SHAPING TOMORROW

ISC High Performance

ISC 2020 DIGITAL

JUNE 22-25

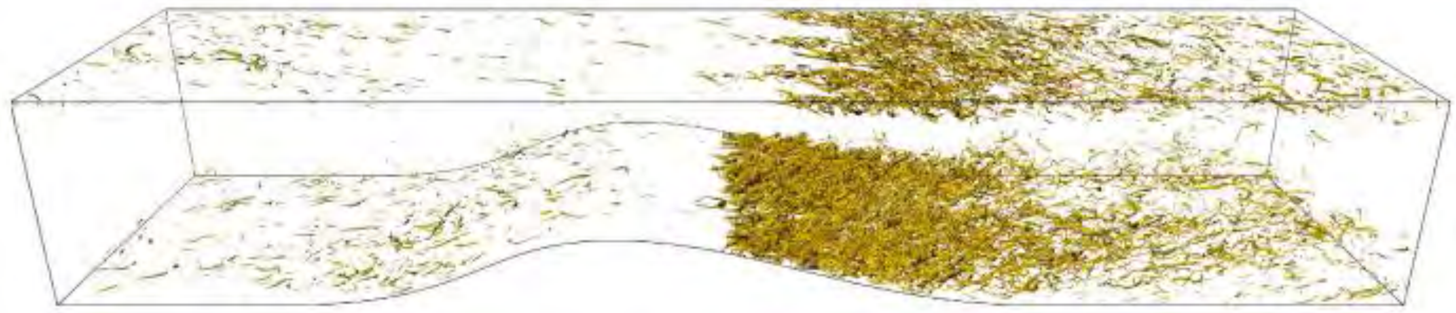
#ISC20

Parameter Identification of RANS turbulence model using Physics-embedded neural network

Shirui Luo (shirui@illinois.edu)

Postdoc at UIUC

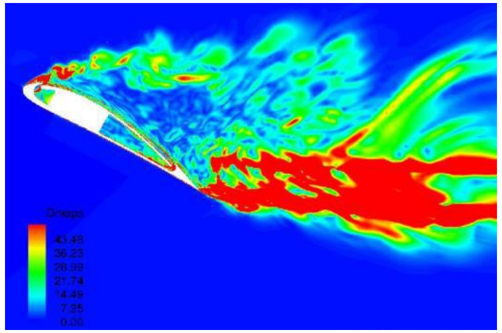
06/20/2020

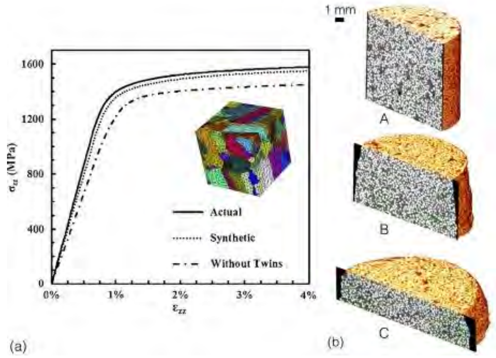


Confluence of Machine Learning with CFD

Scientific Modeling Applications

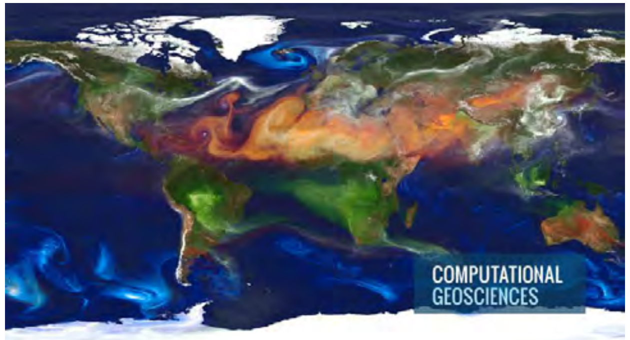
- Fluid turbulence modeling;
- Heterogeneous materials constitutive modeling;
- Dynamics of atmospheric, ocean, and climate system;
- Combustion/Chemical reaction;
- Galaxies formation.

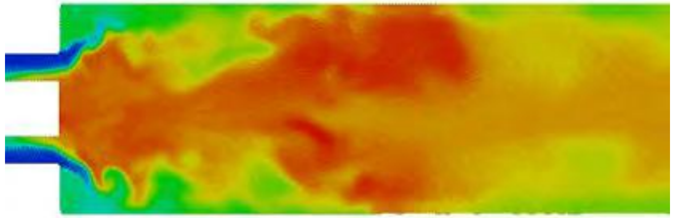




Reasons for using machine learning in CFD

- Do not understand the physics well enough;
- Cannot afford the computational cost to adequately resolve the full-scale physics;
- only calculate details of the large scales while parameterizing the effects of small scales on these explicitly retained large scales statistically.





Turbulence model for Reynolds-averaged Navier–Stokes

Unclosed term

$$\frac{\partial u_i}{\partial t}+\frac{\partial}{\partial x_j}(u_i u_j)=-\frac{\partial p}{\partial x_i}+\nu\frac{\partial^2 u_i}{\partial x_j \partial x_j}$$

$$\frac{\partial u_i}{\partial x_j}=0$$

Reynolds decomposition $\longrightarrow$

$$u_i=\bar{u}_i+u_i',\quad p=\bar{p}+p'$$

$$\frac{\partial \bar{u}_i}{\partial t}+\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j}=-\frac{\partial \bar{p}}{\partial x_i}+\nu\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j}-\frac{\boxed{\partial \tau_{ij}}}{\partial x_j}$$

$$\frac{\partial \bar{u}_i}{\partial x_j}=0$$

The transport equations of turbulent kinetic energy and dissipation rate for κ-ε model: $\tau_{ij}=\overline{u_i'u_j'}\approx\frac{2}{3}\kappa\delta_{ij}-2\nu_T\overline{S_{ij}}$

$$\frac{\partial \kappa}{\partial t}+\bar{u}_i\frac{\partial \kappa}{\partial x_i}=-\tau_{ij}\frac{\partial \bar{u}_i}{\partial x_j}-\varepsilon+\frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\underline{\sigma_K}}\frac{\partial \kappa}{\partial x_i}\right)+\nu\frac{\partial^2 \kappa}{\partial x_i \partial x_i}\qquad \nu_T=\underline{C_\mu}\frac{\kappa^2}{\varepsilon}$$

$$\frac{\partial \varepsilon}{\partial t}+\bar{u}_i\frac{\partial \varepsilon}{\partial x_i}=-\underline{C_{\varepsilon 1}}\frac{\varepsilon}{\kappa}\tau_{ij}\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\underline{\sigma_\varepsilon}}\frac{\partial \varepsilon}{\partial x_i}\right)-\underline{C_{\varepsilon 2}}\frac{\varepsilon^2}{\kappa}+\nu\frac{\partial^2 \varepsilon}{\partial x_i \partial x_i}$$

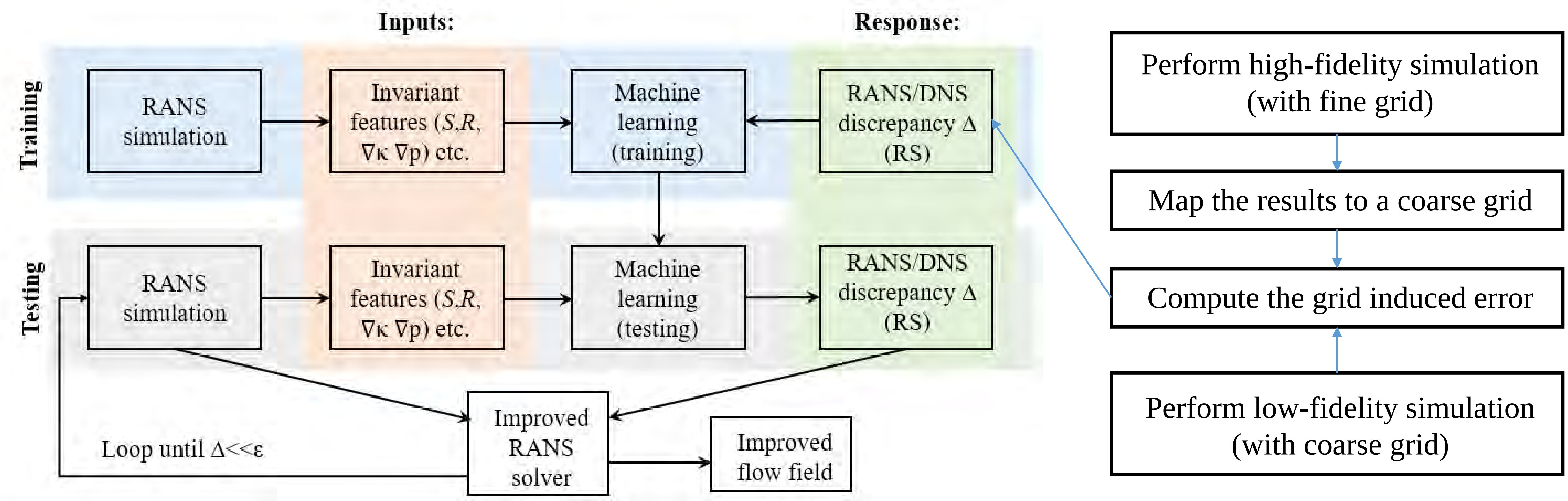
Five tunable parameters: $\boxed{C_{\varepsilon 1}, C_{\varepsilon 2}, C_\mu, \sigma_\kappa, \sigma_\varepsilon}$

various κ-ε turbulence models constants

	Launder & Sharma	Jones & Launder	Chien	Yakhot & Orszag
$C_{\varepsilon 1}$	1.44	1.55	1.35	1.063
$C_{\varepsilon 2}$	1.92	2.0	1.8	1.7215
C_μ	0.09	0.09	0.09	0.0837
σ_k	1.0	1.0	1.0	0.7179
σ_ε	1.3	1.3	1.3	0.7179

Machine Learning RANS turbulence modeling

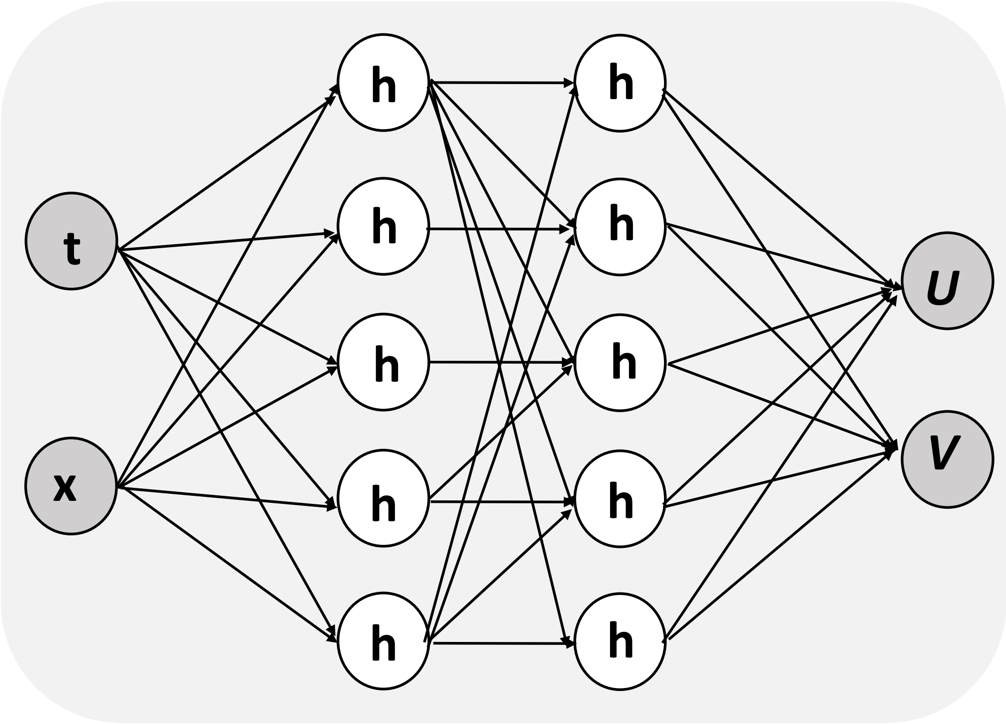
Purely data driven: To build a neural network to model the unclosed term using other mean flow quantities



Drawback : 1) Poor generalization; 2) requires huge amount dataset.

Physics embedded neural network

Feedforward neural network



input nodes *hidden nodes* *output nodes*

Drawbacks of pure data-driven methods in scientific applications:

- Poor generalization due to lack of physics;
- For example, knowledge learned from training data can not be applied on testing cases with a different flow conditions, even the flow equations are the same;
- Requires training sets with sufficiently rich variability to enable extrapolation to new configurations.

Incompressible Navier-Stokes equation:

Conservation of mass: $$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

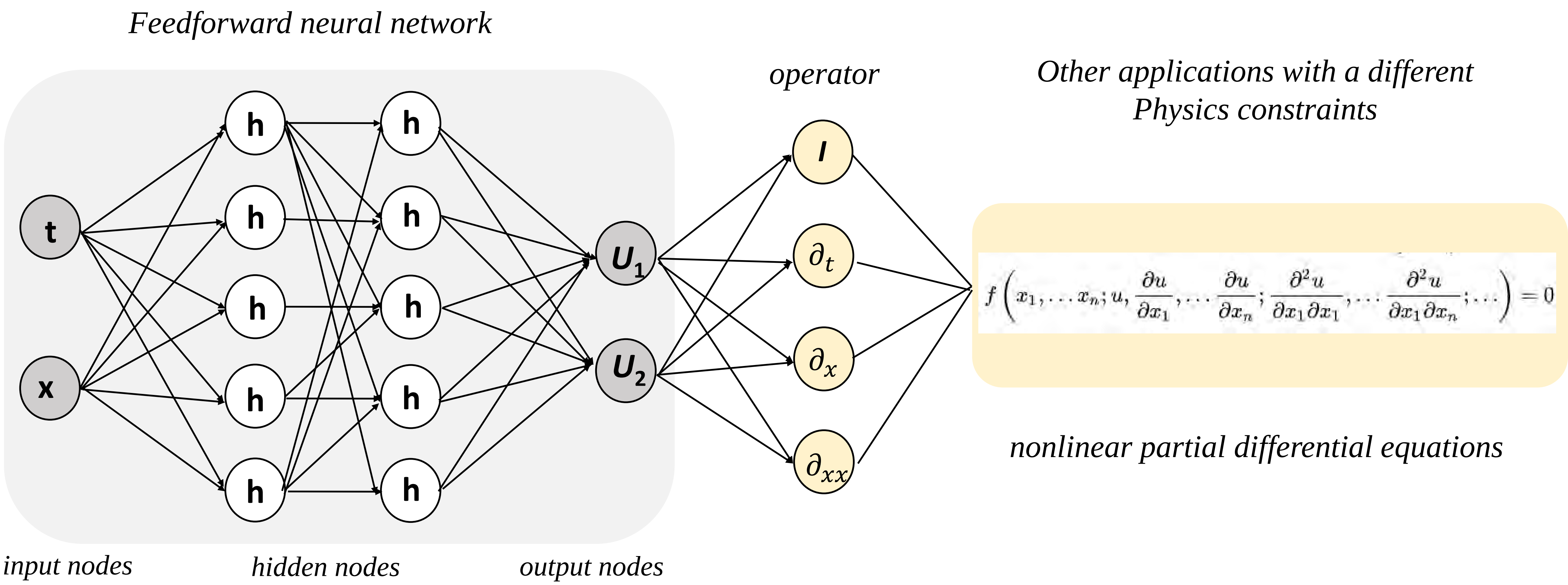
Introduces a **regularization mechanism** using physical laws, penalize overfitted solution which do not satisfy physical equations.

Total loss

$$L = \sum (U^{real} - U^{pred})^2 + \sum (V^{real} - V^{pred})^2 + \boldsymbol{w_f} * \left\| \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right\|$$

Mean Squared Error from modeling

Physics embeded neural network



regularization

Total loss

$$L = \sum\left(U_1^{real} - U_1^{pred}\right)^2 + \sum\left(U_2^{real} - U_2^{pred}\right)^2 + \boldsymbol{w_f} * \|\boldsymbol{f}\|$$

Error from modeling *Error from physics*

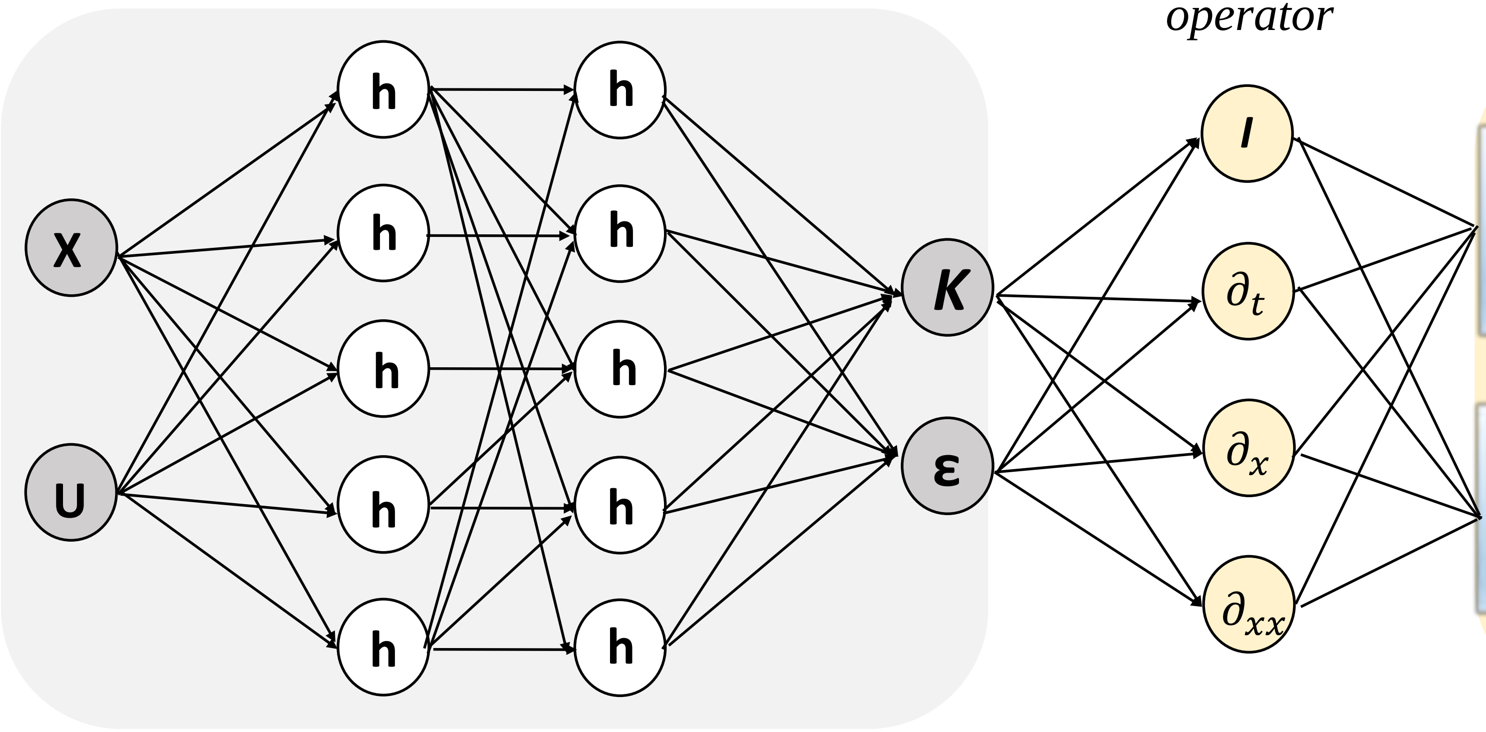
Physics embedded neural network

RANS turbulence modeling

Feedforward neural network

Fluid physics constraints: two transport equations

operator



$$\boldsymbol{f}: \frac{\partial K}{\partial t} + \bar{u}_i \frac{\partial K}{\partial x_i} + \tau_{ij}\frac{\partial \bar{u}_i}{\partial x_j} + \varepsilon - \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_K}\frac{\partial K}{\partial x_i}\right) - \nu\frac{\partial^2 K}{\partial x_i \partial x_i}$$

$$\boldsymbol{g}: \frac{\partial \varepsilon}{\partial t} + \bar{u}_i \frac{\partial \varepsilon}{\partial x_i} + C_{\varepsilon 1}\frac{\varepsilon}{K}\tau_{ij}\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_{\varepsilon 2}\frac{\varepsilon^2}{K} - \nu\frac{\partial^2 \varepsilon}{\partial x_i \partial x_i}$$

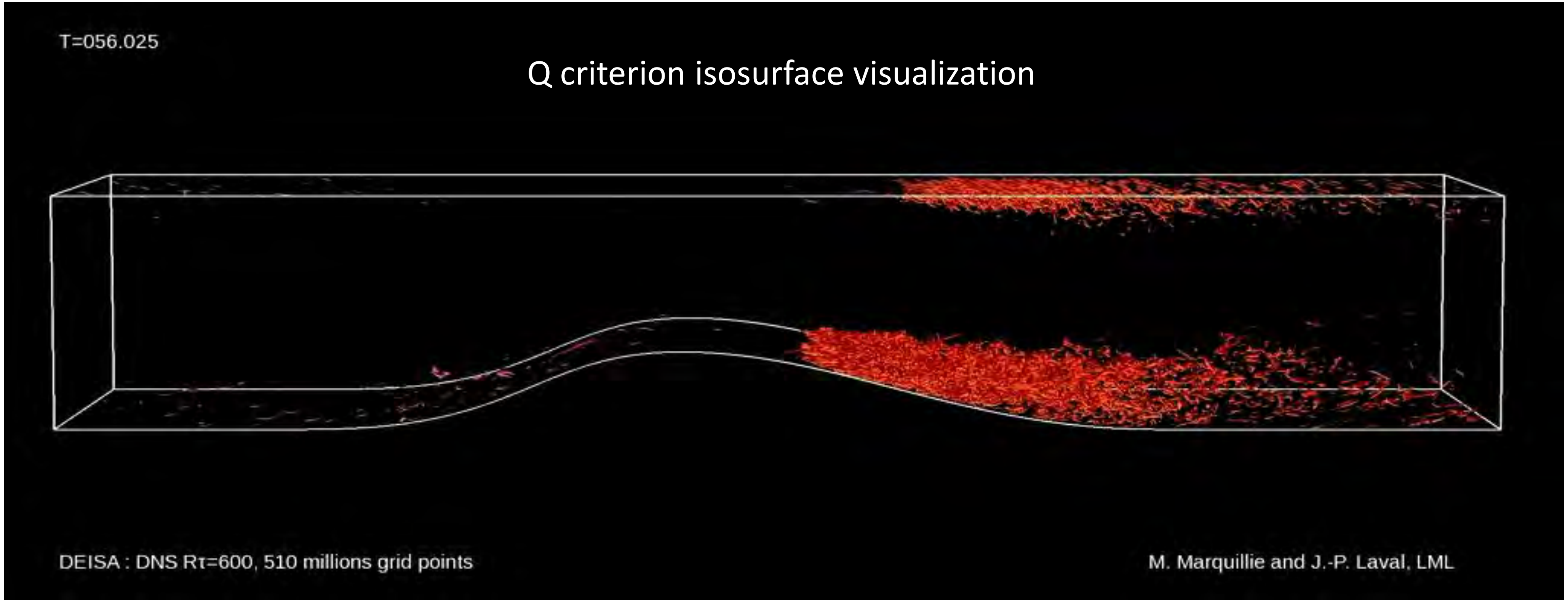
input nodes *hidden nodes* *output nodes*

Five tunable parameters are tf. variable

Total loss $L = \sum\left(K^{real} - K^{pred}\right)^2 + \sum\left(\varepsilon^{real} - \varepsilon^{pred}\right)^2 + \boldsymbol{w_f} * \|\boldsymbol{f}\| + \boldsymbol{w_g} * \|\boldsymbol{g}\|$

Error from modeling *Error from physics*

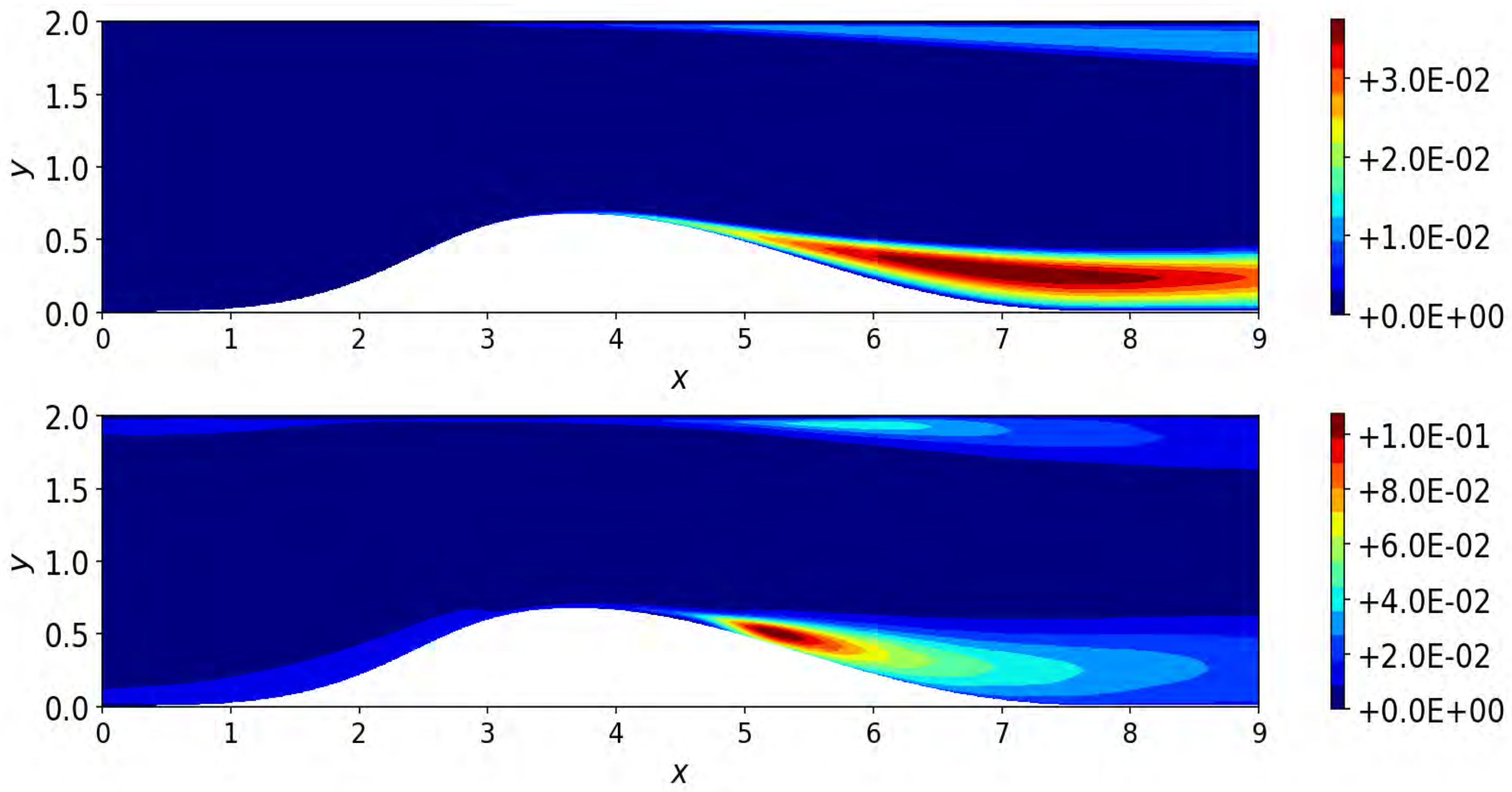
Case Study: Channel flow with a lower curved wall



Direct Numerical Simulation of converging-diverging channel flow   Mesh 1536 X 257 X 384

- DNS were conducted using the numerical code MFLOPS3D developed at LML(Lille Mechanic Laboratory);
- Significant progress in the understanding and modelling of near wall turbulence in Boundary Layer;
- Current database was generated in order to provide data to test and validate turbulence models.

Case Study: Channel flow with a lower curved wall



A comparison of the contour plots of the time averaged turbulent kinetic energy (top) RANS (bottom) DNS when Re_τ=395.

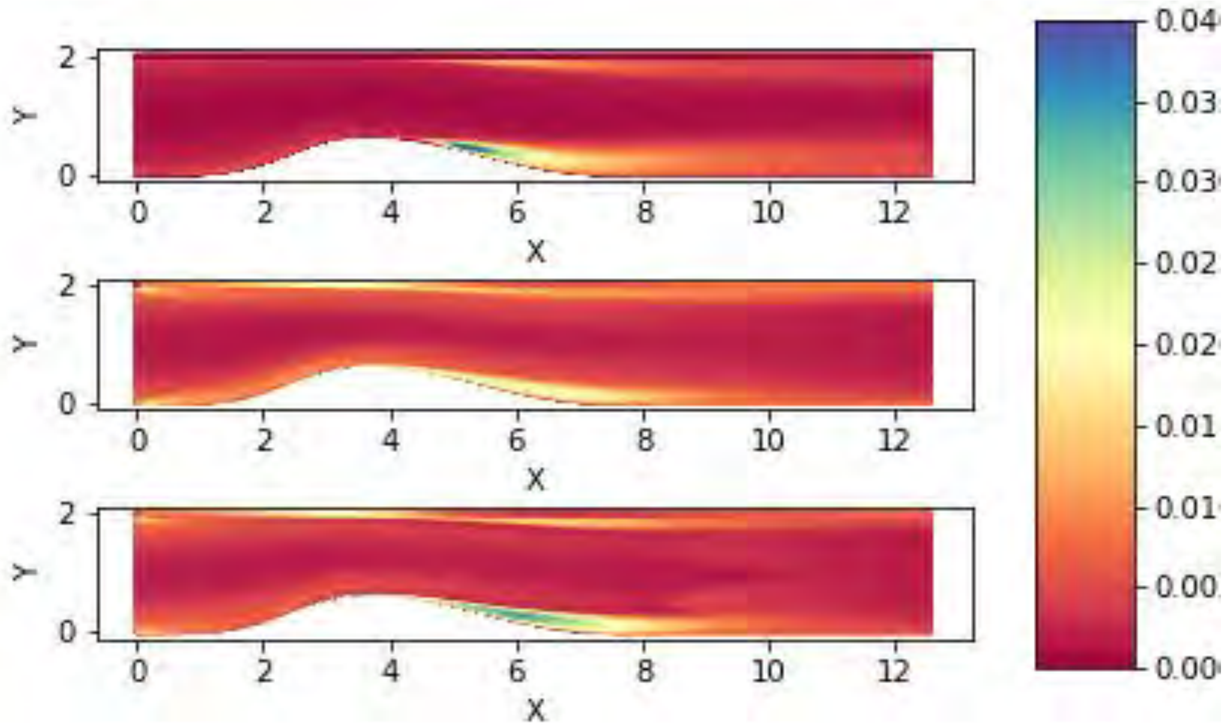
k-ε Equation Physics-embedded neural network

Table 1. Comparison of the inferred constants from neural network

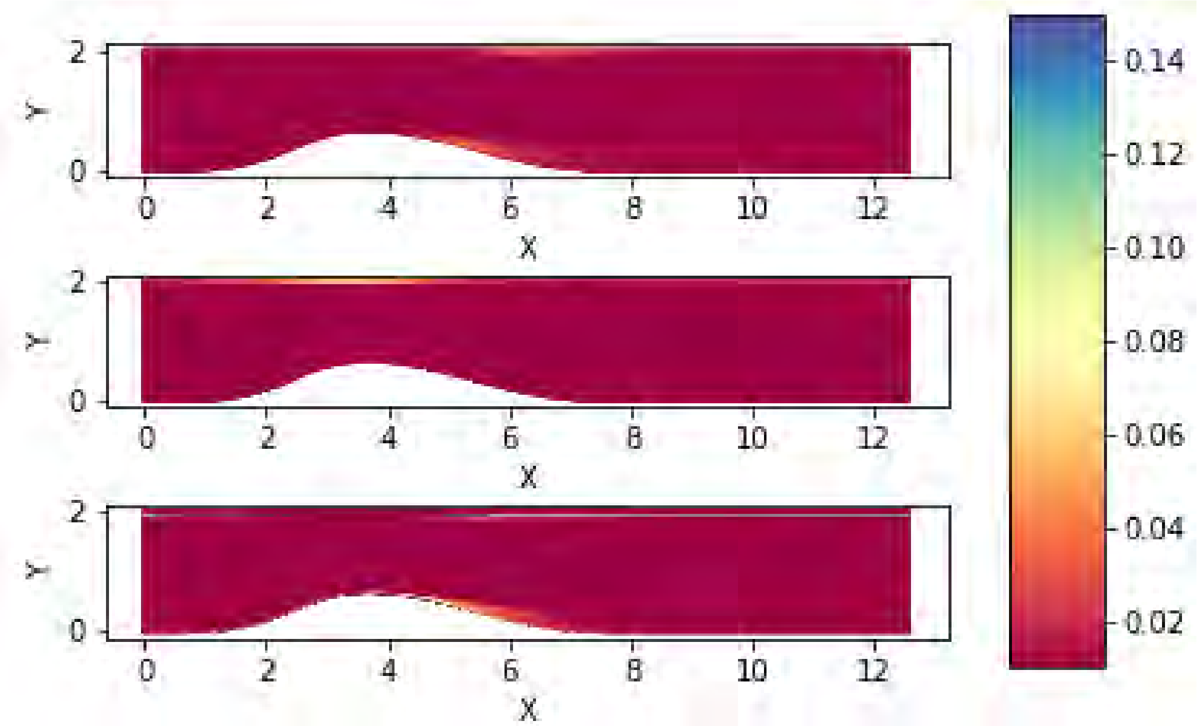
Five constant	Empirical (**Default**)	Empirical (RNG theory)	Jones & Launder	Chien	NN-pred Fix C_μ
$C_{\varepsilon 1}$	1.44	1.063	1.55	1.35	1.302
$C_{\varepsilon 2}$	1.92	1.7215	2.0	1.80	1.862
C_μ	0.09	0.0837	0.09	0.09	**0.09**
σ_κ	1.0	0.7179	1.0	1.0	0.751
σ_ε	1.3	0.7179	1.3	1.3	0.273

- The **Default** are widely used in popular software like Ansys Fluent or OpenFOAM;
- Previously these constants were fitted empirically by limited experimental data, not working well in some unseen cases.

Results and Discussion

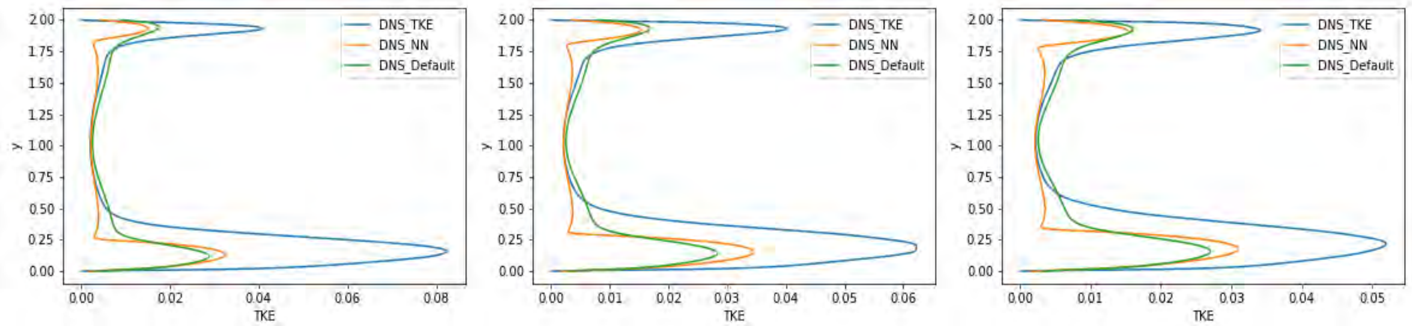


Comparison of the time averaged turbulent kinetic energy from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

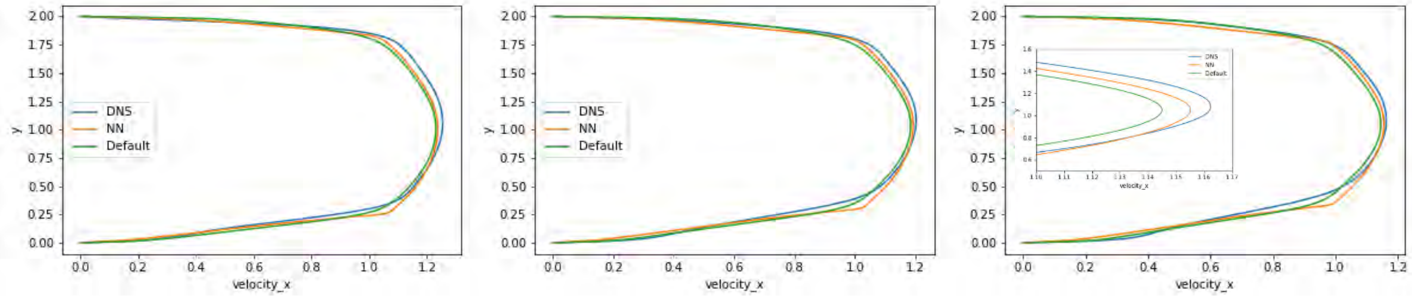


Comparison of the time averaged dissipation rate from different simulations (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.

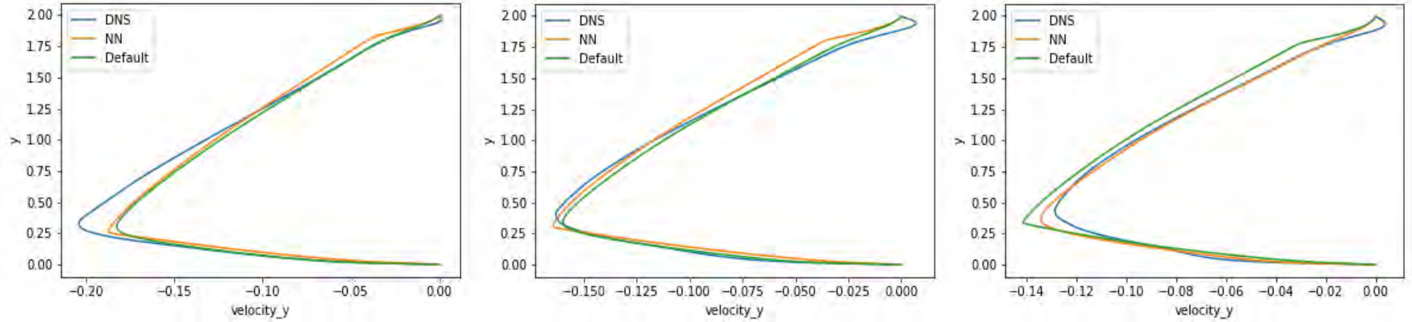
Results and Discussion



Plots of the turbulent kinetic energy along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).

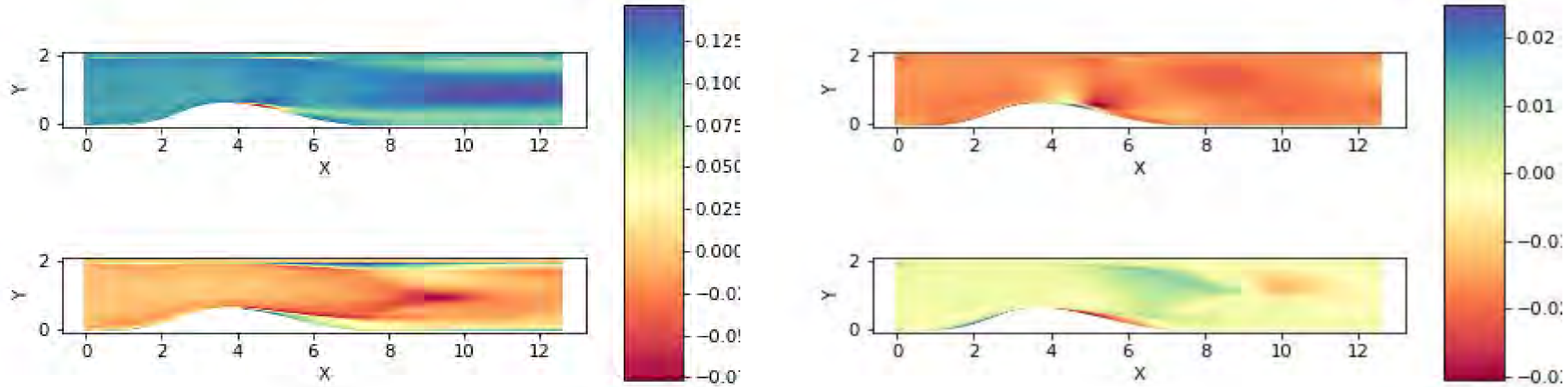


Plots of the x-velocity along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).



Plots of the y-velocity along y axis when x = 5.7306 (left), x = 6.1399 (middle), x = 6.5493 (right).

Results: a posteriori



- Comparison of the error contour plots between RANS and DNS (left is the x-velocity, right is the y-velocity). (top) error between DNS-Default; (bottom) error between DNS-PINNs,
- The error in the bottom is relatively much smaller than the error in the top.

Conclusion

- Used a physics informed neural network (PINN) that is embedded with the turbulent transport equations;
- Physical loss functions are proposed to explicitly impose information of the transport equations to deep learning networks;
- An inverse problem by treating the five parameters in turbulence model as random variables;
- Validated this method on two cases of flow over bump with recommended parameters;
- The mean absolute error of the velocity profile between RANS and DNS decreased by 22% when used the neural network inferred parameters.

Future

- The error between RANS and DNS is still noteworthy;
- Inaccuracy is due to rans's inherent simplifications rather than the inappropriate use of RANS constants;
- Switch to a more sophisticated turbulence model.

Thank you!

Shirui Luo (shirui@illinois.edu)

Postdoc at UIUC

06/20/2020

SPONSORS

Follow us on Twitter at #ISC20 !

ISC 2020 DIGITAL MEDIA SPONSORS

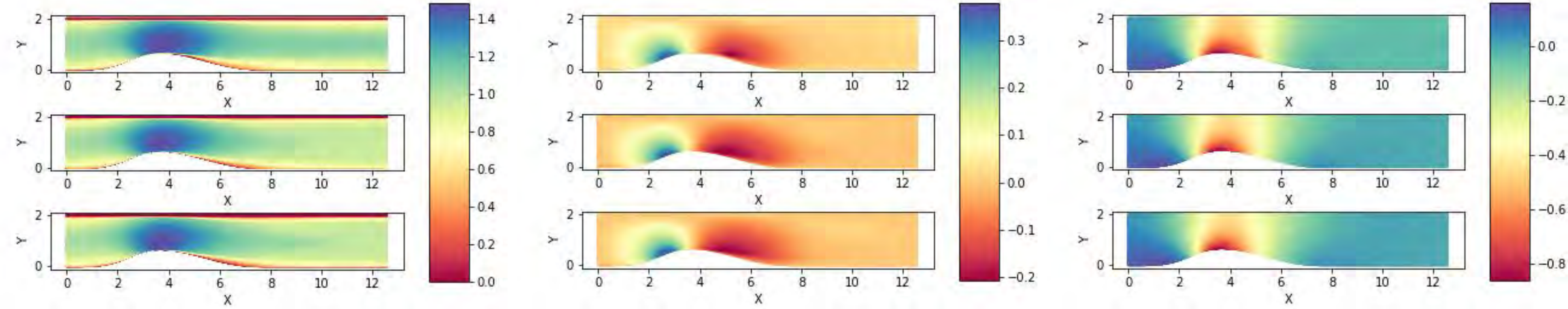
ISC High Performance
SHAPING TOMORROW

- ADMIN Network & Security
- ADMIN Enterprise HPC
- EnterpriseAI – News & Insights for the AI Journey
- HPCwire
- insideHPC
- Linux Magazine
- Primeur Magazine
- Science Node
- Scientific Computing World
- The Next Platform
- The Register®

Follow us on Twitter at #ISC20 !

Appendix

Comparison of the different simulation results (top) DNS, (middle) Default RANS, (bottom) PINNs RANS.



(left)time averaged x-velocity; (middle) time averaged x-velocity; (right) time averaged pressure