FAKULTA APLIKOVANÝCH VĚD ZÁPADOČESKÉ UNIVERZITY V PLZNI

VZÚ PLZEŇ

www.vzuplzen.cz

Employing Nonlinear Transformation of Datasets to Train Neural Networks

Luboš Smolík[1,2], Jan Rendl[2], Radek Bulín[2]

1 – Výzkumný a zkušební ústav v Plzni (Research and Testing Institute Plzen), Diagnostics and Condition Monitoring Team

2 – University of West Bohemia, Faculty of Applied Sciences, NTIS Research Centre

PWP-47/24/009

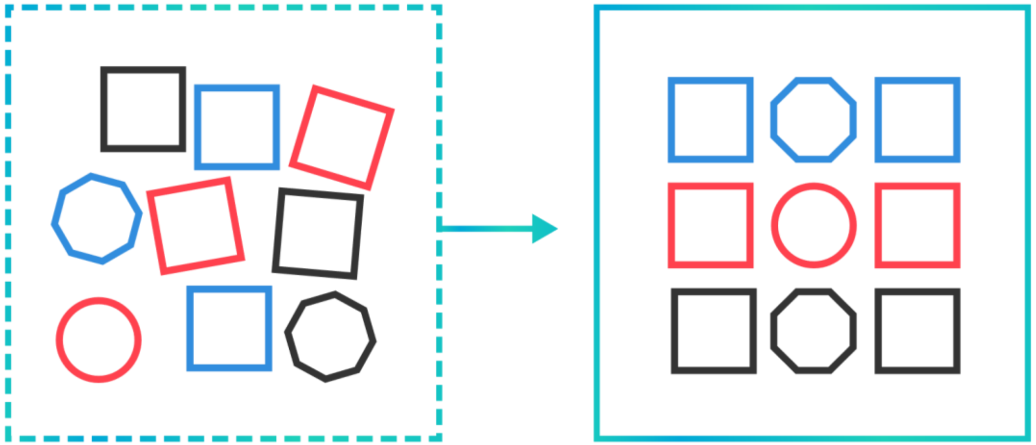
Outline

i. Introduction – Nonlinear Transformations

ii. Co-Simulation in Dynamical Systems

iii. Application of Nonlinear Scaling
   - Hydrodynamic lubrication in journal bearings
   - Training
   - Results & Discussion

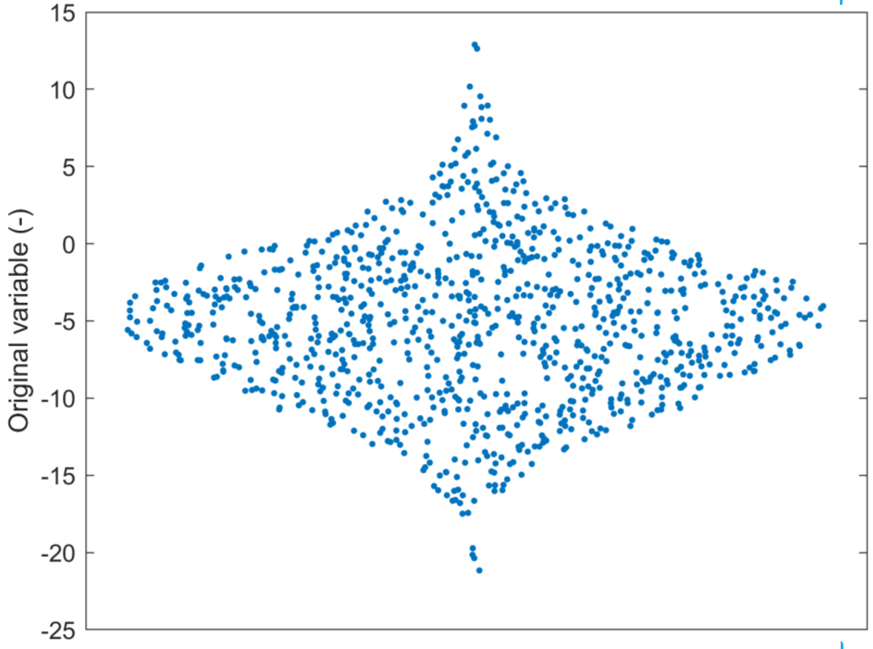
iv. Conclusions



Section i.

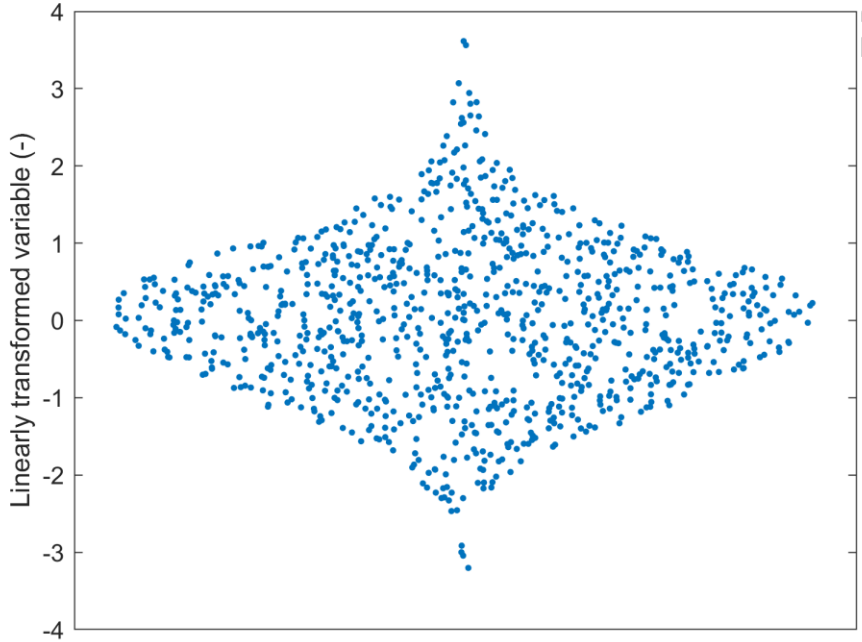
Introduction

Data transformation



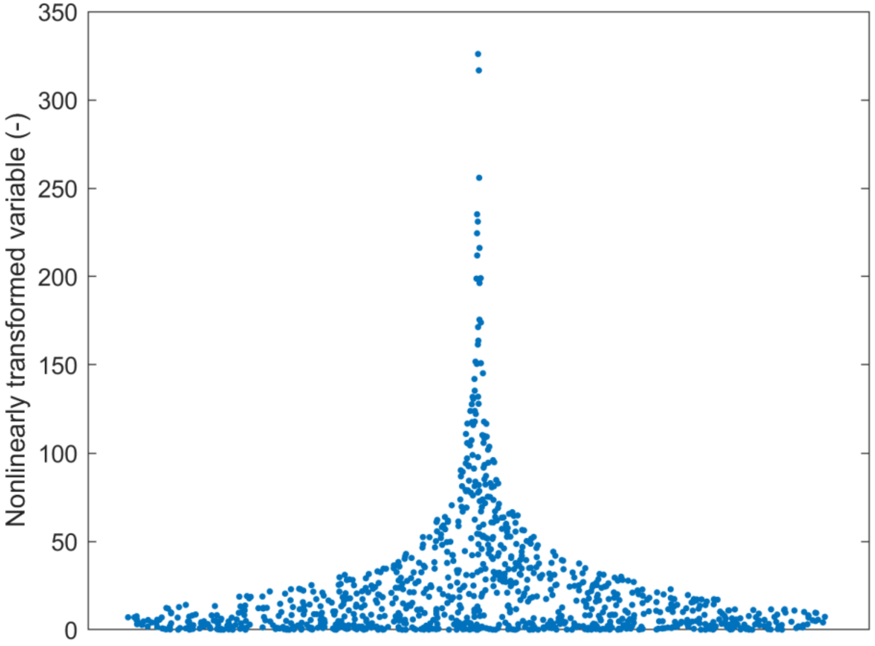
Linear

$$y_i = \frac{x_i - \mathrm{mean}(x)}{\mathrm{std}(x)}$$



Nonlinear

E.g: $z_i = (x_i - \mathrm{mean}(x))^2$



Why transform data?

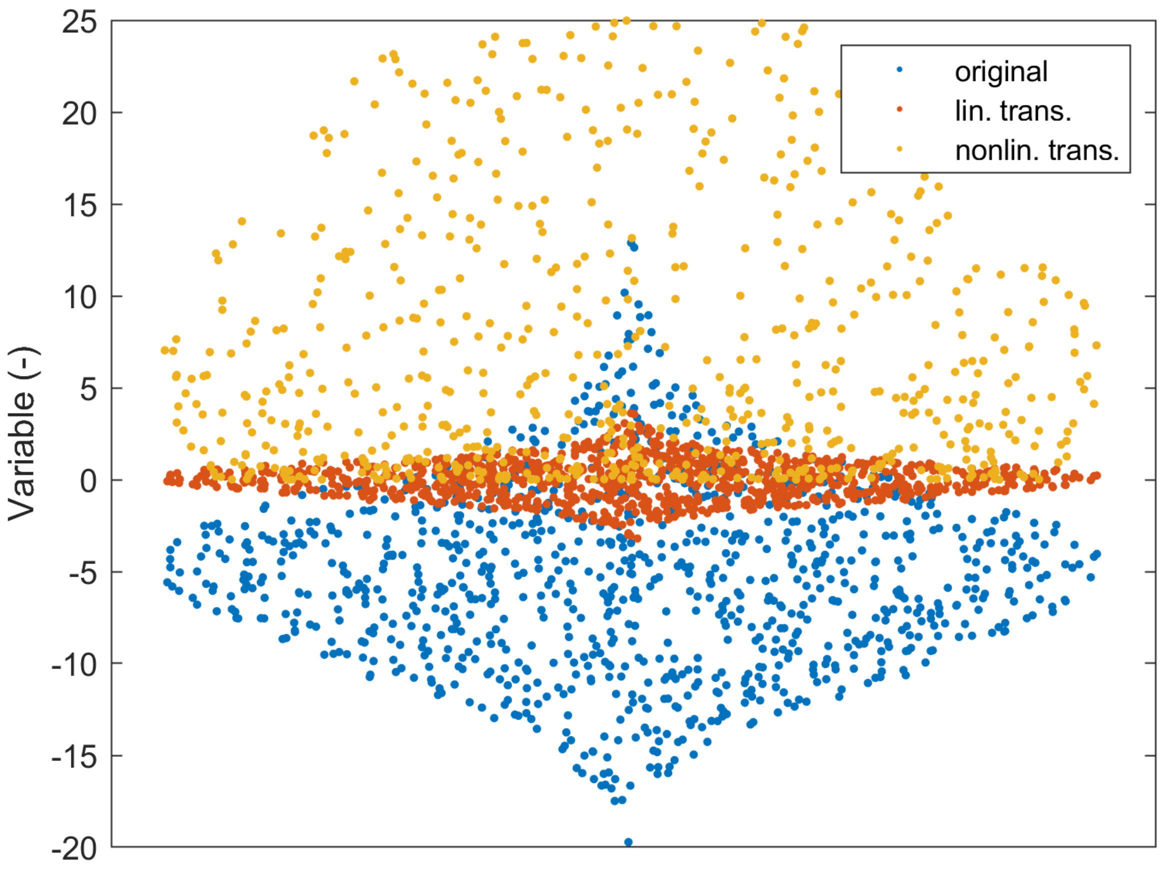
i. To normalize means and variances improving
   - comparability of different variables,
   - numerical stability, and
   - condition number,

ii. To build nondimensional models,

iii. To perform model order reduction,

iv. To perform feature engineering, or

v. To improve performance of classifiers.



Nonlinear data transformation

The most used nonlinear transformation is a decibel scale. It is commonly used in signal processing, communication and accoustics. For example, the **sound pressure level** is:

$$\mathrm{SPL} = 20\log_{10}\left(\frac{p}{2\cdot 10^{-5}}\right)$$

	Noise level in dB	Common Environment	Your conversation would be...
20 Pa	120	Jet engine nearby	
	110	Police siren nearby	IMPOSSIBLE
2 Pa	100	Inside subway train	
	90	Using hair drier	
0,2 Pa	80	Truck passing by	DIFFICULT
	70	Street with car traffic	LOUD VOICE REQUIRED
0,02 Pa	60	Normal conversations at office	
	50	Moderate rainfall	
2 mPa	40	Quiet residential area	EASY
	30	Whispering	
0,2 mPa	20	Rustling leaves	
	10	Breathing	

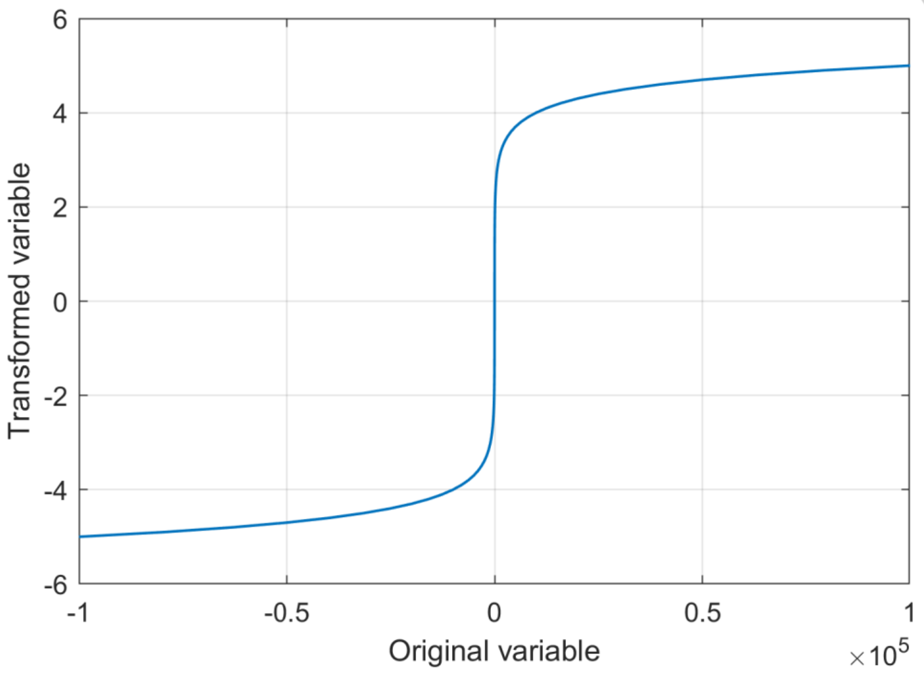
Nonlinear data transformation

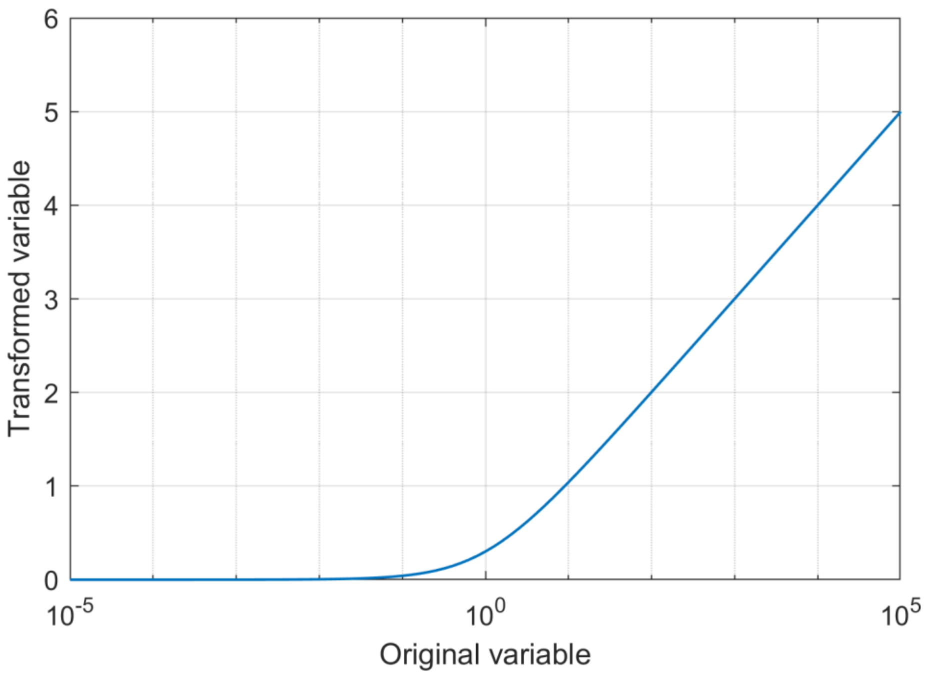
The most used nonlinear transformation is a decibel scale. It is commonly used in signal processing, communication and accoustics. For example, the **sound pressure level** is:

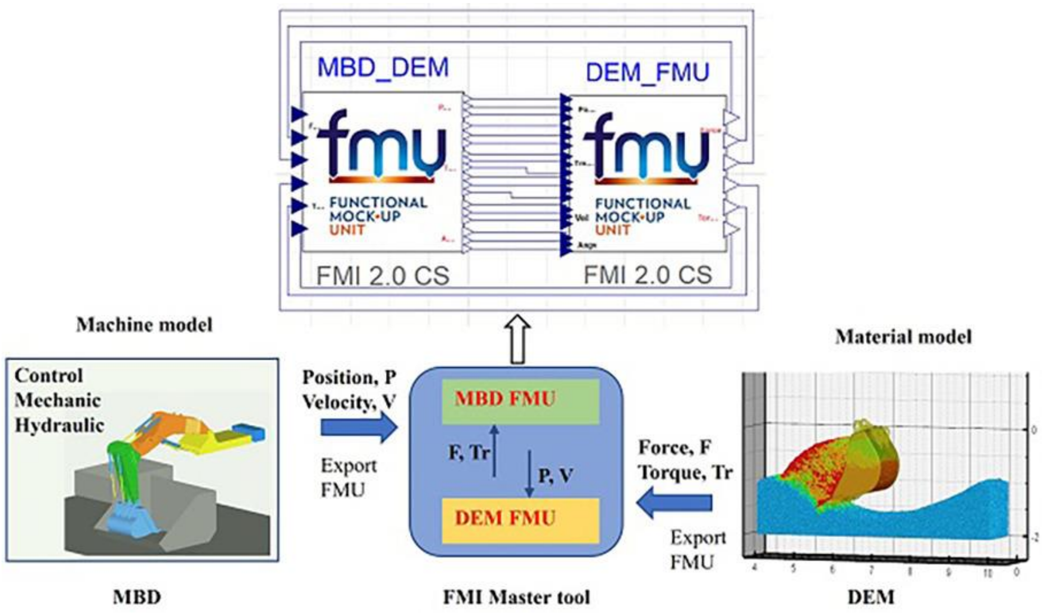
$$\mathrm{SPL} = 20 \log_{10}\left(\frac{p}{2 \cdot 10^{-5}}\right)$$

This transformation cannot transform negative numbers, so we propose the following:

$$x_{\log} = \mathrm{sgn}(x)\ \log_{10}(|x| + 1)$$







Section ii.

Cosimulation

Briefly on cosimulation

Cosimulation is the joint simulation of **loosely coupled** subsimulators.

Subsimulators are **independent**. However, each subsimulator may require **outputs** from other subsimulators as its **input**.

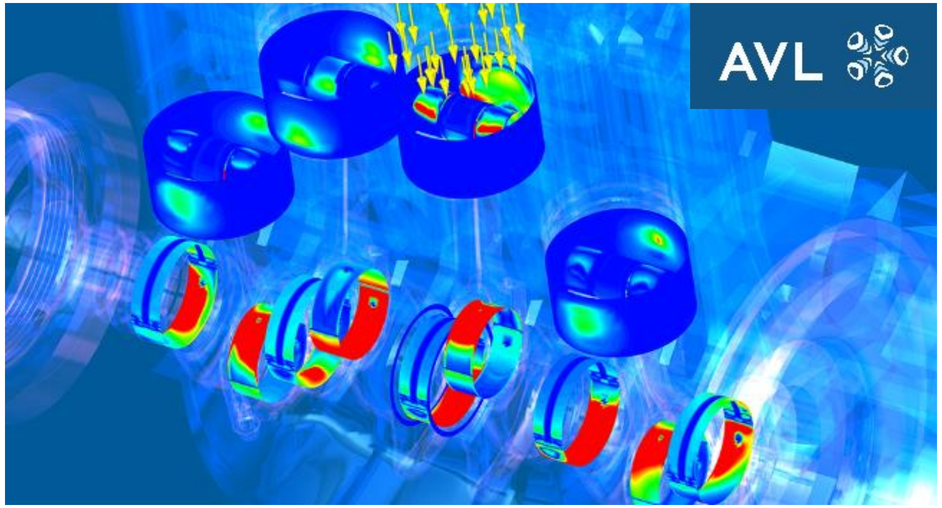
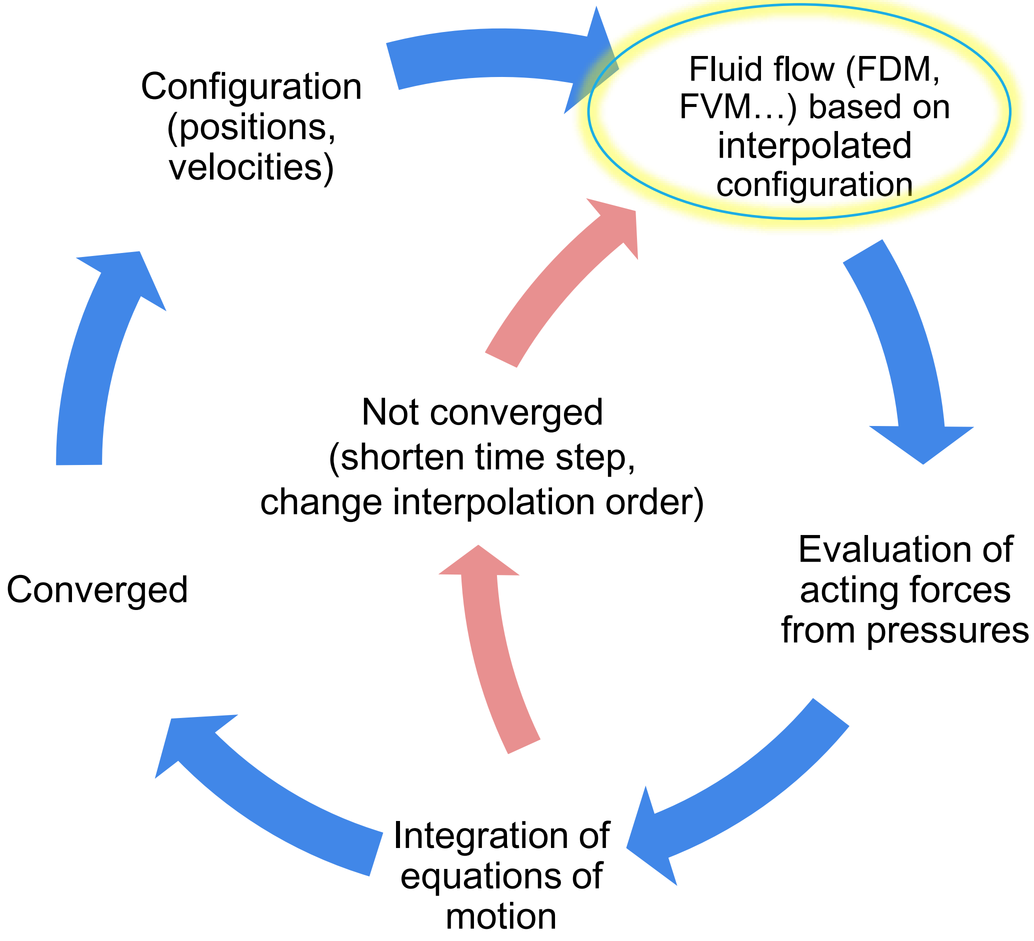


Fig.: Drive train of a combustion engine with hydrodynamic bearings (courtesy of AVL List GmBH)

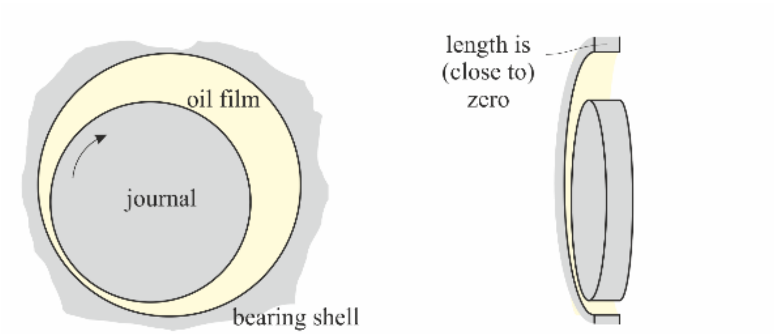
Cosimulation workflow in multibody systems interacting with fluid



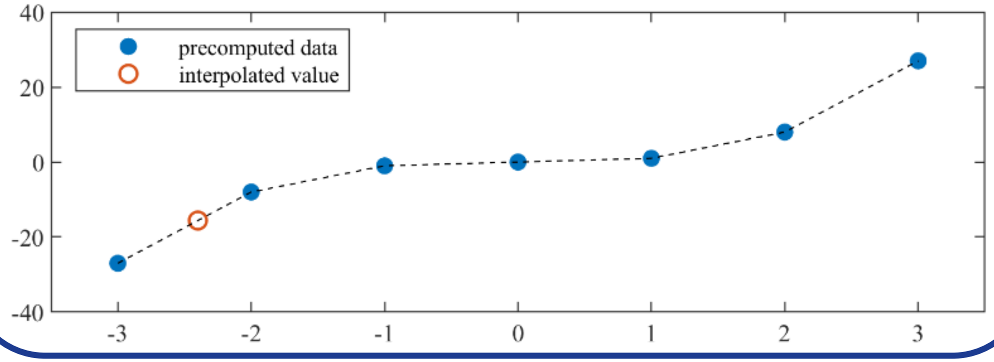
It is possible to speed up the calculation of the partial differential equation?

Techniques that can reduce the computational effort

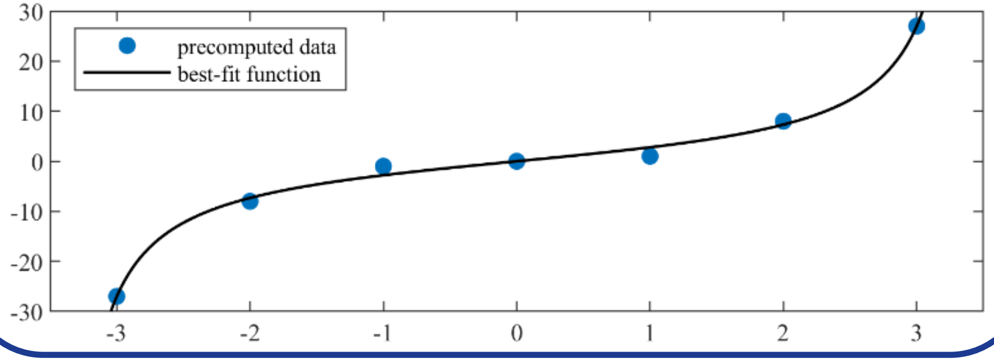
Approximate analytical solution



Database methods (look-up tables)

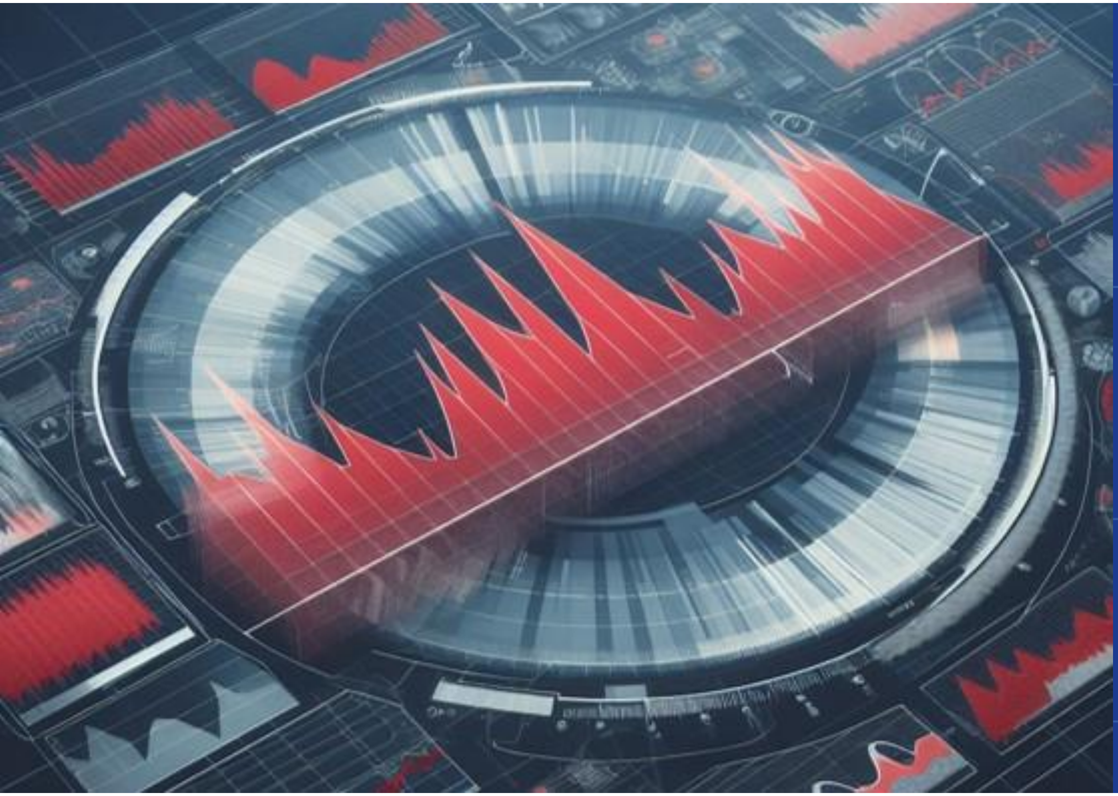


Regression models (best-fit approximations)



Hybrid methods

initial guess / approximation
+
correction function / correction model



Section iii.

Application

Hydrodynamic lubrication in journal bearings

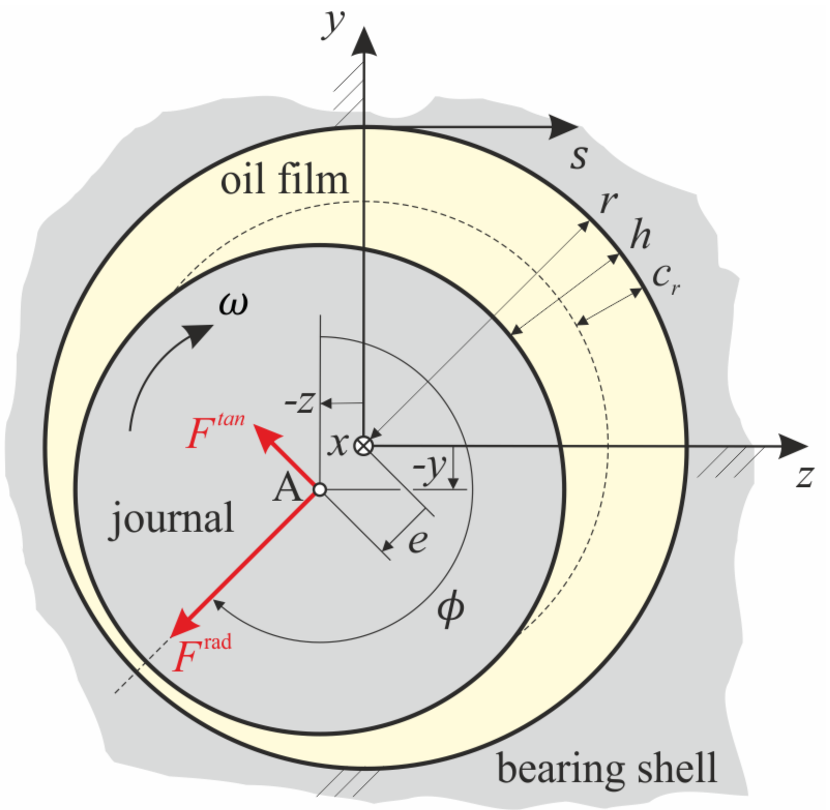
Pressure in thin viscous incompressible film and laminar flow:

$$\frac{\partial}{\partial s}\left(\frac{h^3}{\mu}\frac{\partial p}{\partial s}\right)+\frac{\partial}{\partial x}\left(\frac{h^3}{\mu}\frac{\partial p}{\partial x}\right)=6\,\frac{\partial(\omega\, r\, h)}{\partial s}+12\,\frac{\partial h}{\partial t}$$

Hydrodynamic forces acting on the journal:

$$F^{\mathrm{rad}}=-\iint p\cos\left(\frac{s}{r}-\phi\right)\mathrm{d}s\mathrm{d}x$$

$$F^{\mathrm{tan}}=-\iint p\sin\left(\frac{s}{r}-\phi\right)\mathrm{d}s\mathrm{d}x$$



VZÚ PLZEŇ

Application – Nondimensionalisation of problem

Displacements y, z → relative eccentricity $\varepsilon = \frac{\sqrt{y^2+z^2}}{c_r}$

attitude angle $\phi = \operatorname{atan}\left(\frac{z}{y}\right), \forall y > 0, \forall z \geq 0$

Time t → nondimensional time $\tau = \omega\, t$

Velocities $\dot{y}, \dot{z}$ → derivative of eccentricity $\frac{\mathrm{d}\varepsilon}{\mathrm{d}\tau} = \dot{\bar{\varepsilon}} = \frac{y\,\dot{y}+z\,\dot{z}}{\omega\, c_r \sqrt{y^2+z^2}}$

derivative of angle $\frac{\mathrm{d}\,\phi}{\mathrm{d}\tau} = \dot{\bar{\phi}} = \frac{y\,\dot{z}-z\,\dot{y}}{\omega(y^2+z^2)}$

Forces $F^{\mathrm{rad}}, F^{\tan}$ → nondimensional force $F^i_{\mathrm{non}} = \frac{c_r^2}{\omega\,\mu\, r\, l^3} F^i = \Lambda\, F^i$

This ensures that $f\colon \left(\varepsilon, \phi, \dot{\bar{\varepsilon}}, \dot{\bar{\phi}}\right) \rightarrow \left(F^{\mathrm{rad}}_{\mathrm{non}}, F^{\tan}_{\mathrm{non}}\right)$ holds for **any bearing** with given $\eta = \frac{L}{2\,R}$

Application – Architecture of an artificial neural network

Universal approximation theorem: *Any continuous function can be approximated using a sum of sigmoid activation functions* [i.e. by FFNN]*.*

The ideal number of hidden layers, neurons and their allocation depends on characteristics of the approximated and activation functions.

Hornik et al. (1989): *With any nonconstant activation function, a one-hidden-layer pi-sigma network* [i.e. fully-connected NN] *is a universal approximator* [of one output]*.*

Reference: Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*. 1989, 2(5): 359–366. doi: 10.1016/0893-6080(89)90020-8

Application – Proposed Architecture of a FFNN

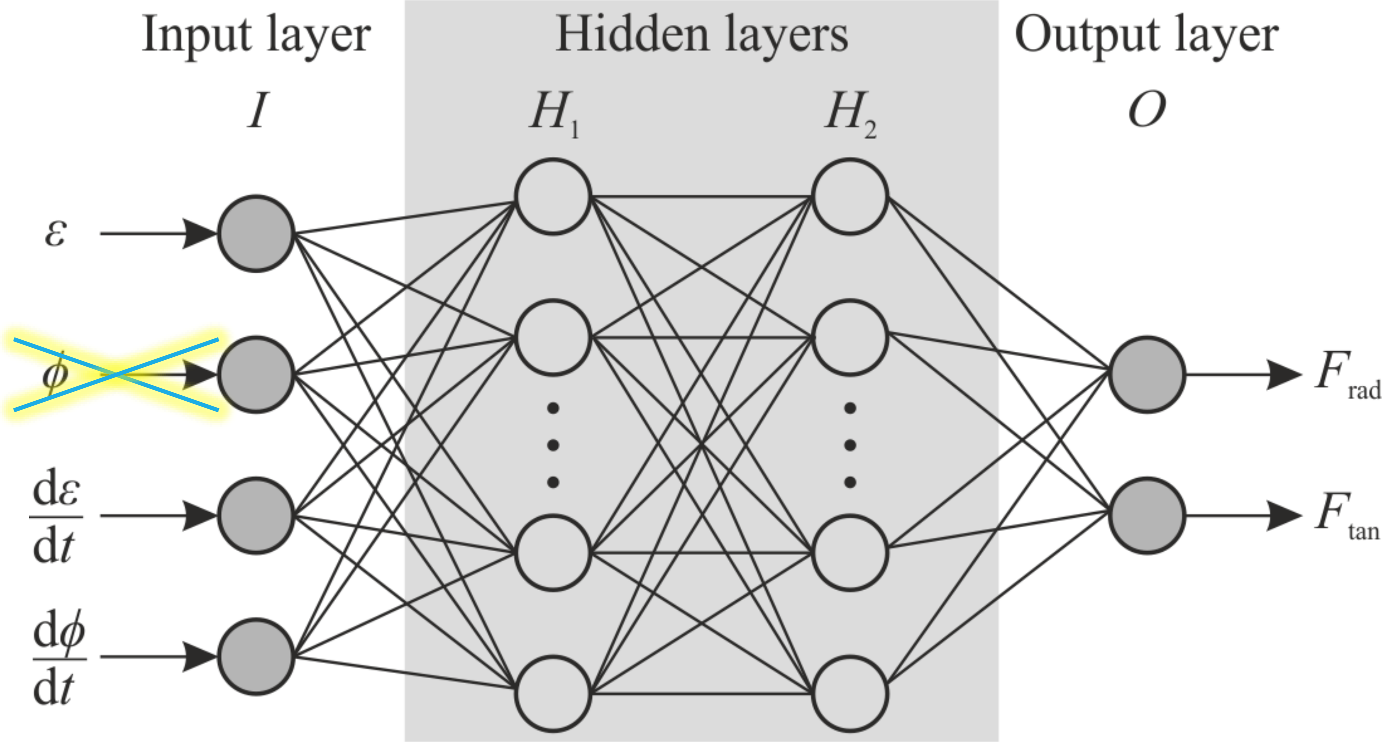


Fig.: Proposed FFNN with two hidden layers; note that for **symmetrical bearings** the input can be simplified

Inputs:
eccentricity and velocities

Outputs:
forces (pressure bypassed)

Transfer function (node *i*, layer *k*):

$$h_i^k = b_i^k + \sum_{j=1}^{n_{k-1}} w_{ij}^k \, o_j^{k-1}$$

bias   weight   output from previous layer

$$o_i^k = \frac{2}{1 + \exp(-2\, h_i^k)}$$

Application – Train data

Train data: database of force vectors in grid of 51×51×51 configuration points

Validation & test data: database of force vectors in 4500 & 30000 random conf. points

Target error: MSE = 0.000001 (better performance than MSE = 0!)

Training method: Bayesian regularisation-backpropagation

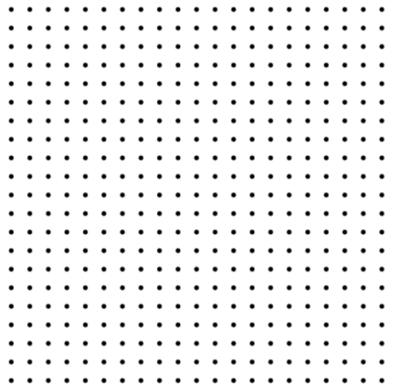


Fig.: Representation of train data

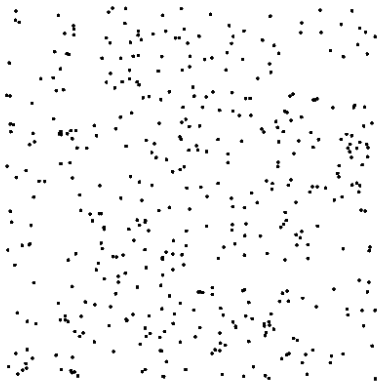
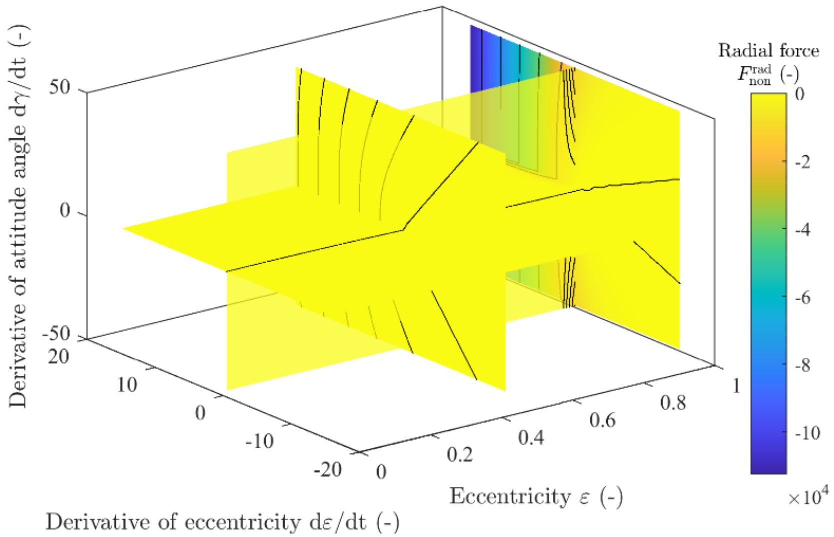
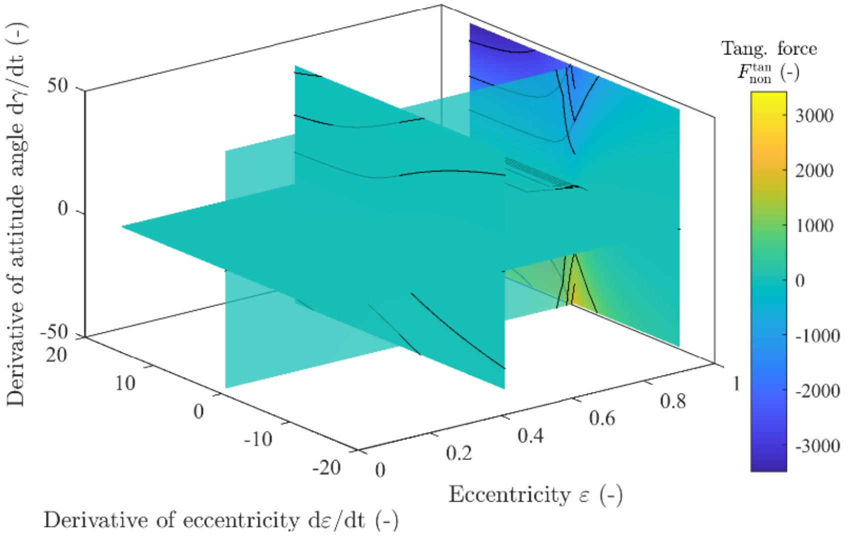


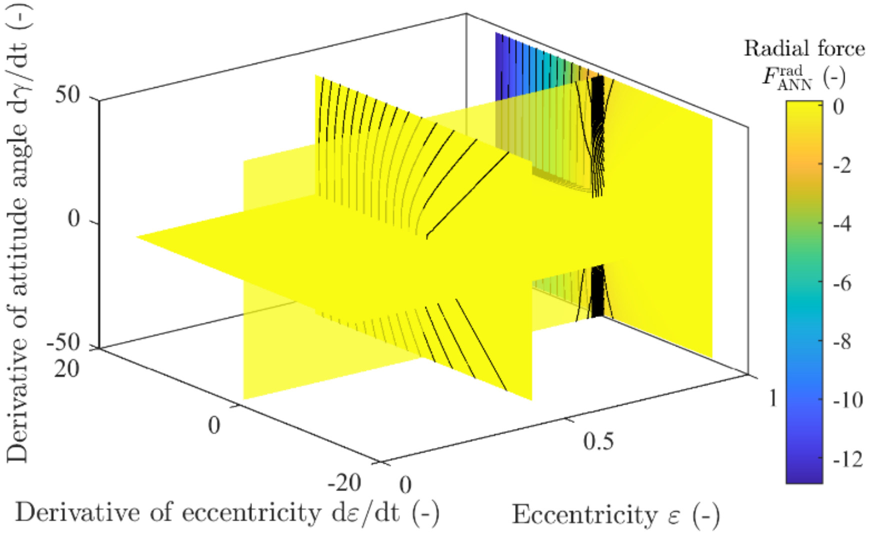
Fig.: Representation of validation & test data

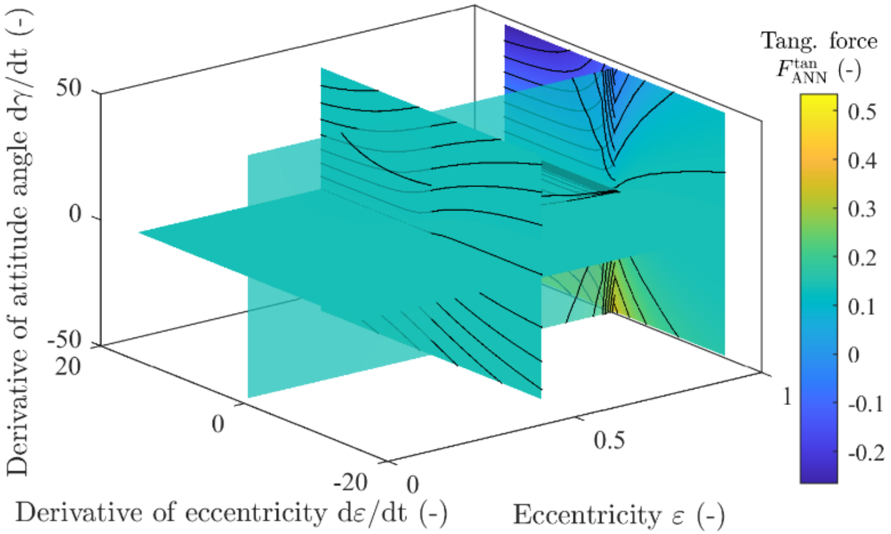
Nondimensional forces





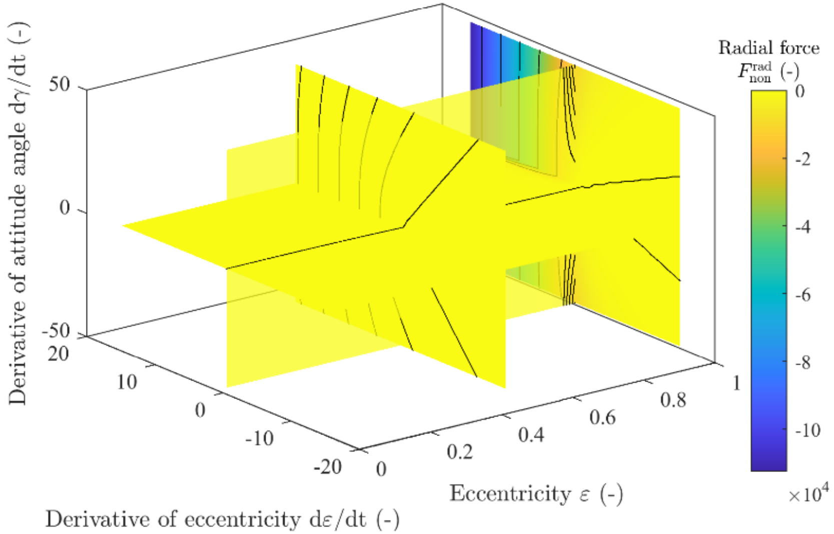
$$F_{\text{target}}^{i} = \frac{F_{\text{non}}^{i} - \text{mean}\left(F_{\text{non}}^{i}\right)}{\text{std}\left(F_{\text{non}}^{i}\right)}$$

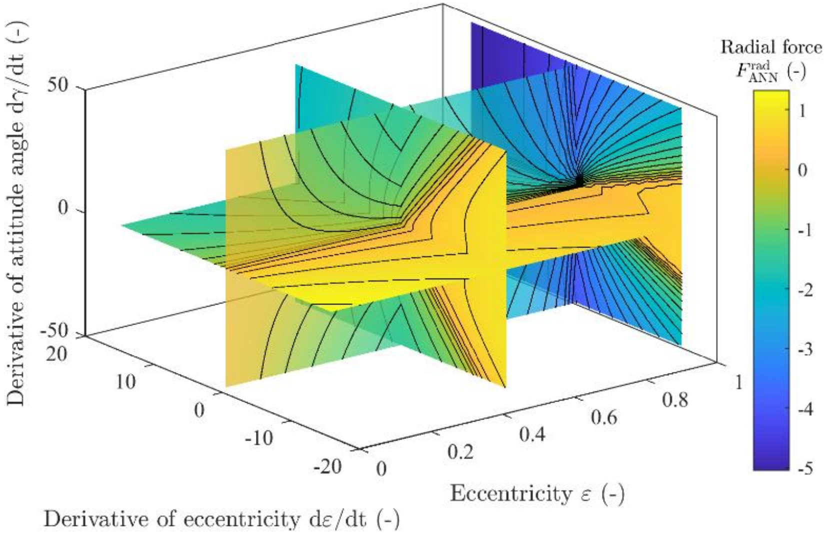




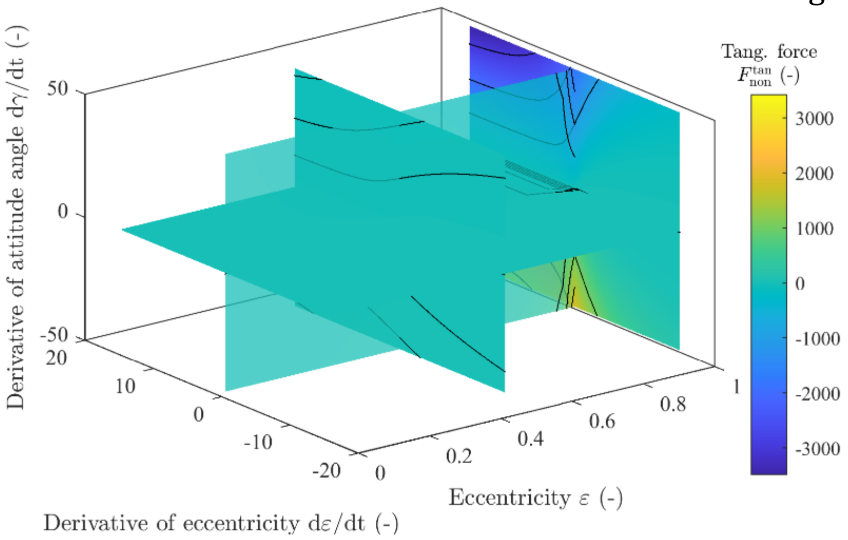
Transformed forces (z-scoring)

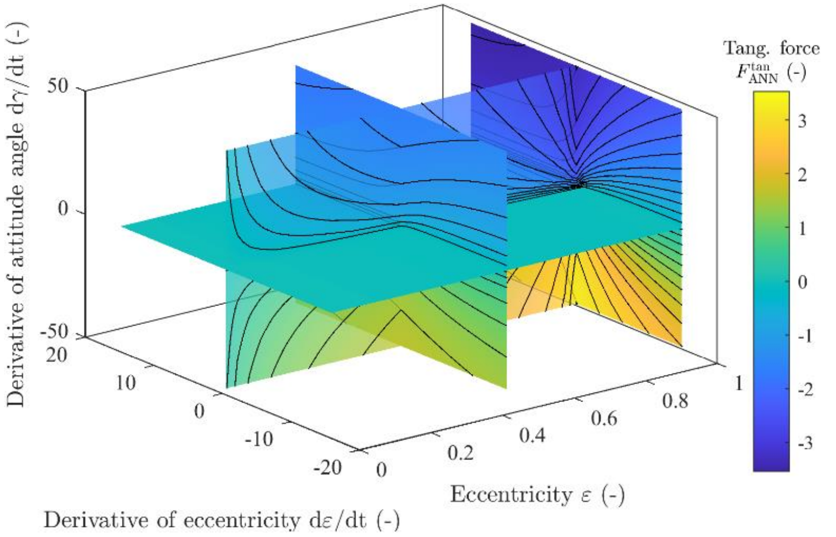
Nondimensional forces





$$F^{i}_{\mathrm{target}} = \mathrm{sgn}\left(F^{i}_{\mathrm{non}}\right) \cdot \log_{10}\left(\left|F^{i}_{\mathrm{non}}\right| + 1\right)$$





Transformed forces (log. scaling)

VZÚ PLZEŇ

Results – Performance evaluation

Performance is evaluated using data that are **unknown to ANN**, i.e. that were not used during the training.

Metrics: i) Root-mean-square error ii) Coefficient of determination

$$\mathrm{RMSE}^i_{\mathrm{met}} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(F^i_{\mathrm{met},j} - F^{\mathrm{i}}_{\mathrm{non},j}\right)^2}$$

$$(R^2)^i_{\mathrm{met}} = 1 - \frac{\sum_{j=1}^{n}\left(F^{\mathrm{i}}_{\mathrm{non},j} - F^i_{\mathrm{met},j}\right)^2}{\sum_{j=1}^{n}\left[F^{\mathrm{i}}_{\mathrm{non},j} - \mathrm{mean}\left(F^{\mathrm{i}}_{\mathrm{non},j}\right)\right]^2}$$

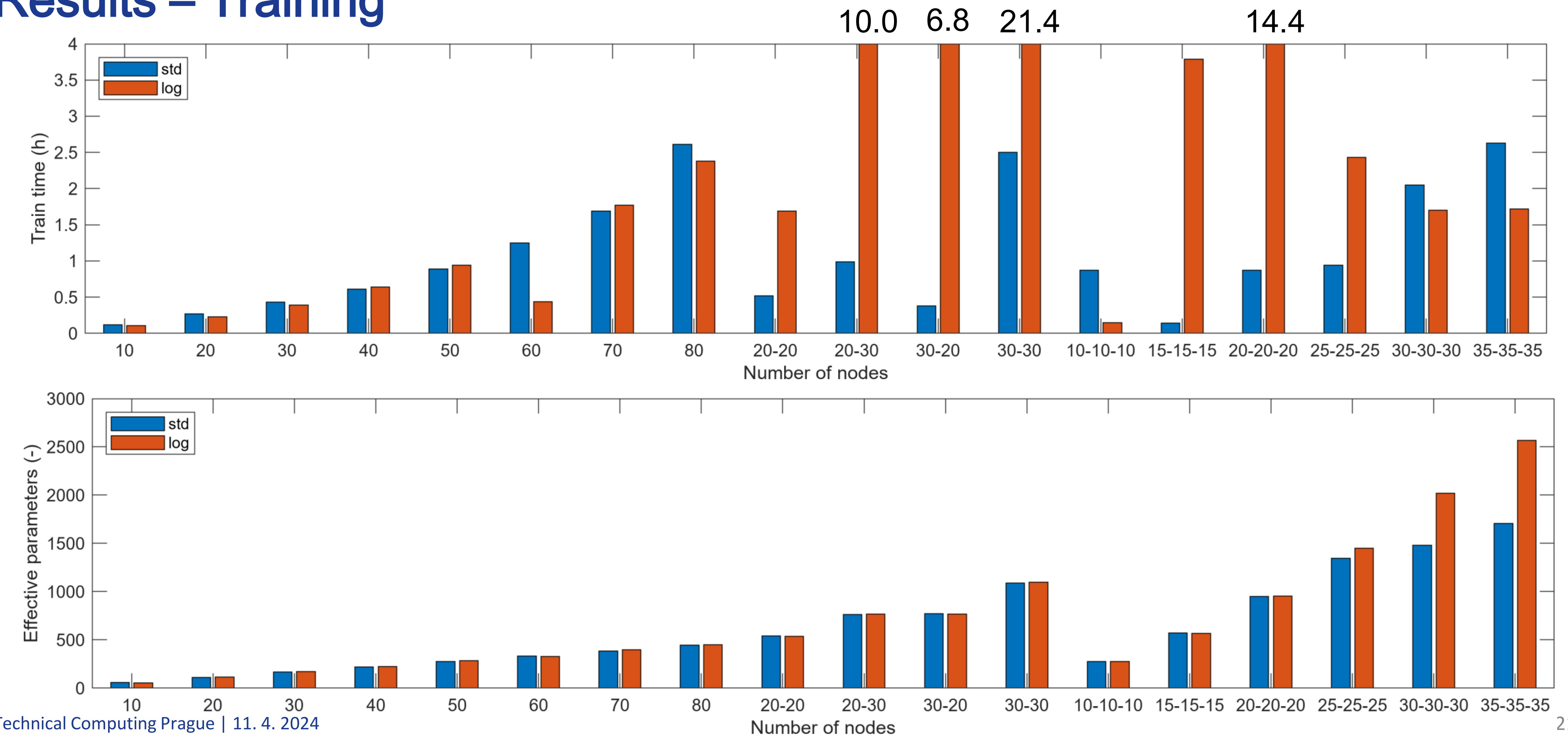
$$i = \{\mathrm{rad}, \tan\}$$
$$\mathrm{met} = \{\mathrm{std}, \log\}$$

Note that performance metrics use **nondimensional forces** – need for reverse transformation!

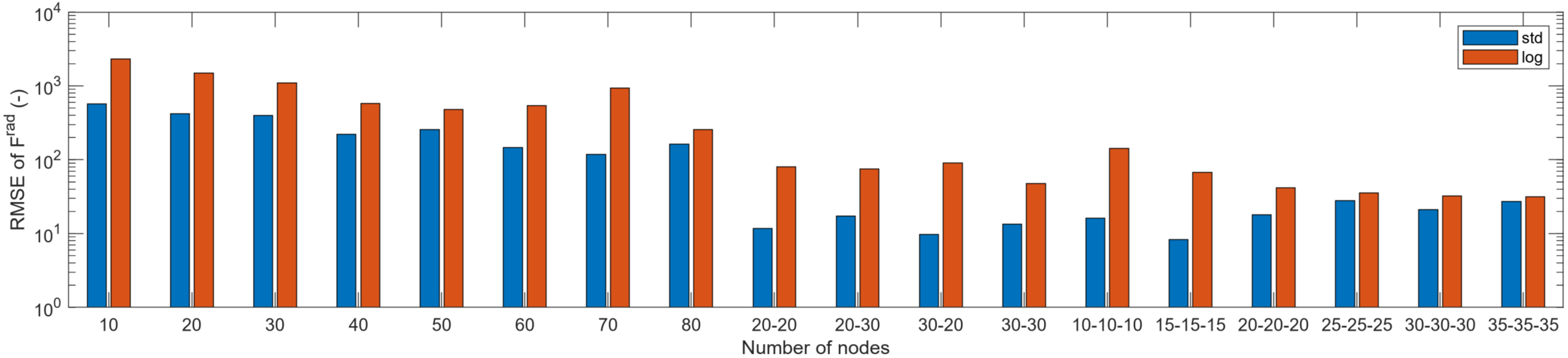
$$F^i_{\mathrm{pred}} = \frac{F^i_{\mathrm{non}} - \mathrm{mean}\left(F^i_{\mathrm{non}}\right)}{\mathrm{std}\left(F^i_{\mathrm{non}}\right)}$$

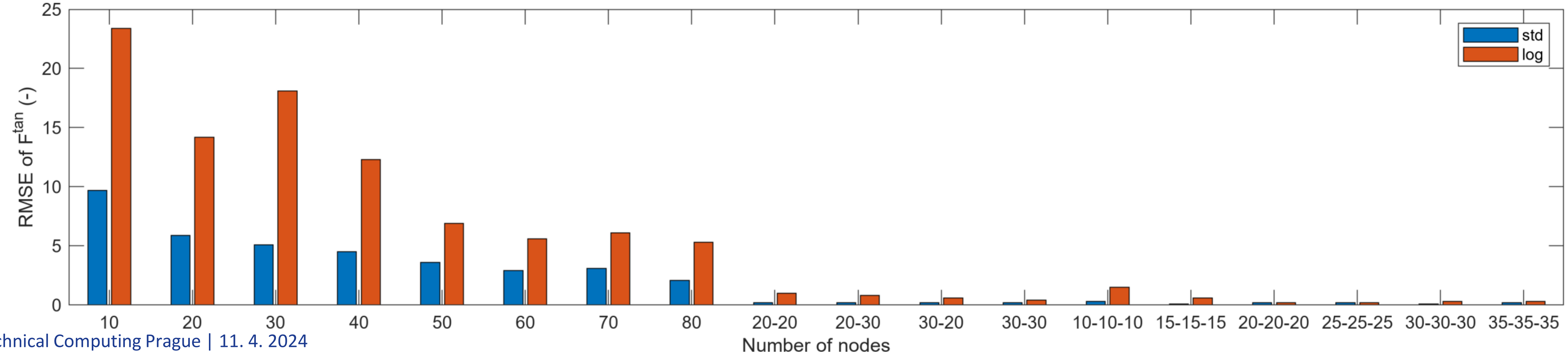
$$F^i_{\mathrm{pred}} = \mathrm{sgn}\left(F^i_{\mathrm{non}}\right) \cdot \log_{10}\left(\left|F^i_{\mathrm{non}}\right| + 1\right)$$

Results – Training

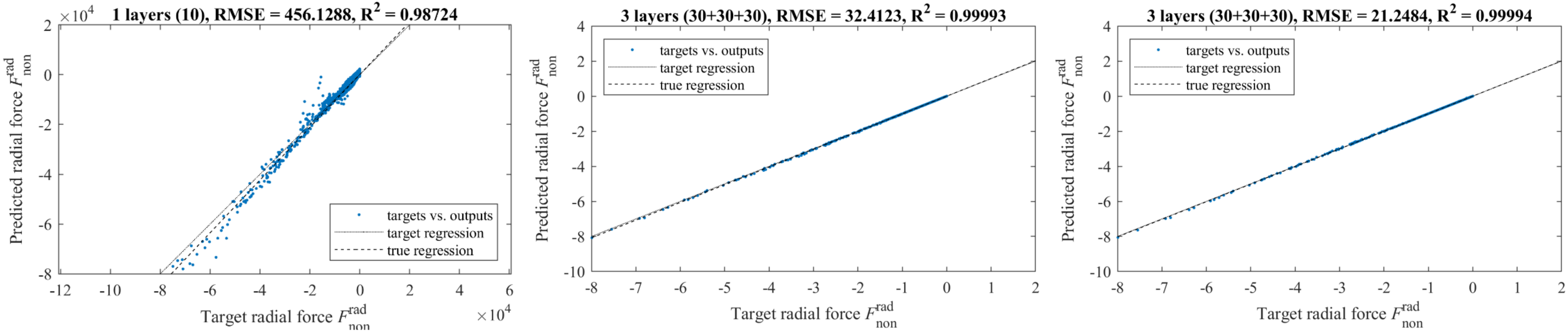


Results – RMSE



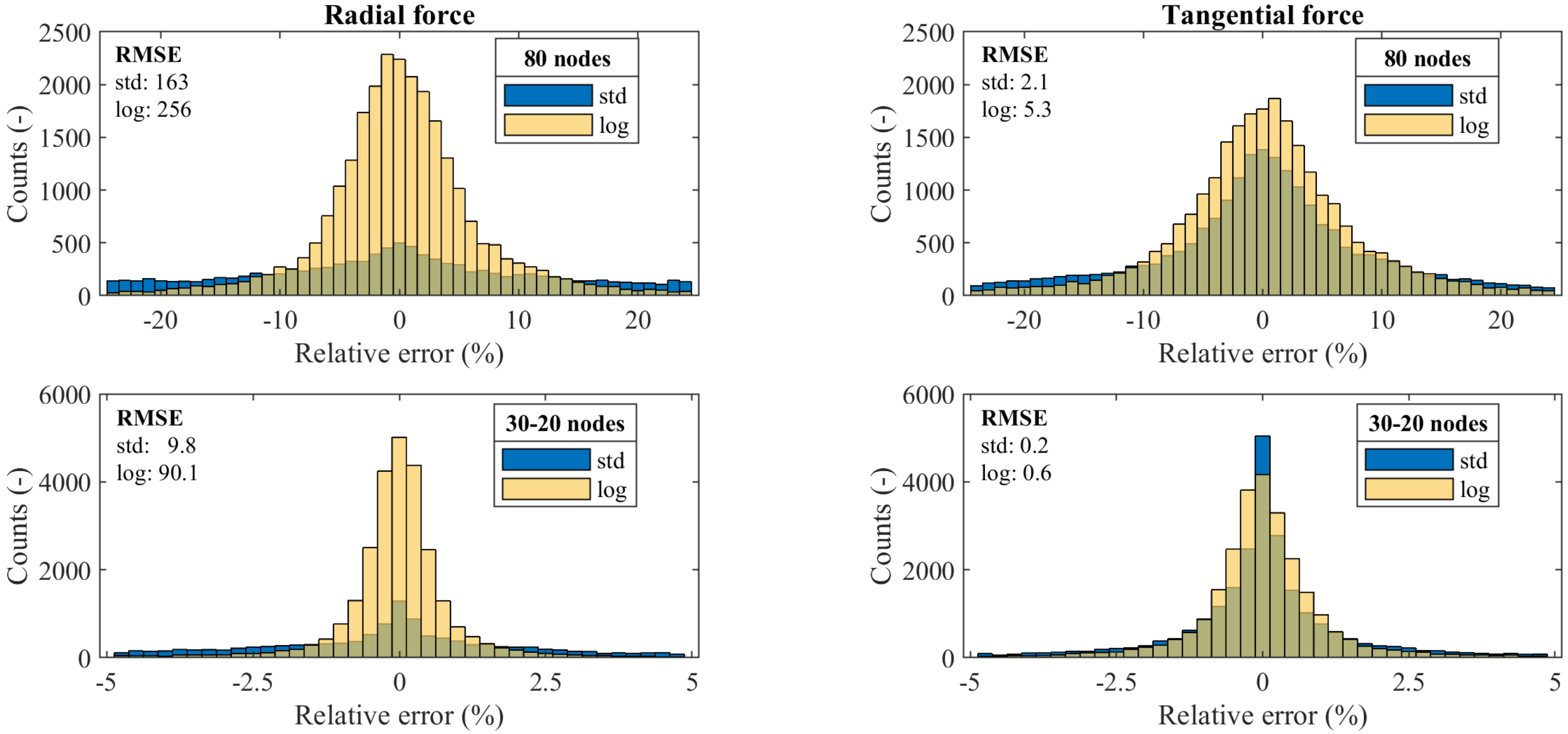


Results – Coefficients of determination



Coefficients of determination (R^2) are not sensitive enough to measure the performance of ANNs that have high accuracy.

Results – Relative errors of FFNNs with 1 and 2 layers



Results – Relative errors of FFNNs with 3 layers

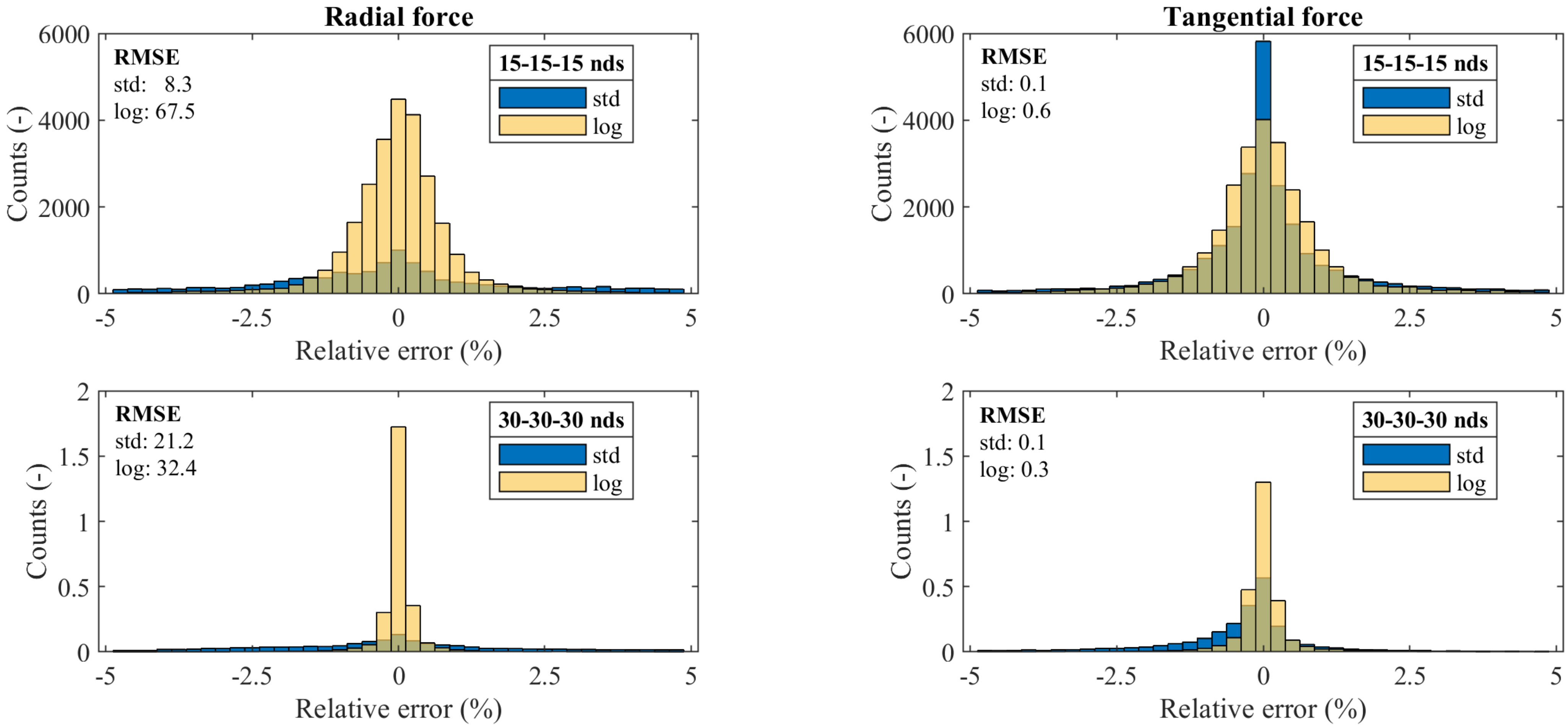


Fig: Relative errors of ANN with log scaled data are lower than those of ANN with z-scored data, although the RMSE values suggest the opposite!

Section iv.

Conclusions

Summary

i. RMSE and R^2 are not always the best metrics to evaluate the performance.

ii. RMSE and MSE depend on data scaling – do not use them to compare differently scaled models!

iii. Nonlinear transformation of train data (targets) can improve the distribution of prediction relative errors – similar relative errors over different orders of magnitude.

iv. One can use weights in a cost function instead of data transformation. With the uniform weights, the data transformation requires less flops (i.e. CPU time).

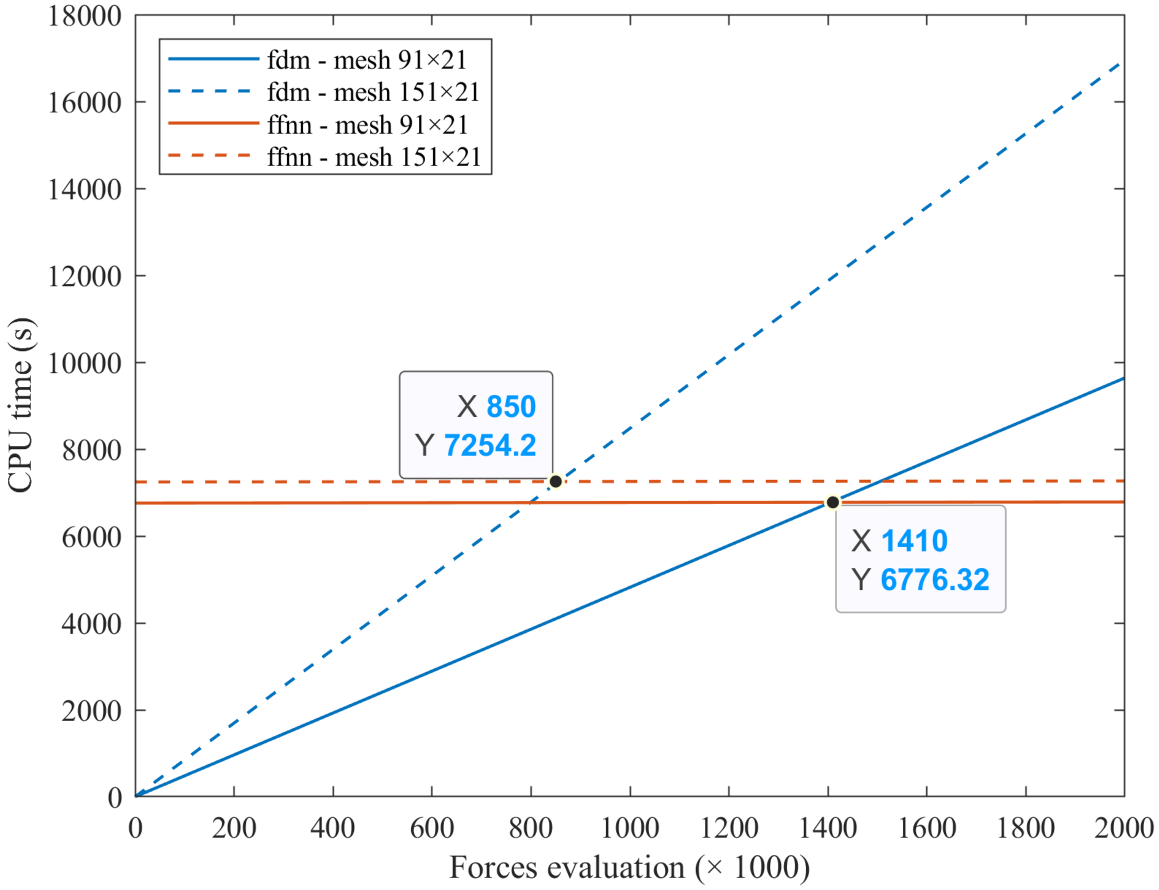
CPU time (assuming single CPU)

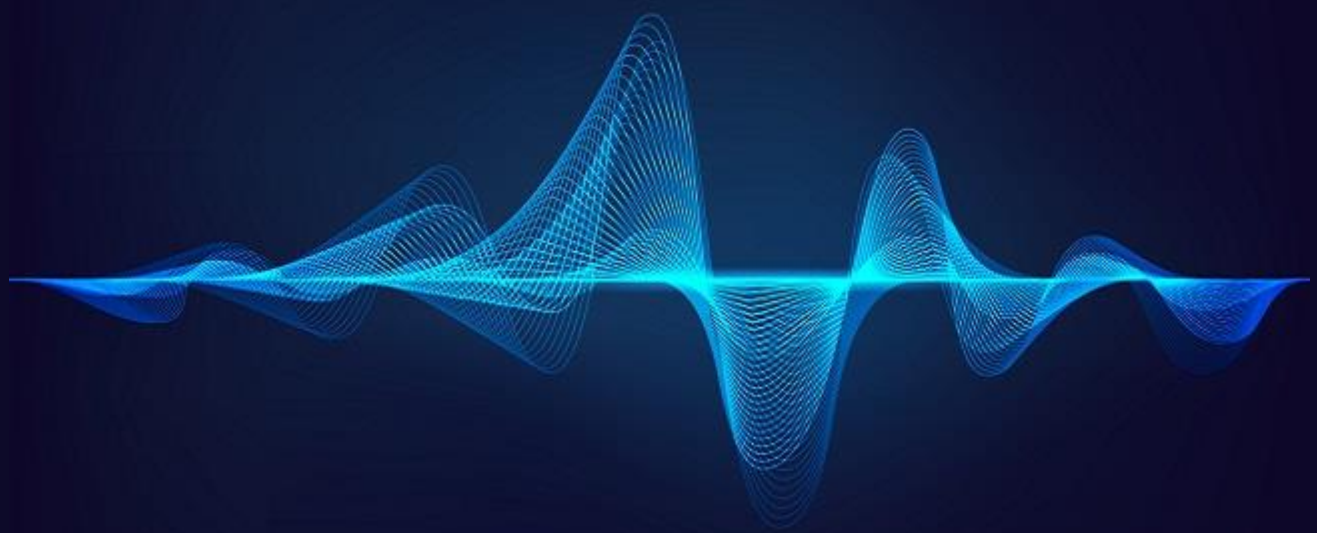
Standard approach:	assembling model	several seconds
	1000 evaluations of forces:	4.82 s at 91×21 mesh
		8.48 s at 151×21 mesh
ANN:	database construction:	639.4 s at 91×21 mesh
		1124 s at 151×21 mesh
	training:	1.7 h (30-30-30 and 35-35-35 ANNs)
	1000 evaluations of forces:	0.012 s

Questions regarding CPU time

Is the database construction and training of ANN justifiable?

Which approach (standard that solves partial differential equation directly or ANN) is more efficient in which situation?





Na jedné vlně

Luboš Smolík smolik@vzuplzen.cz

Jan Rendl rendlj@ntis.zcu.cz

Radek Bulín rbulin@ntis.zcu.cz

Výzkumný a zkušební ústav Plzeň s.r.o.

Tylova 1581/46, 301 00 Plzeň

Tel.: +420 371 430 700

E-mail: obchod@vzuplzen.cz

www.vzuplzen.cz

This presentation was supported by the institutional support for long-term conceptual development of research institutions provided by the Ministry of Industry and Trade of the Czech Republic to VZÚ Plzeň.