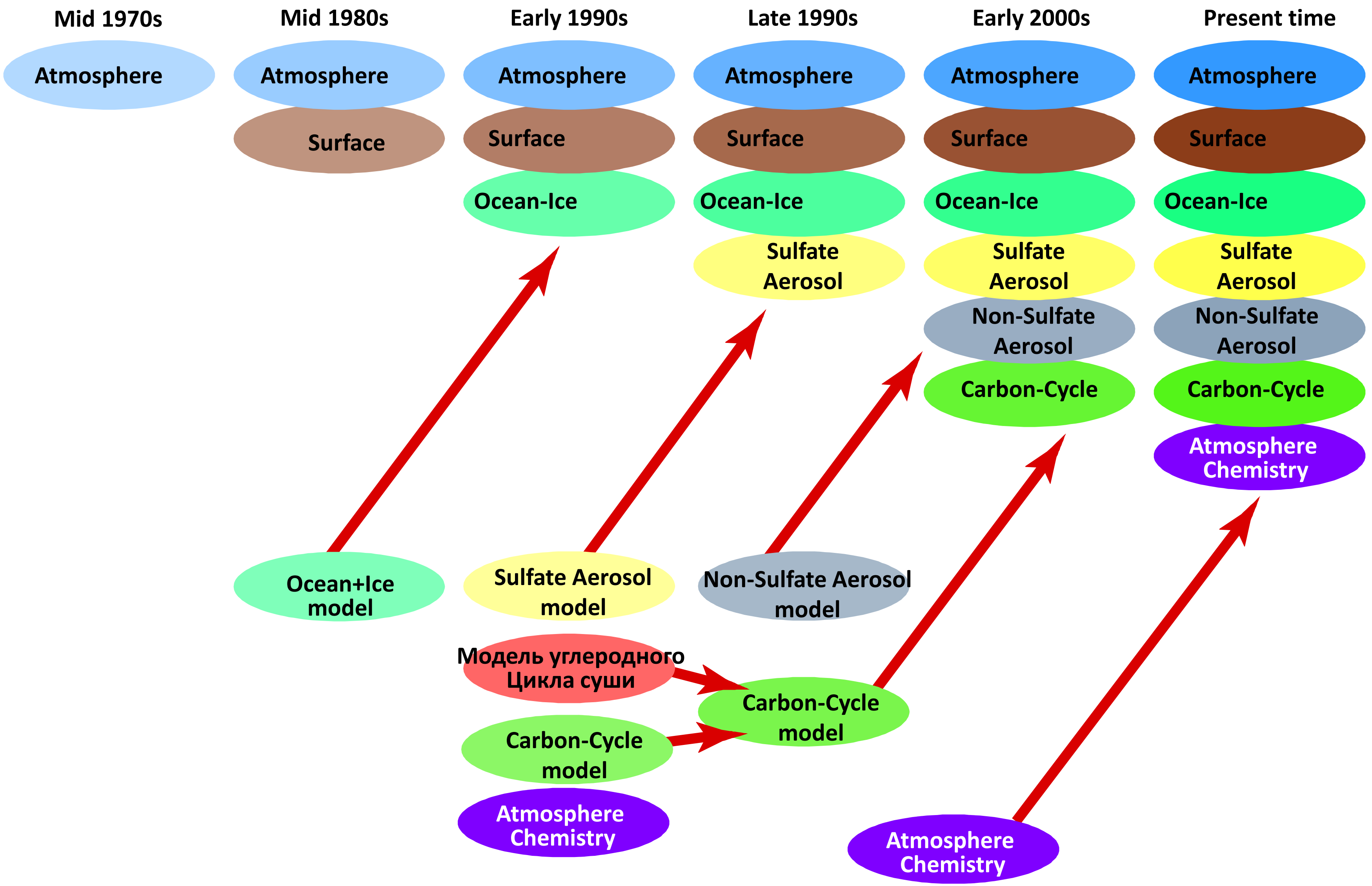
MOSCOW STATE UNIVERSITY RESEARCH COMPUTING CENTER

Applying neural network for identification of land surface model parameters

Chernyshev Ruslan[1,2], Krinitskiy Mikhail[3,1], Stepanenko Viktor[1,2,4]
1 – Research Computing Center, Lomonosov MSU, Moscow, Russia
2 – Faculty of Geography, Lomonosov MSU, Moscow, Russia
3 - Shirshov Institute of Oceanology, Russian Academy of Sciences, Moscow, Russia
4 - Moscow Center for Fundamental and Applied Mathematics, Moscow, Russia

Earth system models structure evolution



Neural network identification of thermal conductivity equation

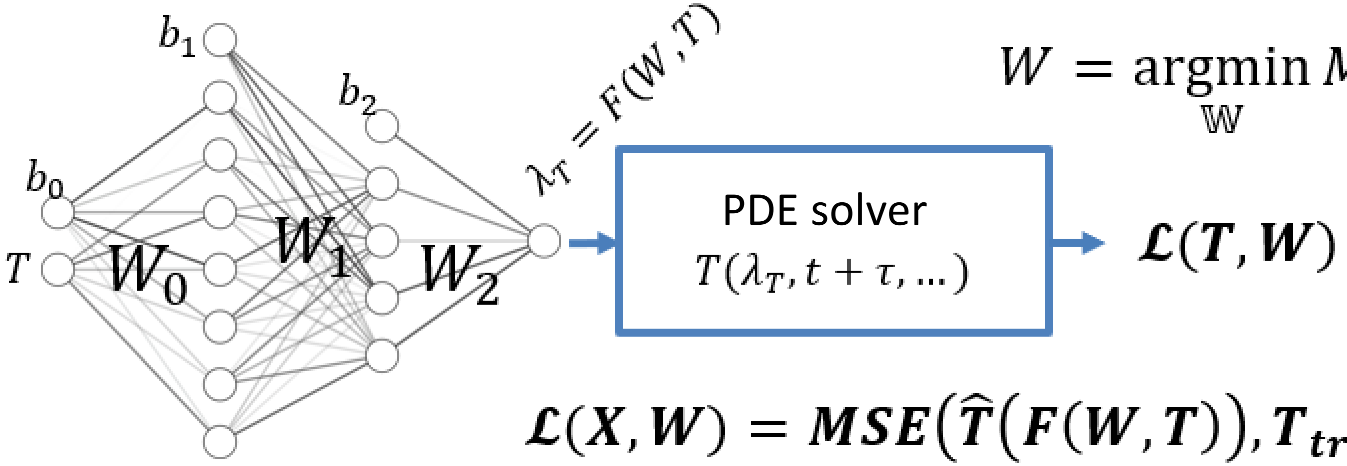
$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(T)\frac{\partial T}{\partial z}\right)$$

$\lambda(T)$ can't measure

T – can be measured

The task is to find a dependency $\lambda_{NN}(T)$ represented by a neural network, given a solution T of this equation and which minimizes the error of T_{NN} approximation in respect to T, where T_{NN} is a solution of with $\lambda_T(T) = \lambda_{NN}(T)$

$$\widehat{\lambda_T} = \phi\left(b_2 + W_2\psi\left(b_1 + W_1\psi(b_0 + W_0 T)\right)\right) = F(W,T)$$ - Neural network inside



$$W = \underset{\mathbb{W}}{\operatorname{argmin}}\, MSE(T_{true}, F(T,W))$$

$$\mathcal{L}(X,W) = MSE\left(\hat{T}\left(F(W,T)\right), T_{true}\right)$$ Base loss function

$$\frac{\partial \mathcal{L}(T,W)}{\partial W} = 2\left(\hat{T}\left(F(T,W)\right) - T_{true}\right) * \frac{\partial \hat{T}}{\partial F} * \frac{\partial F}{\partial W}$$ Base gradient

Step 1 : T_{true} Analytic "true" mode

Step 2 : T_{true} from measured data

Four problems

first

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(T)\frac{\partial T}{\partial z}\right).$$

Non-linear thermal diffusion equation

second

$$\frac{\partial W}{\partial t} = \frac{\partial}{\partial z}\left(\lambda_W(W)\frac{\partial W}{\partial z}\right)$$

Non-linear liquid diffusion equation

third

$$\frac{\partial W}{\partial t} = \frac{\partial}{\partial z}\left(\lambda_W(W)\frac{\partial W}{\partial z}\right) + \frac{\partial \gamma_W(W)}{\partial z}$$

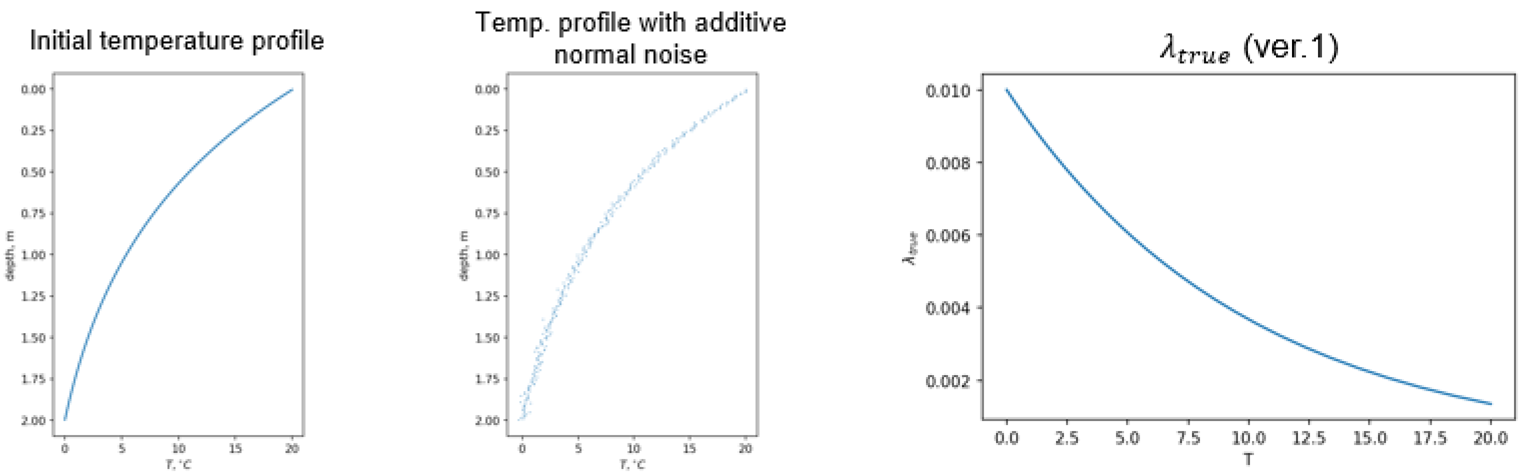
Richards equation for liquid moisture

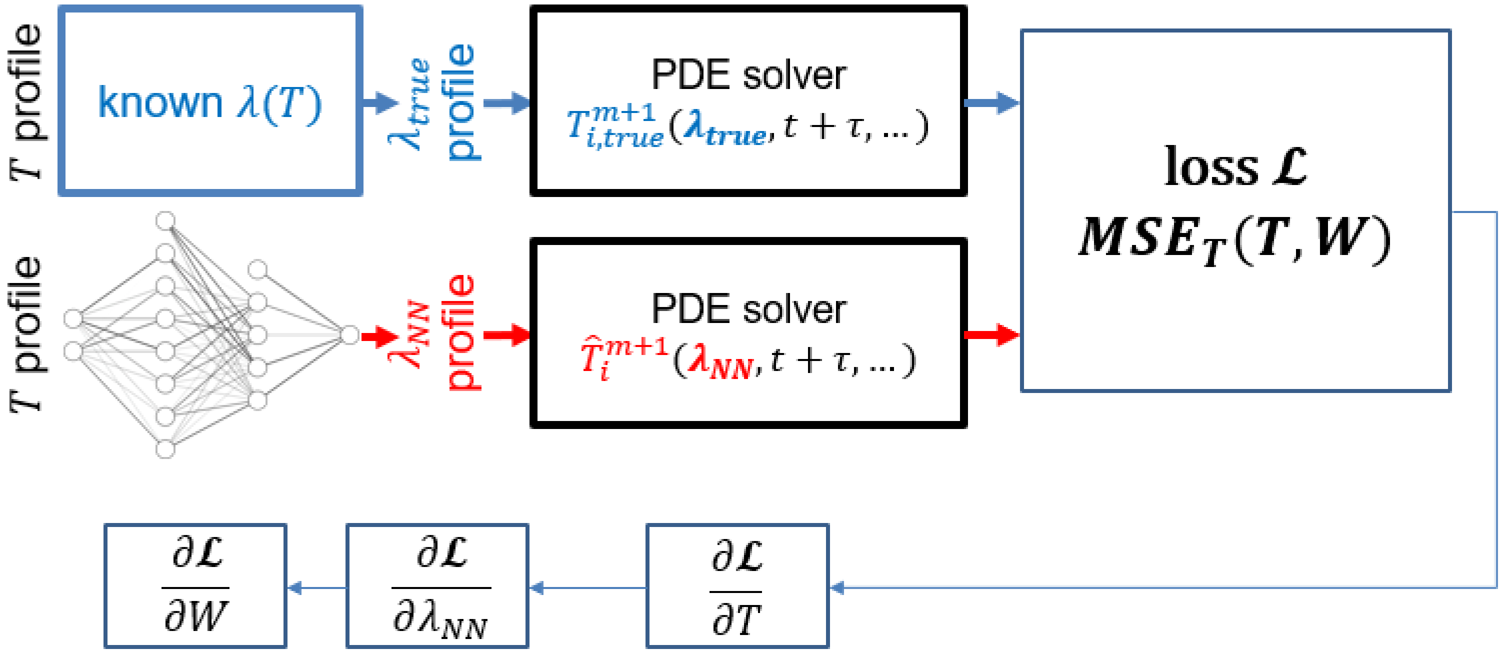
fourth

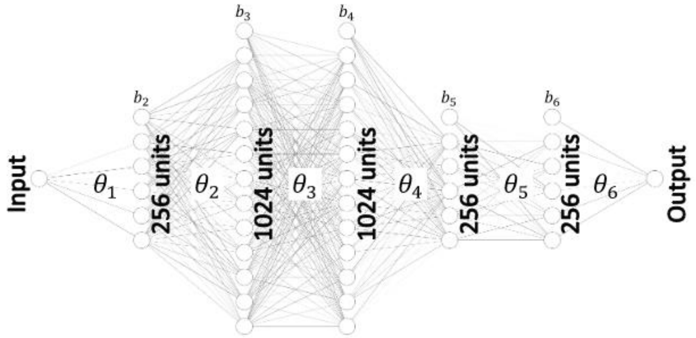
$$\begin{cases} \frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(W)\frac{\partial T}{\partial z}\right), \\ \frac{\partial W}{\partial t} = \frac{\partial}{\partial z}\left(\lambda_W(W)\frac{\partial W}{\partial z}\right) + \frac{\partial \gamma_W(W)}{\partial z}. \end{cases}$$

Richards equation + thermal conductivity

Neural network training specification





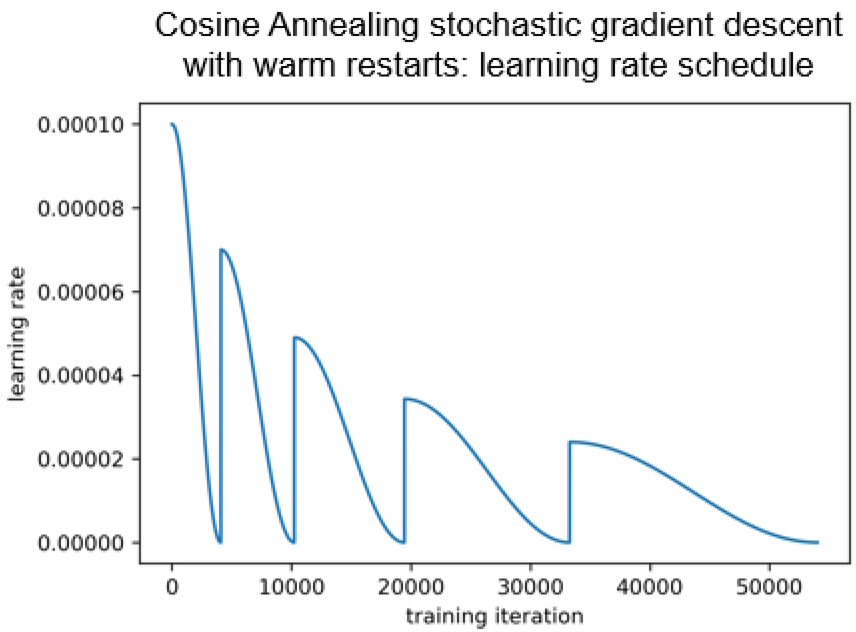


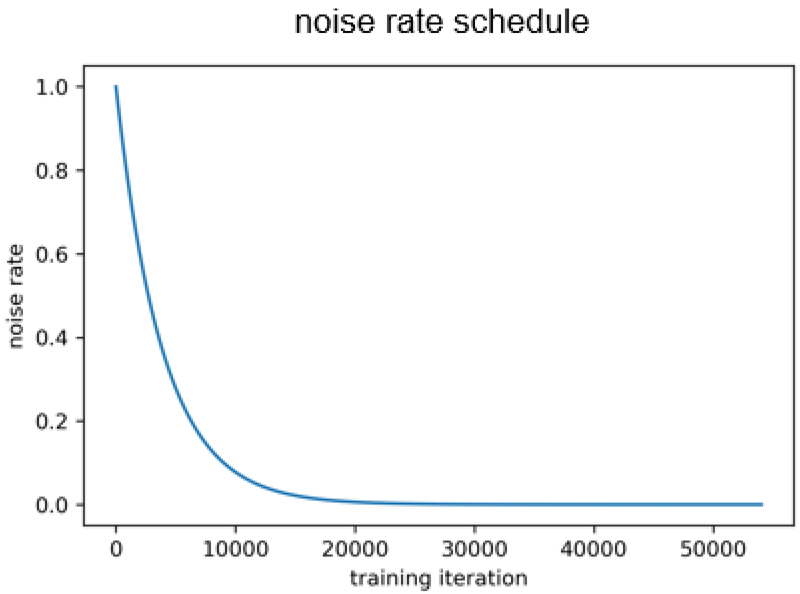
NN architecture

Optimizations

$$g_t = \nabla_W^{(t)} \mathcal{L}(T, W)$$
$$W_{t+1} = W_t - lr * g_t * \boldsymbol{S_A}$$

$$T_n = T + nr * N * \boldsymbol{S_n}$$
$$N \sim \mathcal{N}(0, \mathbb{I})$$



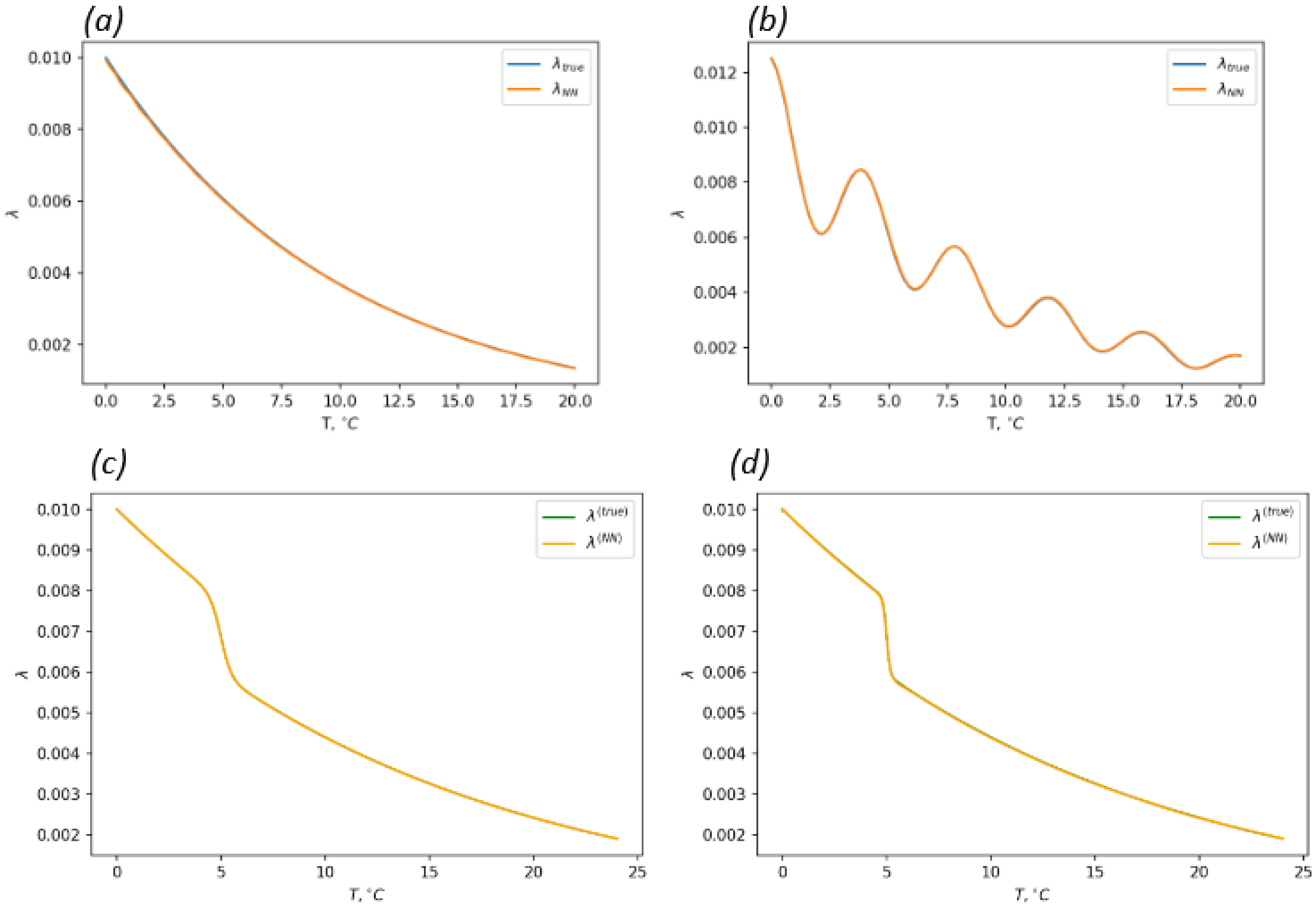


$\boldsymbol{S_A}$: a scale for the gradient according to Adam optimization scheme: Kingma D. P., Ba J. "Adam: A Method for Stochastic Optimization" // arXiv:1412.6980 [cs]. 2017.
$\boldsymbol{S_n}$: constant noise scale

Additional:
1. batch normalization
2. learning rate scheduling
3. scheduling of noise rate
4. additional regularization loss terms improving convergence

Penalties:
1. negative gradients of the network w.r.t. its input
2. negative output values
3. non-zero output values in the origin

Neural network results



First problem

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(T)\frac{\partial T}{\partial z}\right).$$

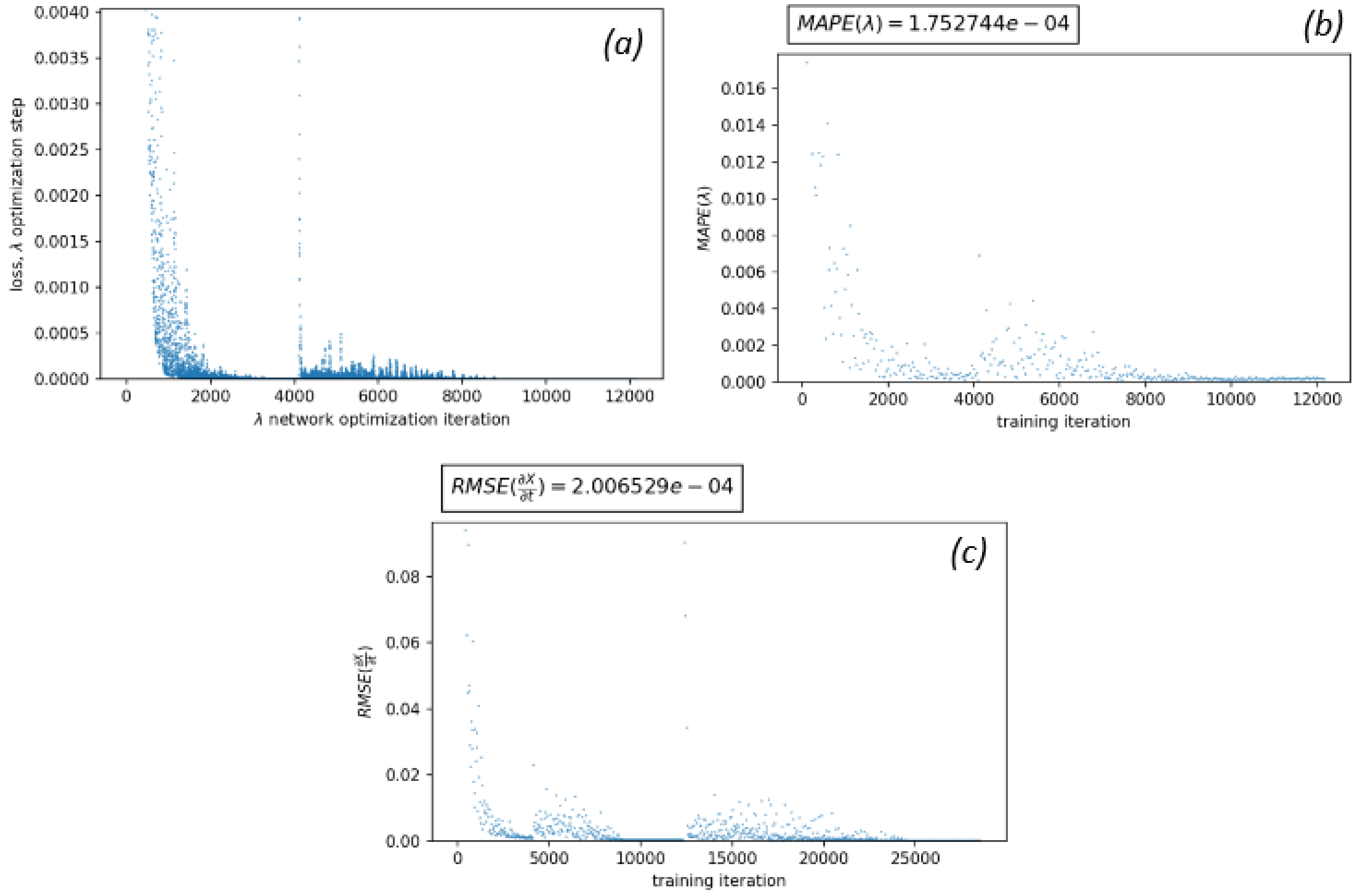
Comparing “true” and NN models:
a) Exponent-like behavior

b) Exponent-cos-like behavior

c, d) threshold-like behavior

Optimization characteristics

Loss function and quality



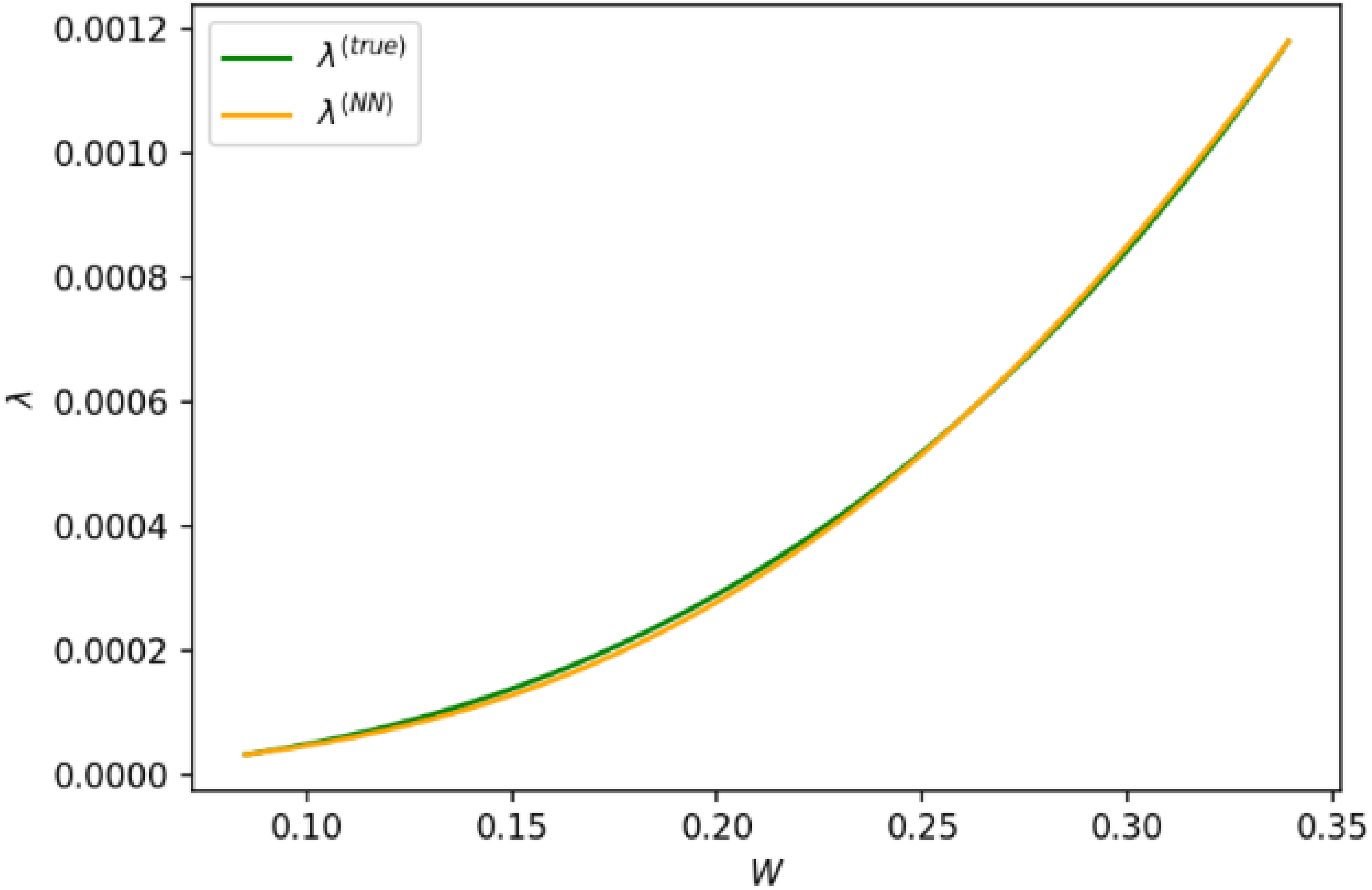
First problem

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z}\left(\lambda(T)\frac{\partial T}{\partial z}\right).$$

a) Absolute loss
b) MAPE loss
c) RMSE loss

$$MAPE(\lambda_{NN}, \lambda) = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\lambda_{i,NN} - \lambda_i}{\lambda_i + \xi}\right|.$$

Neural network results



Second problem

$$\frac{\partial W}{\partial t} = \frac{\partial}{\partial z}\left(\lambda_W(W)\frac{\partial W}{\partial z}\right)$$

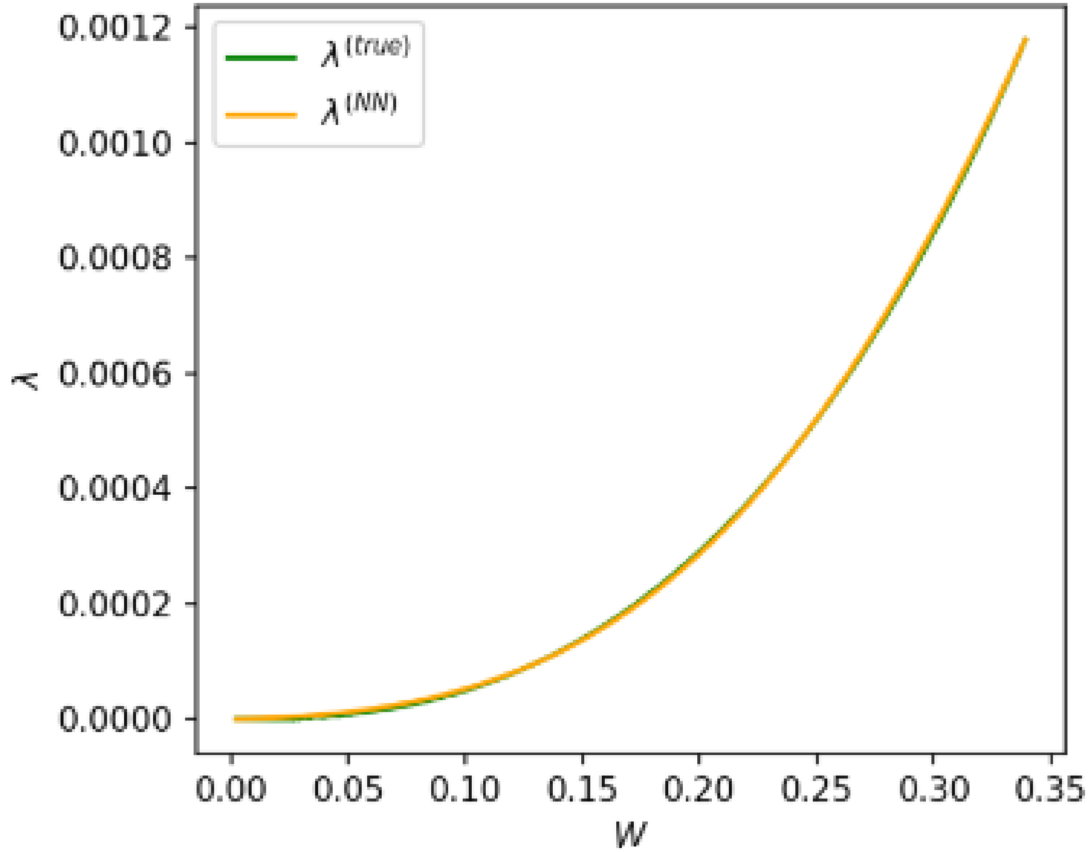
$MAPE(\lambda_W(W))$
2.3×10^{-2}

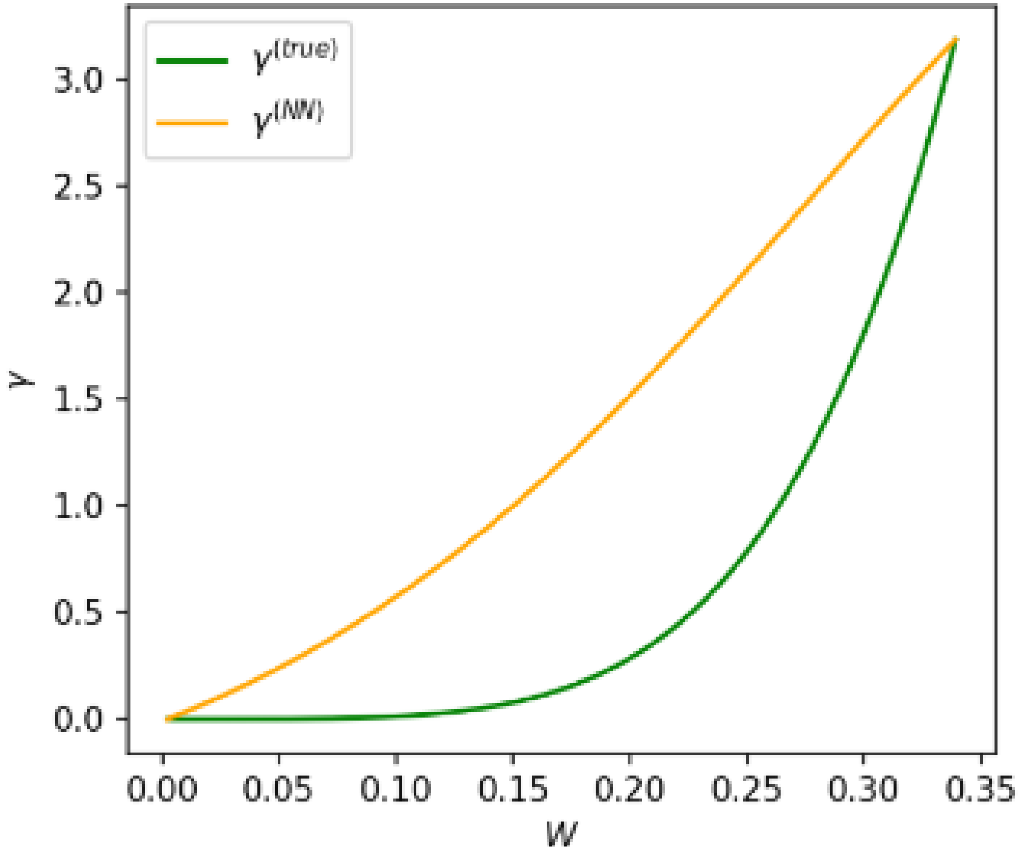
$RMSE(\frac{\partial W}{\partial t})$
7.7×10^{-4}

Neural network results

Third problem

$$\frac{\partial W}{\partial t} = \frac{\partial}{\partial z}\left(\lambda_W(W)\frac{\partial W}{\partial z}\right) + \frac{\partial \gamma_W(W)}{\partial z}$$

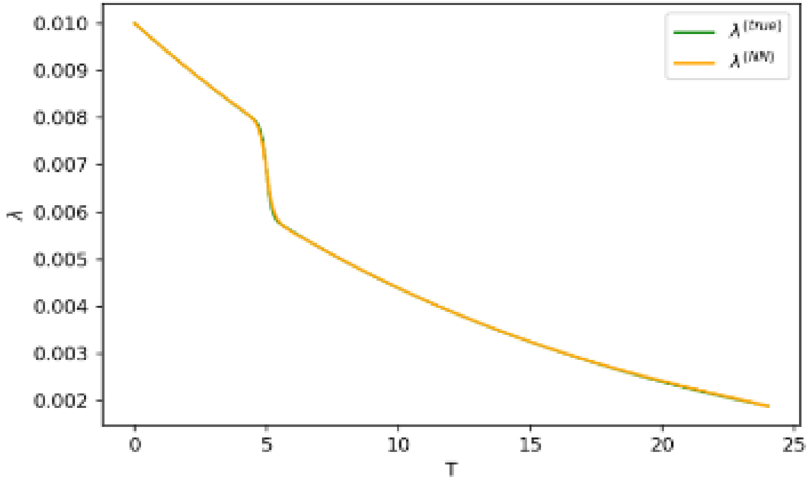




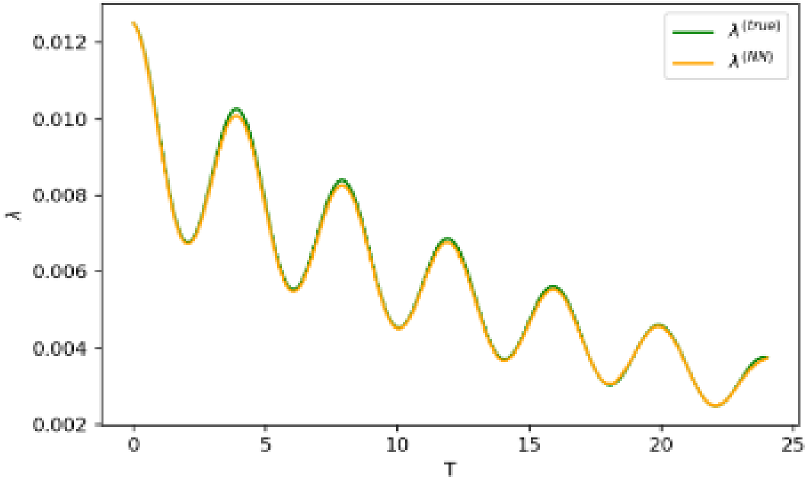
$MAPE(\lambda_W(W))$	$MAPE(\gamma(W))$	$RMSE(\frac{\partial W}{\partial t})$
4.4×10^{-2}	9.4×10^{-1}	3.0×10^{-4}

Mixed precision 16/32

(a)

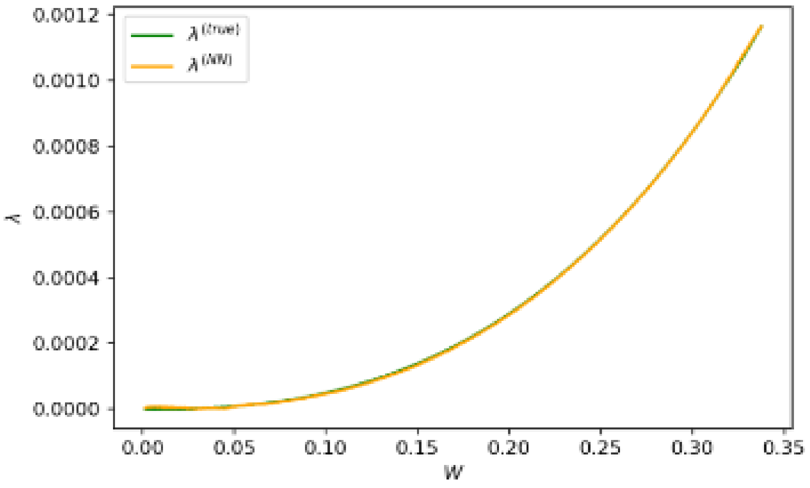


(b)



Partial “internal” fp16 calculus

(c)



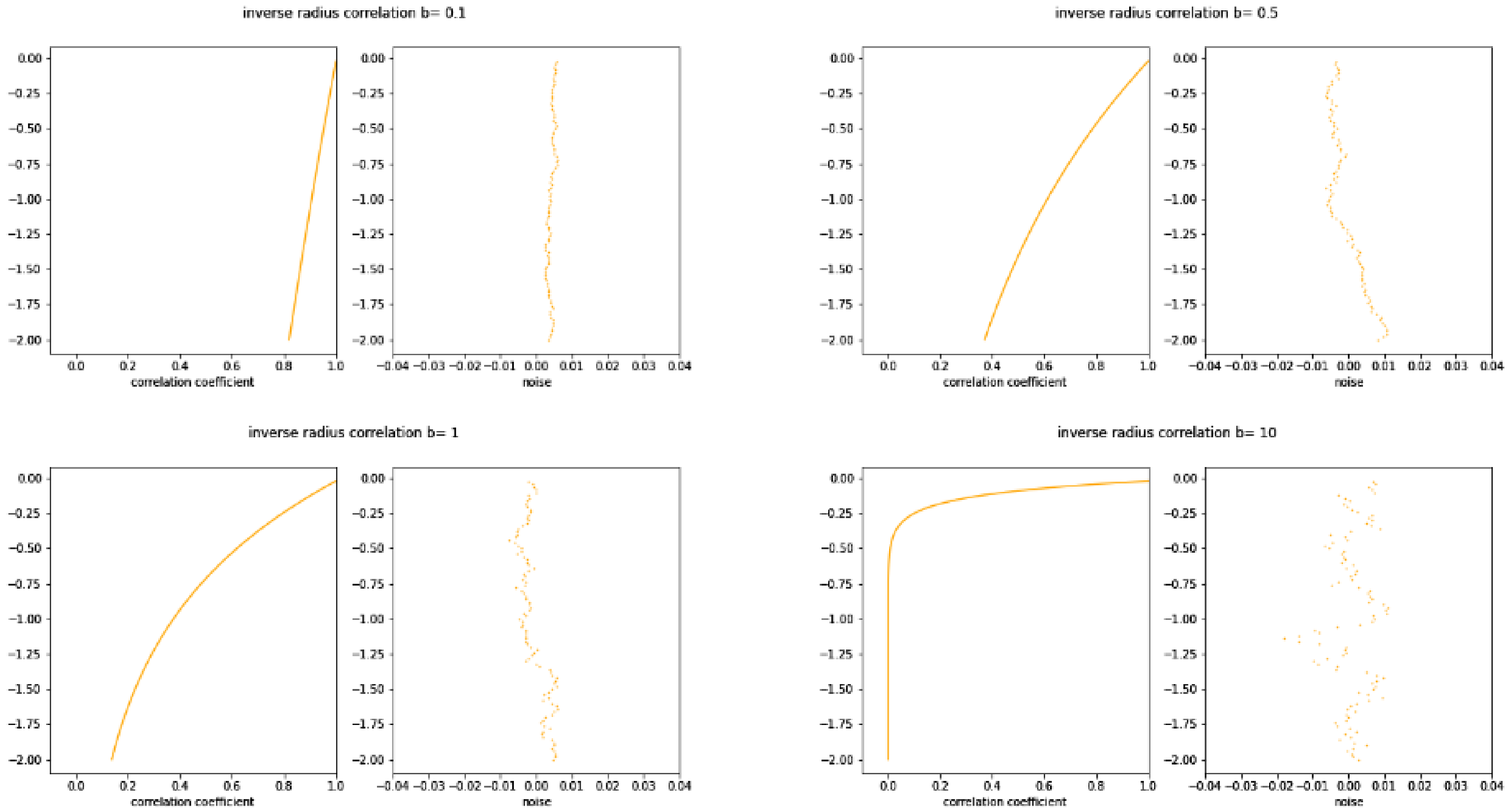
Did not lost in accuracy but increased in calculus speed

	$MAPE(\lambda_W(W))$	$RMSE(\frac{\partial W}{\partial t})$
a)	1.2×10^{-3}	2.0×10^{-1}
b)	1.0×10^{-2}	4.5×10^{-2}
c)	3.2×10^{-2}	1.5×10^{-4}

Sensitivity to uncertainty data

$$\overline{\epsilon_i \epsilon_j} = \sigma^2 e^{-b|z_i - z_j|}$$

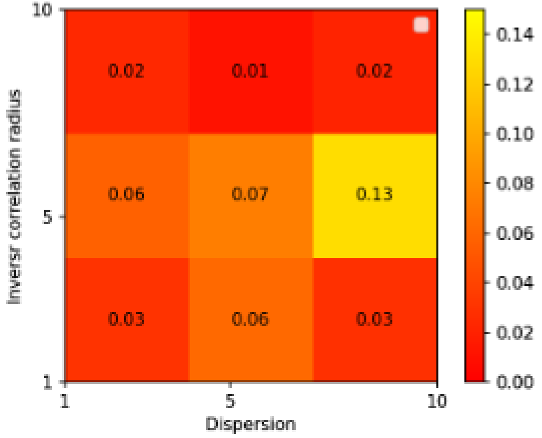
b – 1/Rc, Rc – correlation radius (м).

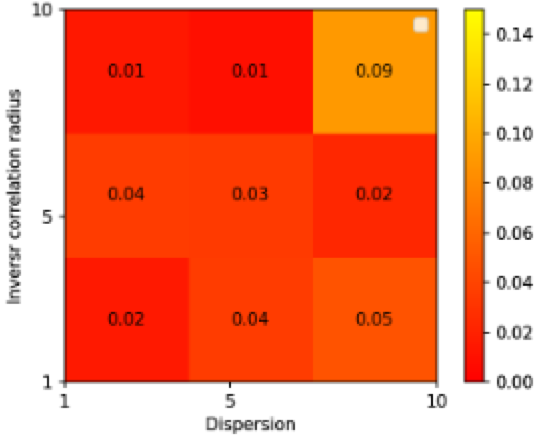


Correlation coefficient and vertical noise distribution

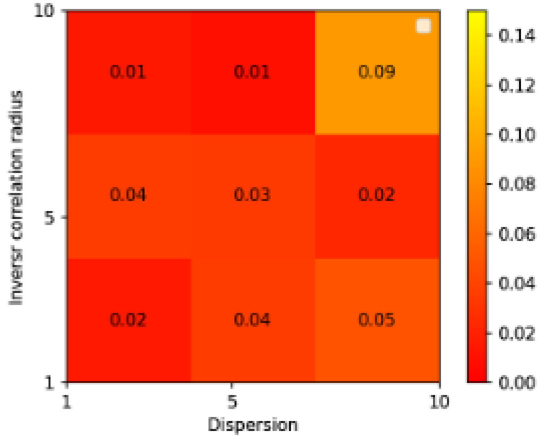
Sensitivity to uncertainty of input data. Results

- *Fixed learning iterations (epochs)*
- *Correlation radius variation*
- *Dispersion variation*





$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$



Doesn't have strict dependence between error correlation radius and loss

Conclusion and plans

- *Developed new algorithm for PDE identification. With proved quality on analytic model data.*
- *Doesn’t have strict dependence between error correlation radius and loss*

- *Further research and comparations based on Lomonosov MSU Meteorological Observatory data*
- *Further comparation whit conservative PDE solving methods*