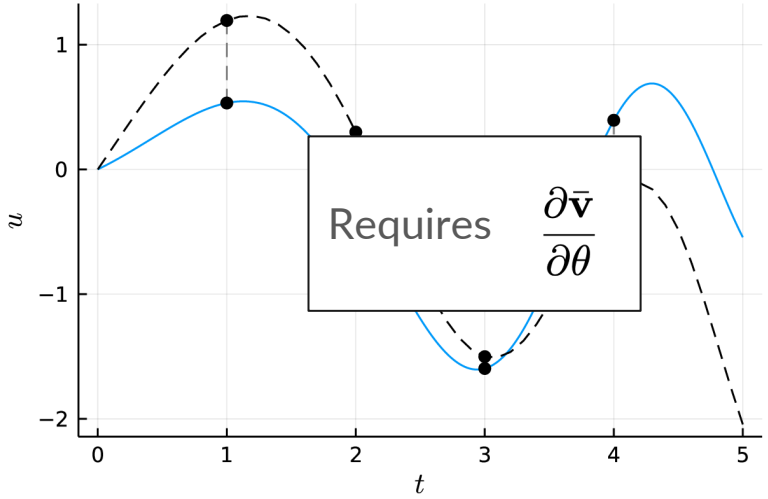
- **Many open questions**
  - Which symmetries are most important?
  - How to link symmetry to (numerical) stability?



4. Change the loss function







a-priori: derivative fitting

a-posteriori: trajectory fitting

$$\text{Loss} = \left\| \overline{\mathcal{F}(\mathbf{u})} - \mathcal{F}(\bar{\mathbf{u}}) - \mathbf{m}_\theta(\bar{\mathbf{u}}) \right\|^2$$

$$\text{Loss} = \sum_{i=1}^{N_t} \|\bar{\mathbf{v}}(t_i) - \bar{\mathbf{u}}(t_i)\|^2, \text{ where } \frac{\mathrm{d}\bar{\mathbf{v}}}{\mathrm{d}t} = \mathbf{f}(\bar{\mathbf{v}}) + \mathbf{m}_\theta(\bar{\mathbf{v}})$$

4. Discretization-consistent loss function


Exact expressions for the unresolved stress in a finite-volume based large-eddy simulation

Syver Døving Agdestein[a,b,*], Roel Verstappen[c], Benjamin Sanderse[a,b]

[a] Scientific Computing Group, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, 1098 XG, The Netherlands
[b] Centre for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, PO Box 513, Eindhoven, 5600 MB, The Netherlands
[c] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen , Nijenborgh 9, Groningen, 9747 AG, The Netherlands

- Consistent training data requires precise knowledge of **discretization** and **filter**
- Discretization also induces a filter, or: **a grid filter induces a discretization**

$$\frac{u(x_{i+1}) - u(x_i)}{h} = \frac{1}{h}\int \partial_x u \,\mathrm{d}x =: \overline{\partial_x u}^h$$

- Through this viewpoint, FVM and LES *both feature a closure term*

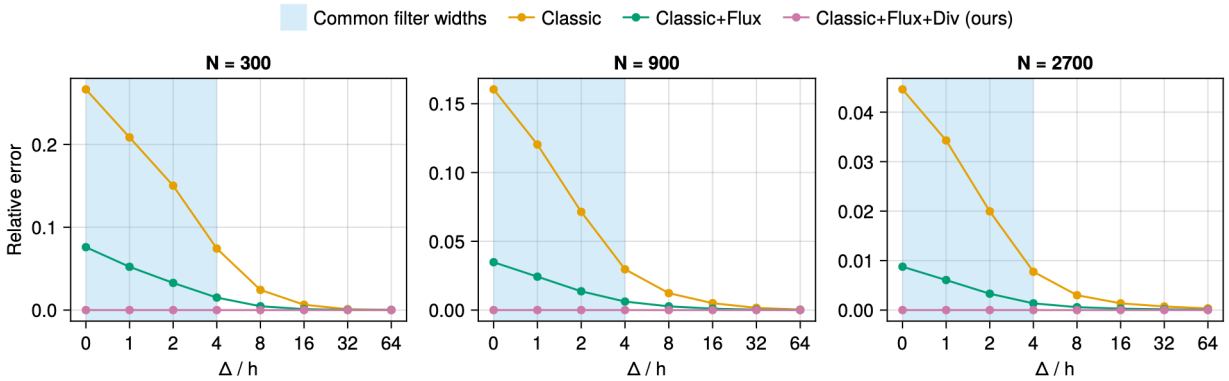
LES: $$\tau^h(\mathbf{u}, \bar{\mathbf{u}}) = \overline{\mathbf{c}(\mathbf{u})}^\Delta - \mathbf{c}(\bar{\mathbf{u}}^\Delta)$$

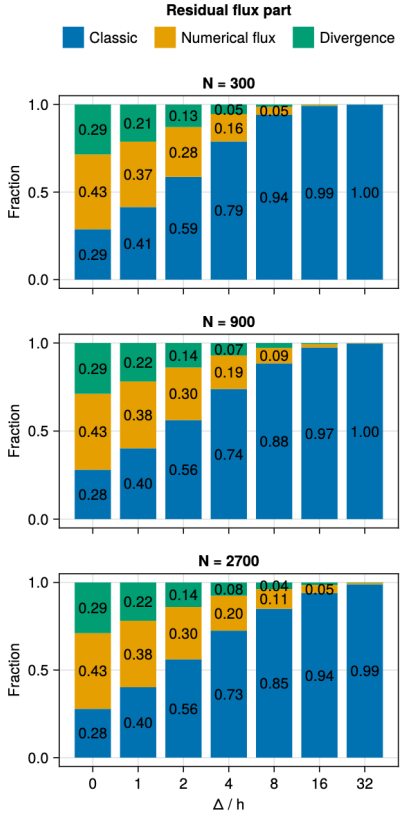
FVM: $$\tau^h(\mathbf{u}, \bar{\mathbf{u}}) = \mathbf{c}(\mathbf{u}) - \mathbf{c}^h(\bar{\mathbf{u}}^h)$$

FVM-LES: $$\tau^h(\mathbf{u}, \bar{\mathbf{u}}) = \overline{\mathbf{c}(\mathbf{u})}^\Delta - \mathbf{c}^h(\bar{\mathbf{u}}^{\Delta,h})$$ exact training target

4. Discretization-consistent loss function

- Discretization-consistent training target is **non-local & non-symmetric**
  - -> Important implications for closure model form
- Machine precision errors in DNS-aided LES

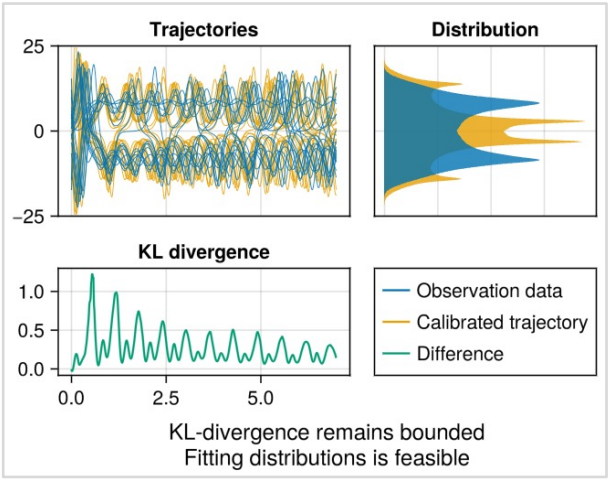


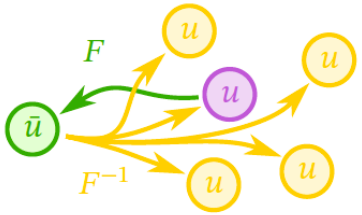


5. From ODEs to SDEs

Why probabilistic / stochastic?

- Model reduction leads to stochasticity (Mori-Zwanzig).
- Turbulence is chaotic. Interested in **statistics**.
- Effect of **non-uniqueness**.
- Uncertainty quantification.
- Regularize and stabilize.





5. From ODEs to SDEs

Ten questions concerning the large-eddy simulation of turbulent flows

Stephen B Pope
Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA
E-mail: pope@mae.cornell.edu

New Journal of Physics **6** (2004) 35
Received 3 December 2003
Published 16 March 2004
Online at http://www.njp.org/ (DOI: 10.1088/1367-2630/6/1/035)

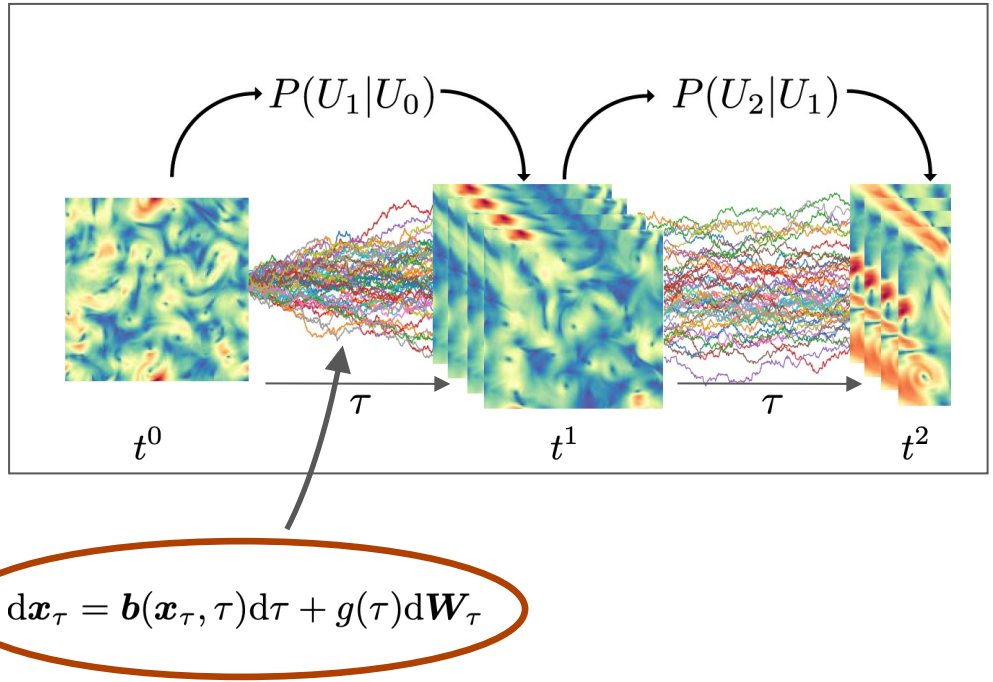
In the LES context, it should be appreciated that fundamentally $\boldsymbol{W}(\boldsymbol{x}, t)$ is defined as the solution to the LES equations, not as the spatially filtered value of $\boldsymbol{U}(\boldsymbol{x}, t)$, which we denote by $\overline{\boldsymbol{U}}(\boldsymbol{x}, t)$.

However, we may ask, is it possible (in principle) to have a perfect LES model such that $\boldsymbol{W}(\boldsymbol{x}, t)$ equals $\overline{\boldsymbol{U}}(\boldsymbol{x}, t)$? The answer is *no*; the reason being that $\overline{\boldsymbol{U}}(\boldsymbol{x}, t)$ is a random field, whose future evolution is not determined by its current state. Thus, while we may impose $\boldsymbol{W}(\boldsymbol{x}, 0) = \overline{\boldsymbol{U}}(\boldsymbol{x}, 0)$ as an initial condition, for $t > 0$, $\overline{\boldsymbol{U}}(\boldsymbol{x}, t)$ has a statistical distribution, and hence there is no value of $\boldsymbol{W}(\boldsymbol{x}, t)$ which equals $\overline{\boldsymbol{U}}(\boldsymbol{x}, t)$. This argument is developed more fully in section 13.5.6 of Pope [5].

Stochastic forecasting through generative models

- Sample **ensemble**
- Each time step, solve an SDE in pseudo-time
- Need to find drift + diffusion

Generative modeling
=
Finding SDEs

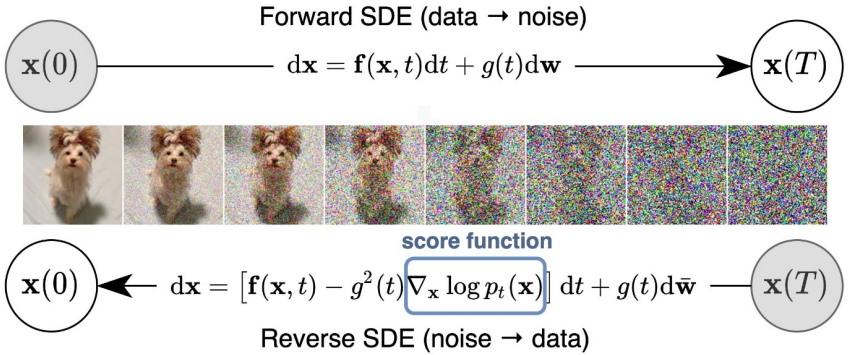


Denoising diffusion probabilistic models (DDPMs)

- Use SDE toolbox from Gen. AI!
- **“Score matching” [1]:**
  Learn SDE to generate images from noise

Issue: DDPMs are not physical

- Maps back and forth to Gaussians
- Infinite-time process
- ***f*** and ***g*** are given; score function needs to be learned



[1] Song, Yang, and Ermon. "Generative modeling by estimating gradients of the data distribution." Advances in neural information processing systems 32 (2019).

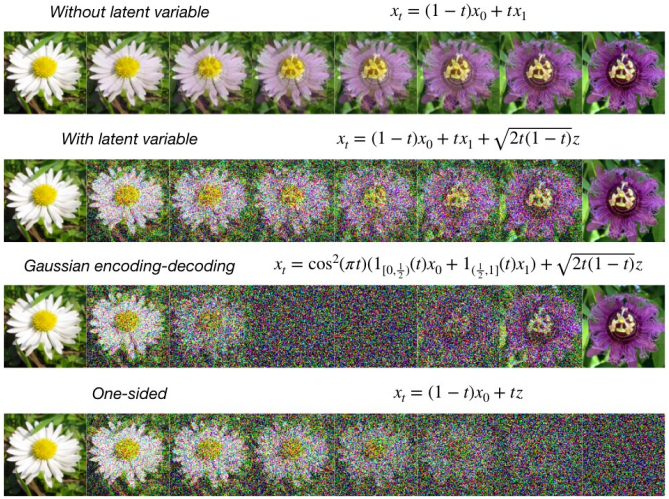
Stochastic interpolants (SI [1])

Stochastic interpolants:

- Interpolate between images
- Related to **flow matching** [2]

Advantages:

- Base distribution can be “**anything**”
- Reach the target distribution in **finite time**
- Diffusion term can be chosen **after training**



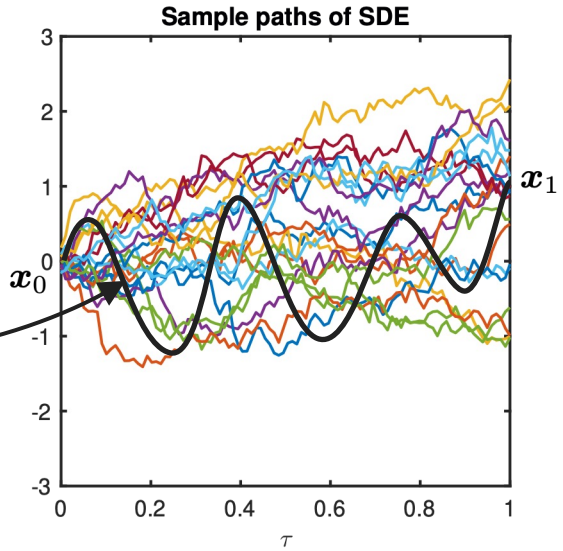
[1] Albergo, Michael S., Nicholas M. Boffi, and Eric Vanden-Eijnden. "Stochastic interpolants: A unifying framework for flows and diffusions." arXiv:2303.08797 (2023).
[2] Y. Lipman, M. Tancik, and J. Lu, Flow matching for generative modelling, arXiv:2210.02747, 2022.

The stochastic interpolant

- Define an interpolant between base and target distributions

Stochastic interpolant

$$\boldsymbol{x}_\tau \approx \boldsymbol{I}_\tau(\boldsymbol{x}_0, \boldsymbol{x}_1) = \alpha_\tau \boldsymbol{x}_0 + \beta_\tau \boldsymbol{x}_1 + \gamma_\tau \boldsymbol{W}_\tau$$



- The drift is found by minimizing the loss function:

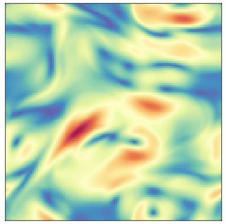
$$L(\theta) = \int_0^1 \mathbb{E}\left[|\boldsymbol{b}_\theta(\boldsymbol{I}_\tau(\boldsymbol{x}_0, \boldsymbol{x}_1), \tau) - \boldsymbol{R}_\tau(\boldsymbol{x}_0, \boldsymbol{x}_1)|^2\right] \mathrm{d}\tau$$

Neural network

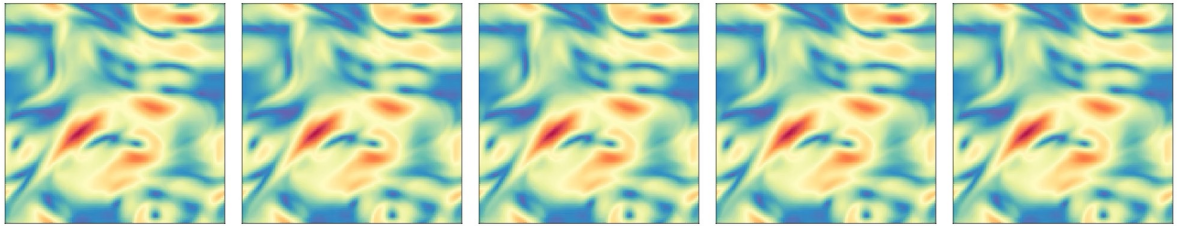
Interpolant derivative

$$\alpha(0) = 1, \quad \alpha(1) = 0$$
$$\beta(0) = 0, \quad \beta(1) = 1$$
$$\boldsymbol{W}_\tau = \sqrt{\tau}\boldsymbol{z}, \quad \boldsymbol{z} \sim N(0, 1)$$

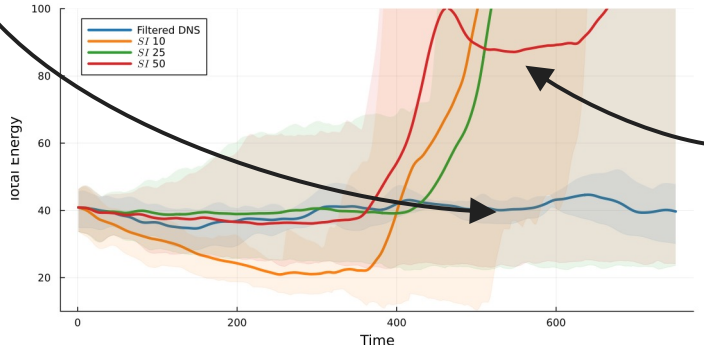
Vanilla stochastic interpolant results



Ground truth
(filtered DNS)

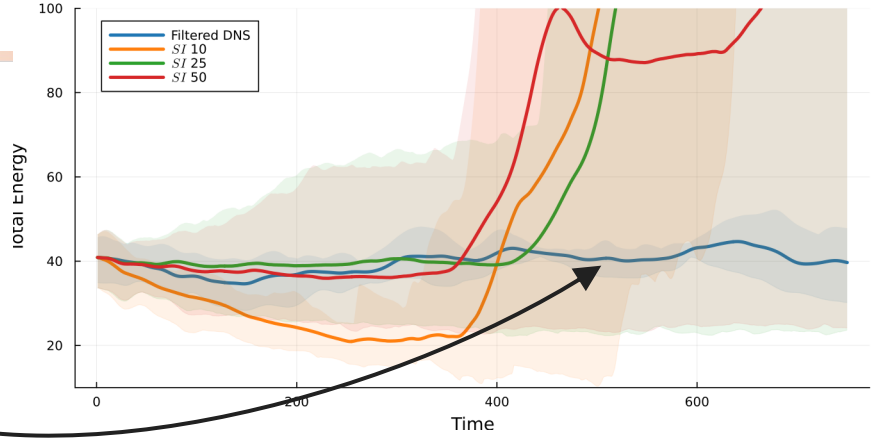


Example realizations



Vanilla SIs have **unphysical energy behavior**

Structure-preserving SI



- Should not enforce energy stability on individual trajectories
  - Flow only statistically energy stable
- **Idea**: tune the α_τ, β_τ coefficients for energy consistency:

$$L_{\text{energy}}(\alpha_\tau, \beta_\tau) = \int_0^1 \mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_1)} \left[\mathrm{d} \left[\frac{1}{2} ||\boldsymbol{I}_\tau(\boldsymbol{x}_0, \boldsymbol{x}_1)||_2^2 \right] - k_\tau(\boldsymbol{x}_0, \boldsymbol{x}_1) \right] \mathrm{d}\tau$$

Need Ito's lemma

Desired rate of change

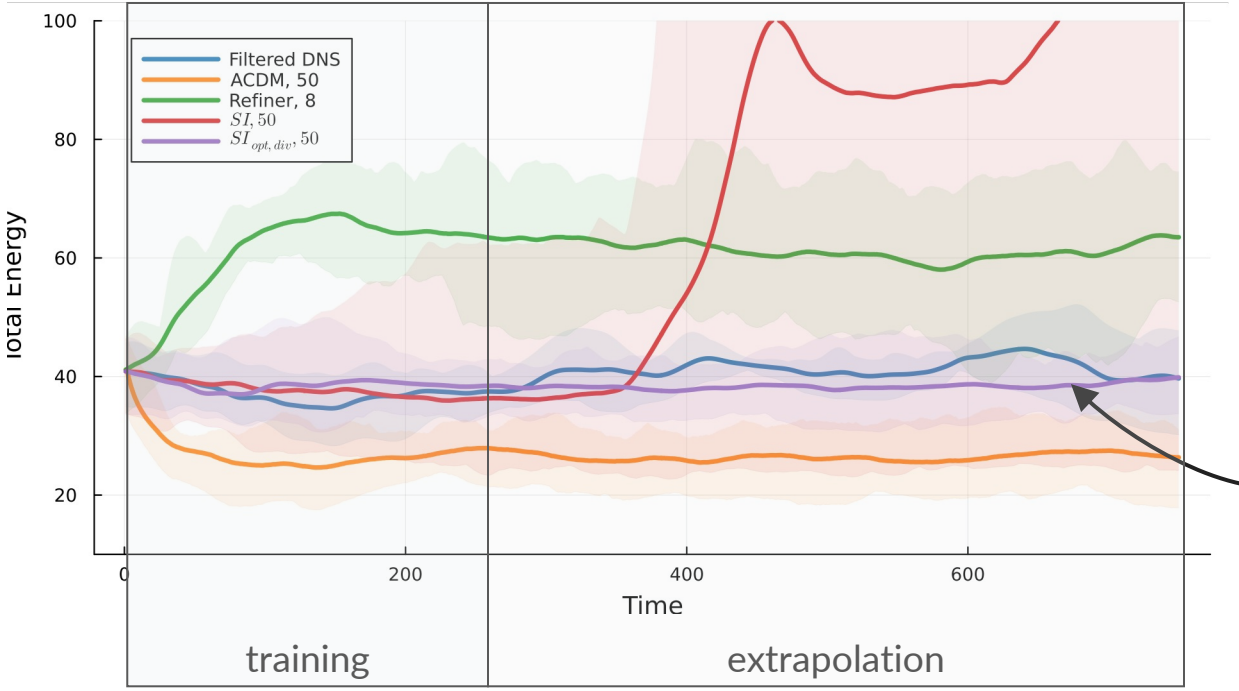
Optimized stochastic interpolant results

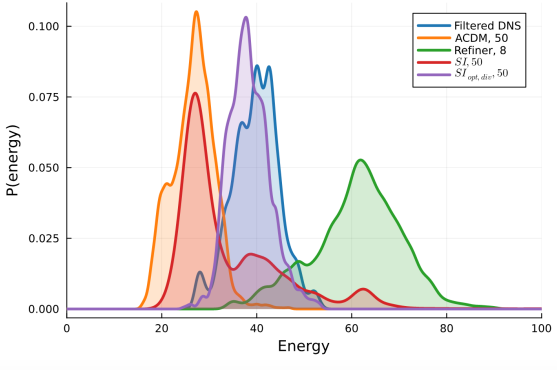

Physics-aware generative models for turbulent fluid flows through energy-consistent stochastic interpolants

Nikolaj T. Mücke[a], Benjamin Sanderse[a,b]

[a] Scientific Computing, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, 1098 XG, The Netherlands
[b] Centre for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, PO Box 513, Eindhoven, 5600 MB, The Netherlands

25





Significant improvement in energy behavior

Conclusions

- **Stable and accurate** closure models *without a-posteriori or reinforcement learning*
  1. Auxiliary variables (energy stability)
  2. Discretization-consistency: filter and loss function
  3. Stochastic models – use gen. ML tools - towards probabilistic LES

- Preserving **symmetries** (equivariance) helps in generalization and reducing required training data

- Read Pope's work

Outlook

- Generalizability – high Reynolds numbers, where there is no DNS training data
- Online model calibration (data assimilation) – experimental data