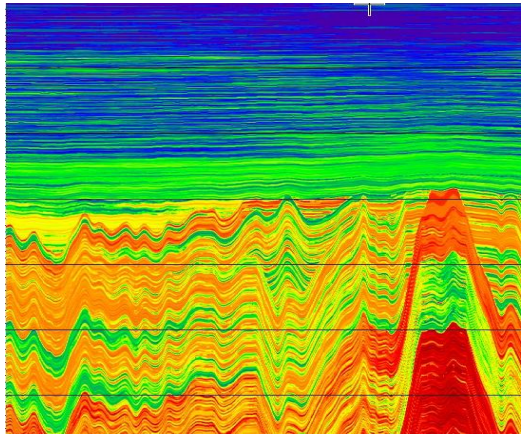
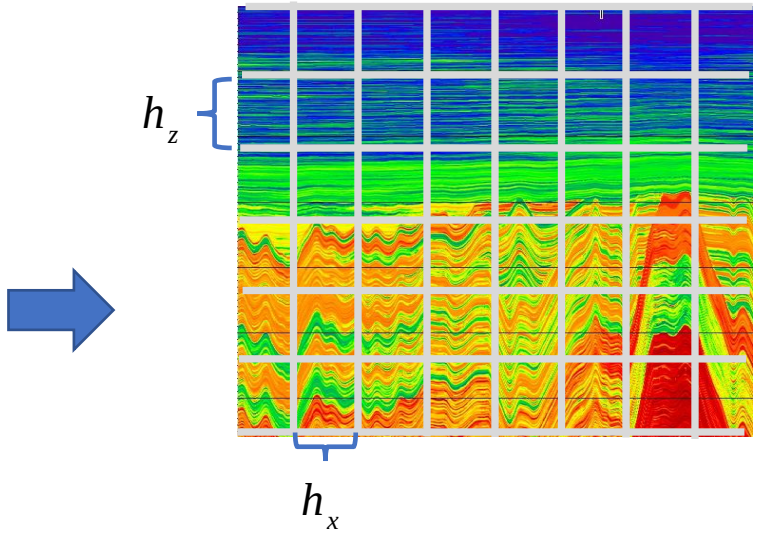
Seismic modelling with dispersion mitigation by machine learning

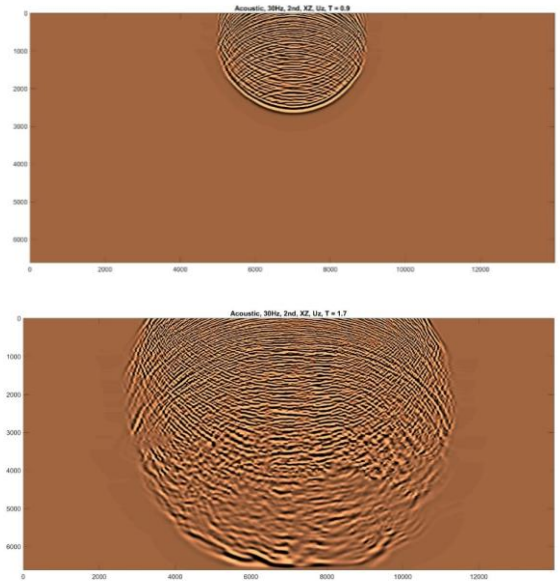
V. Lisitsa - Sobolev Institute of Mathematics SB RAS, Russia

K. Gadylshin, K. Gadylshina, D. Vishnevsky - Institute of Petroleum Geology and Geophysics SB RAS, Russia

Motivation







+

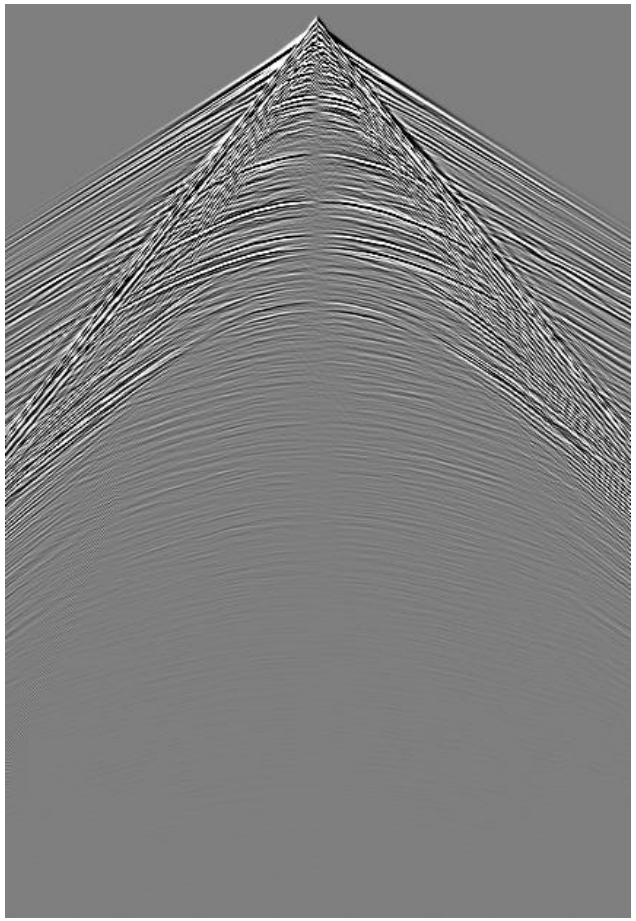
$$\rho \overset{\mathrm{r}}{u}_{tt} - \nabla \cdot \sigma = 0$$

$$\varepsilon = 0.5(\nabla \overset{\mathrm{r}}{u} + \nabla \overset{\mathrm{r}}{u}^{T})$$

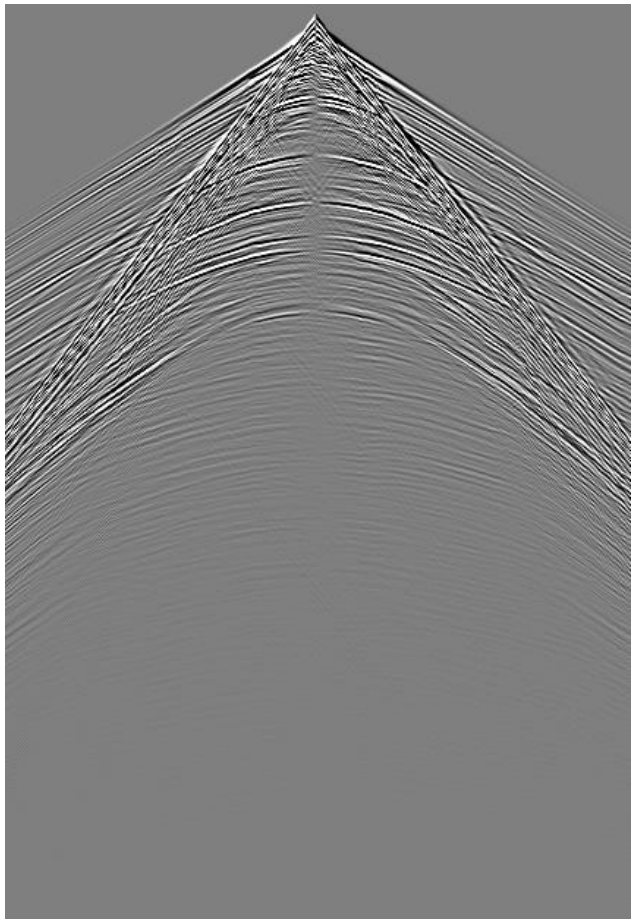
$$\sigma = C\varepsilon$$

Motivation

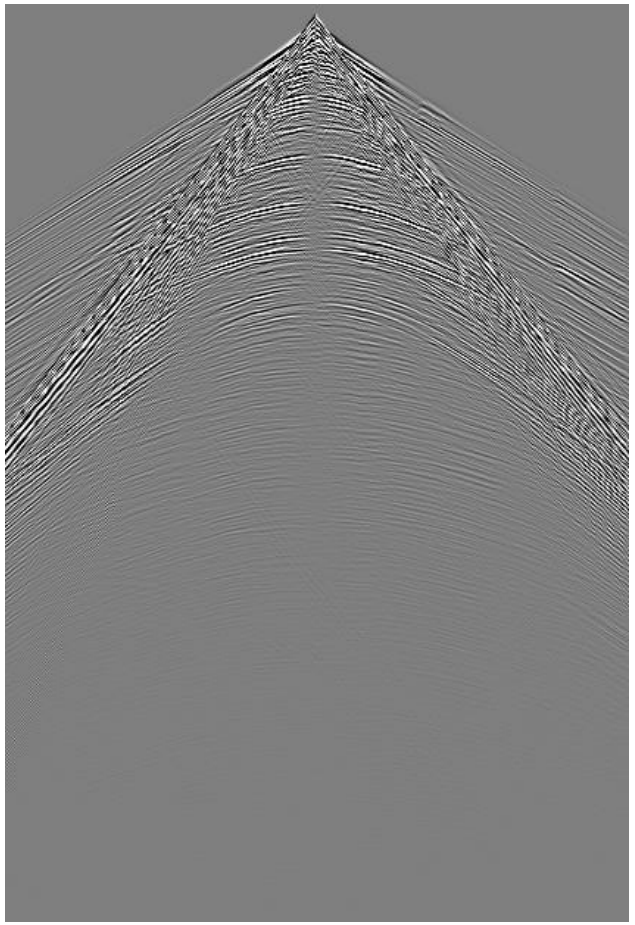
ux h=2.5m



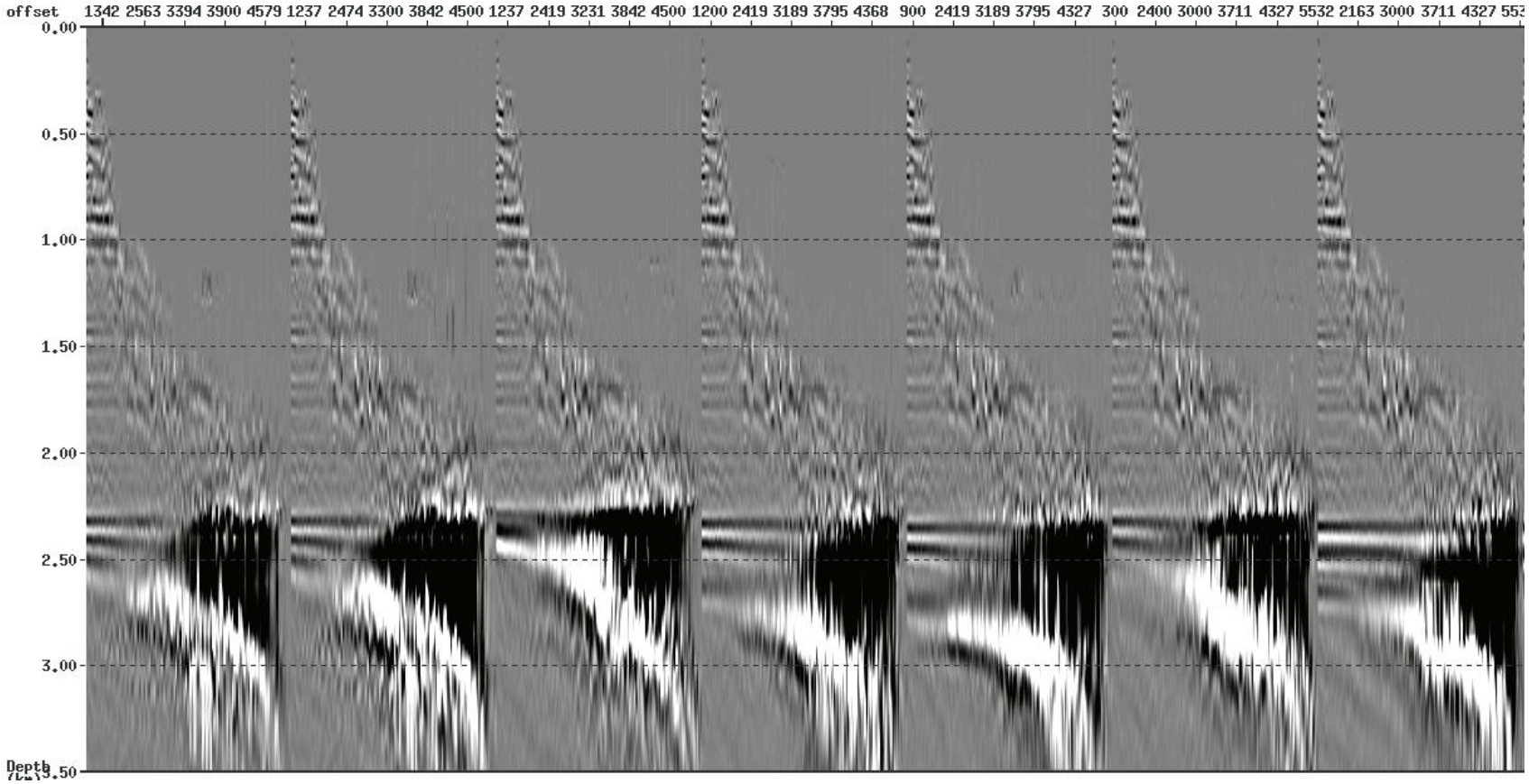
ux h=10m



difference



Motivation

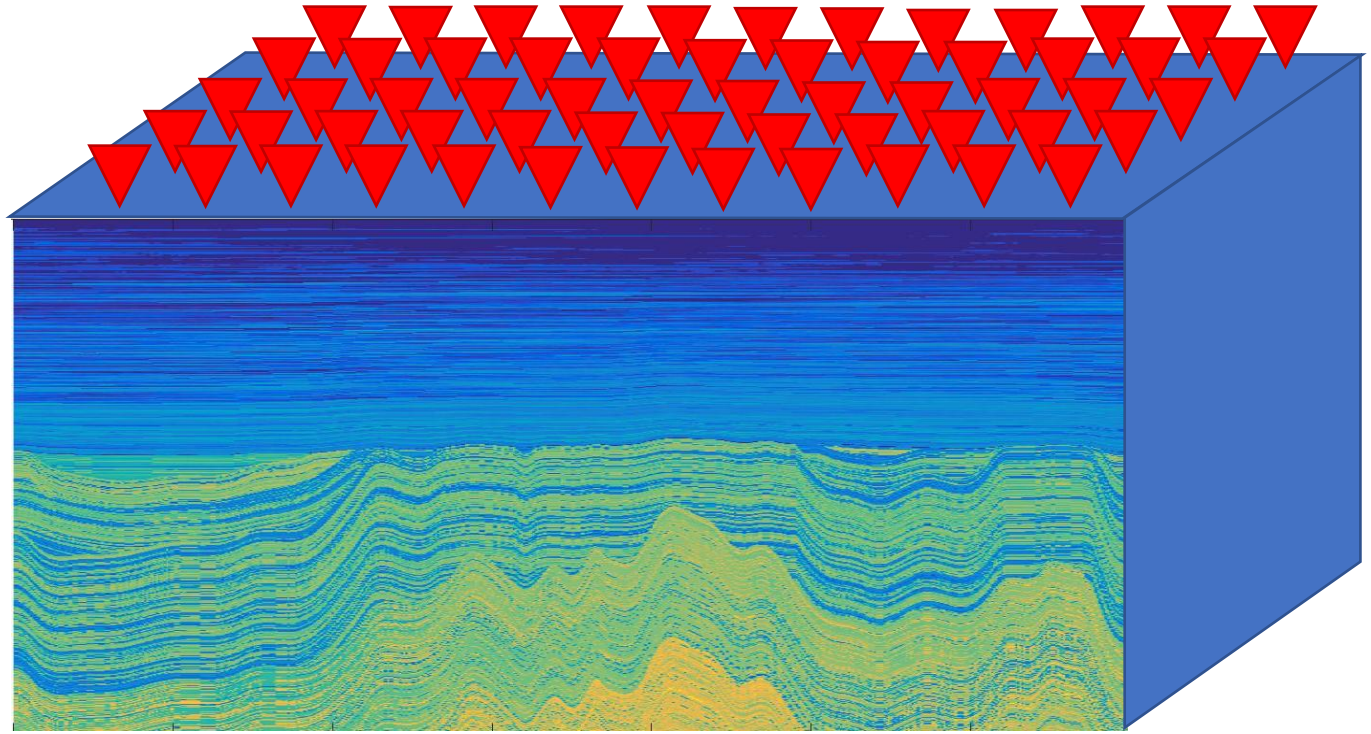


Due to the numerical dispersion induced phase shift

Motivation – ways to improve accuracy

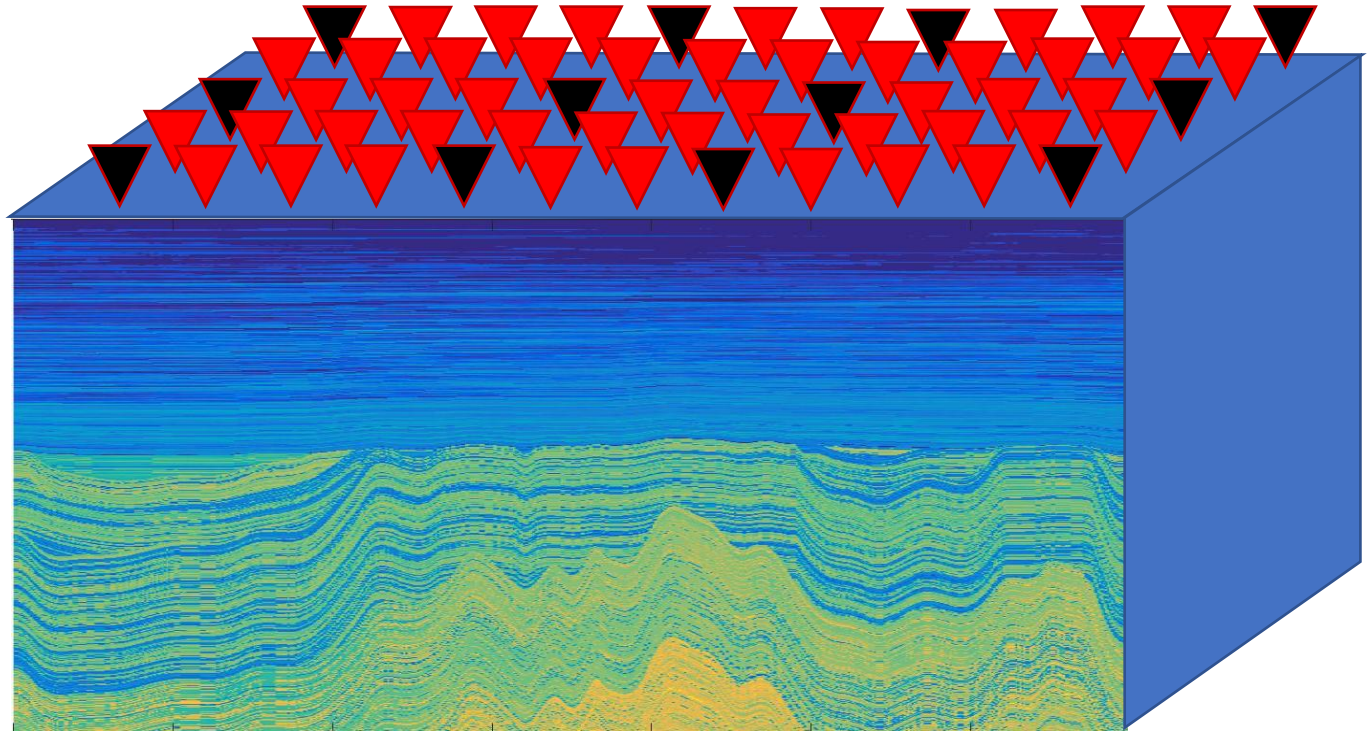
1. Change the numerical method
   - increase in flops, thus in computational time
   - accuracy is limited
2. Reduce the grid step
   - theoretically accurate solution can be obtained
   - Dimensions curse – reduction of a grid step by the factor of 2 increases the number of grid points by the factor of 8, and computational time by 16!
3. Post-processing
   - Dispersion depends on the way the signal travelled

Seismic modeling



Do we need to compute accurately all common-shot gathers?

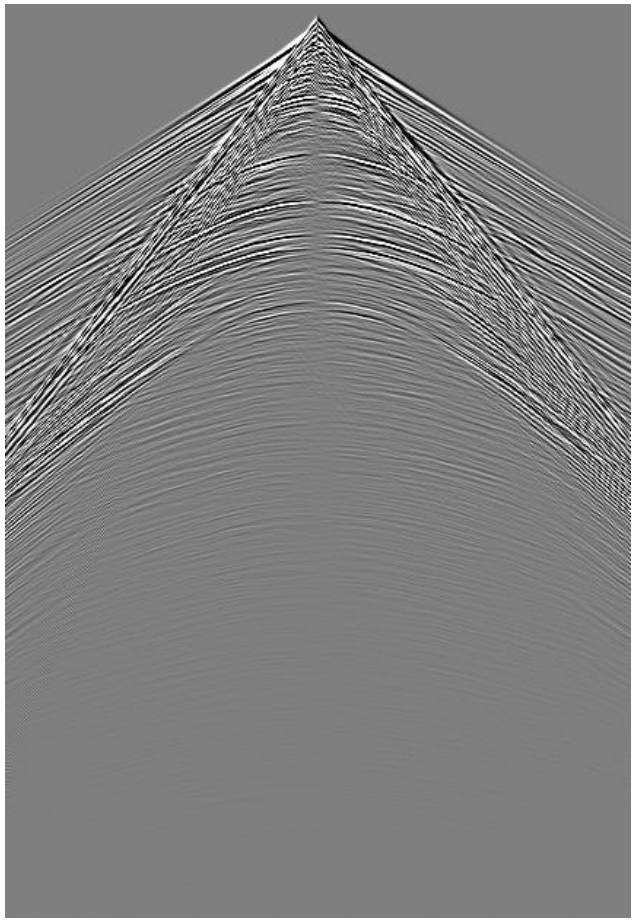
Seismic modeling



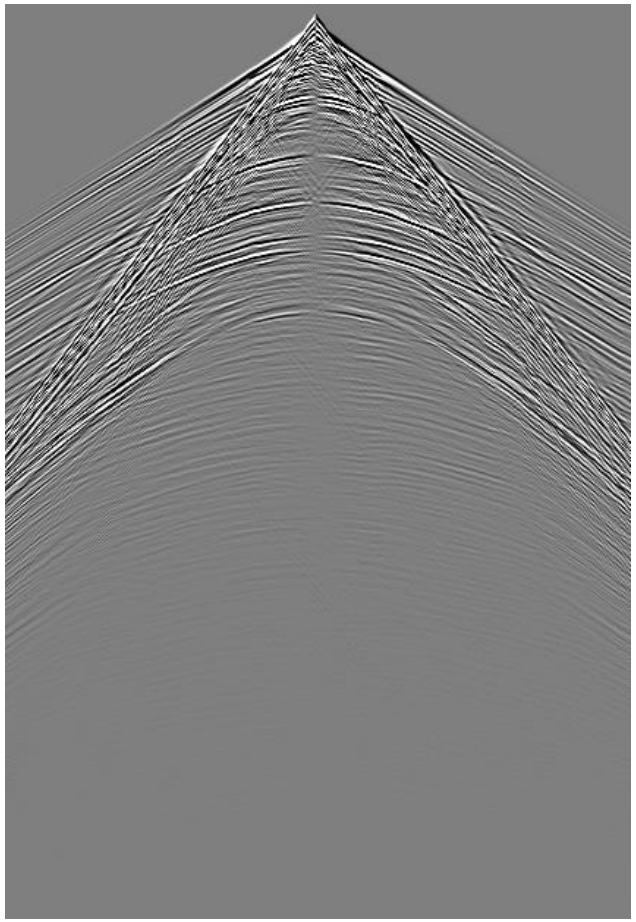
1. Compute all common-shot gathers inaccurately using coarse mesh
2. Compute few gathers with high accuracy – training set
3. Train neural network to suppress dispersion
4. Apply NN to improve data

Modelled data example

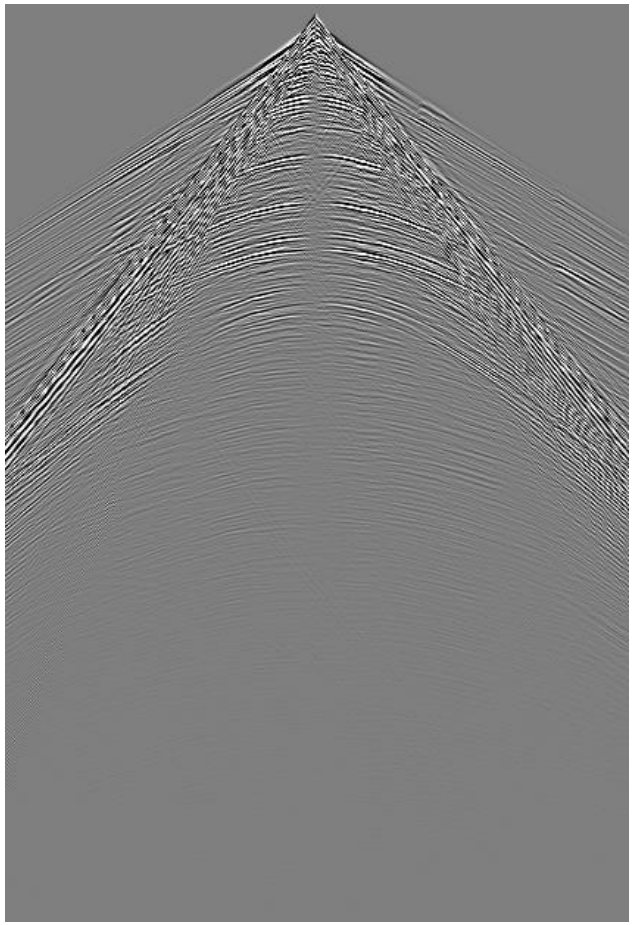
ux h=2.5m



ux h=10m

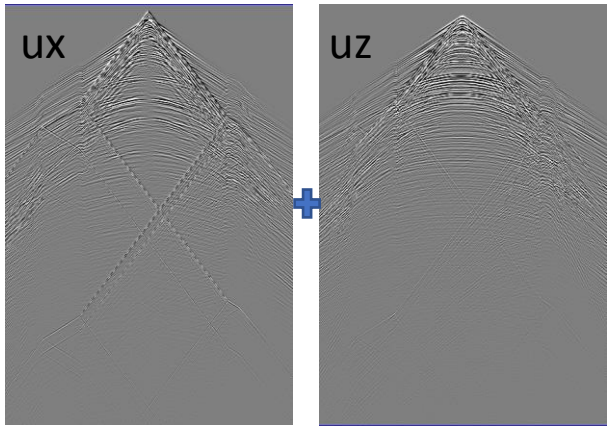


difference

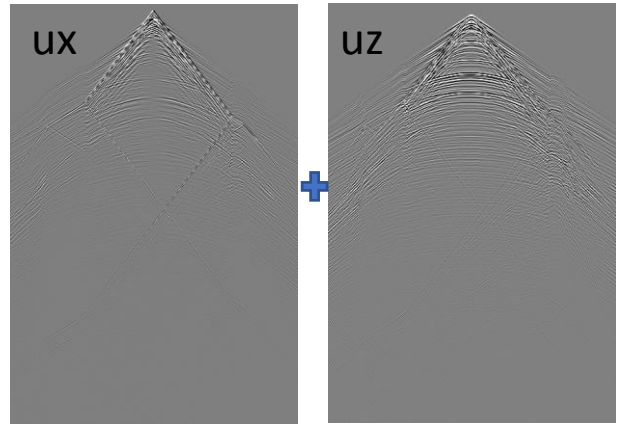


Numerical **D**ispersion **M**itigation deep neural **NET**work (NDM-NET)

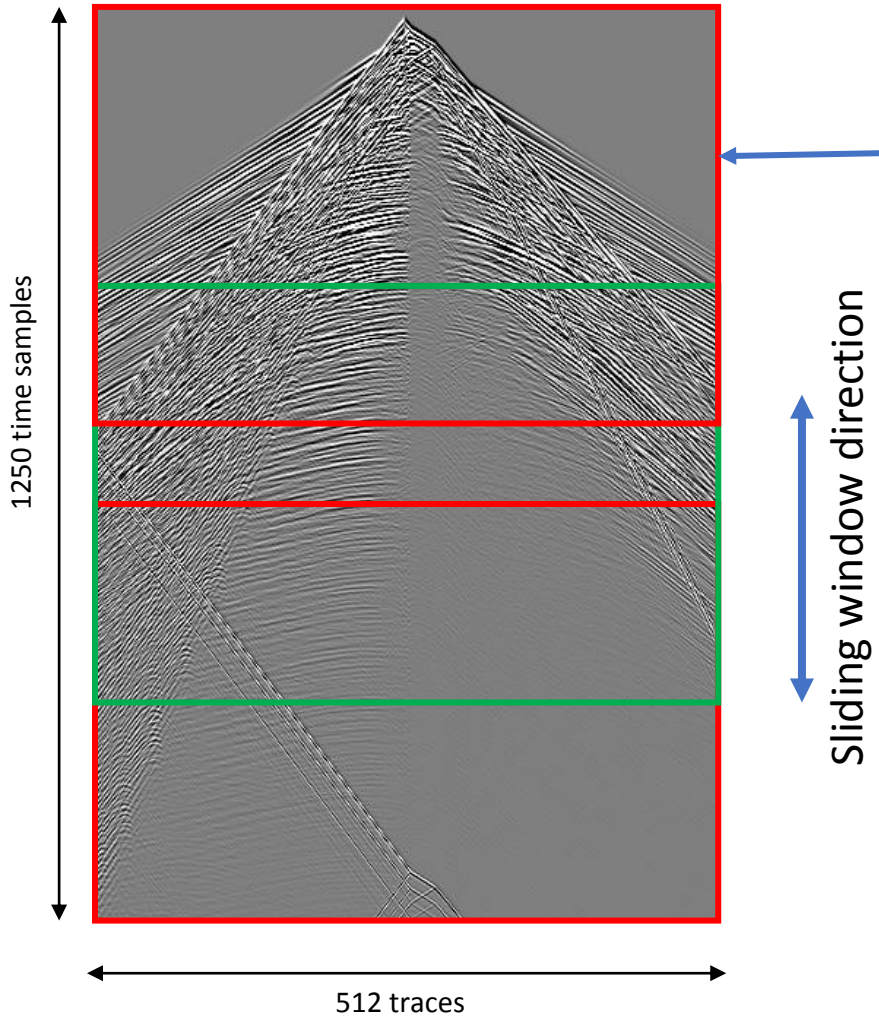
Data with numerical dispersion



Data **without** numerical dispersion



Creating the training dataset



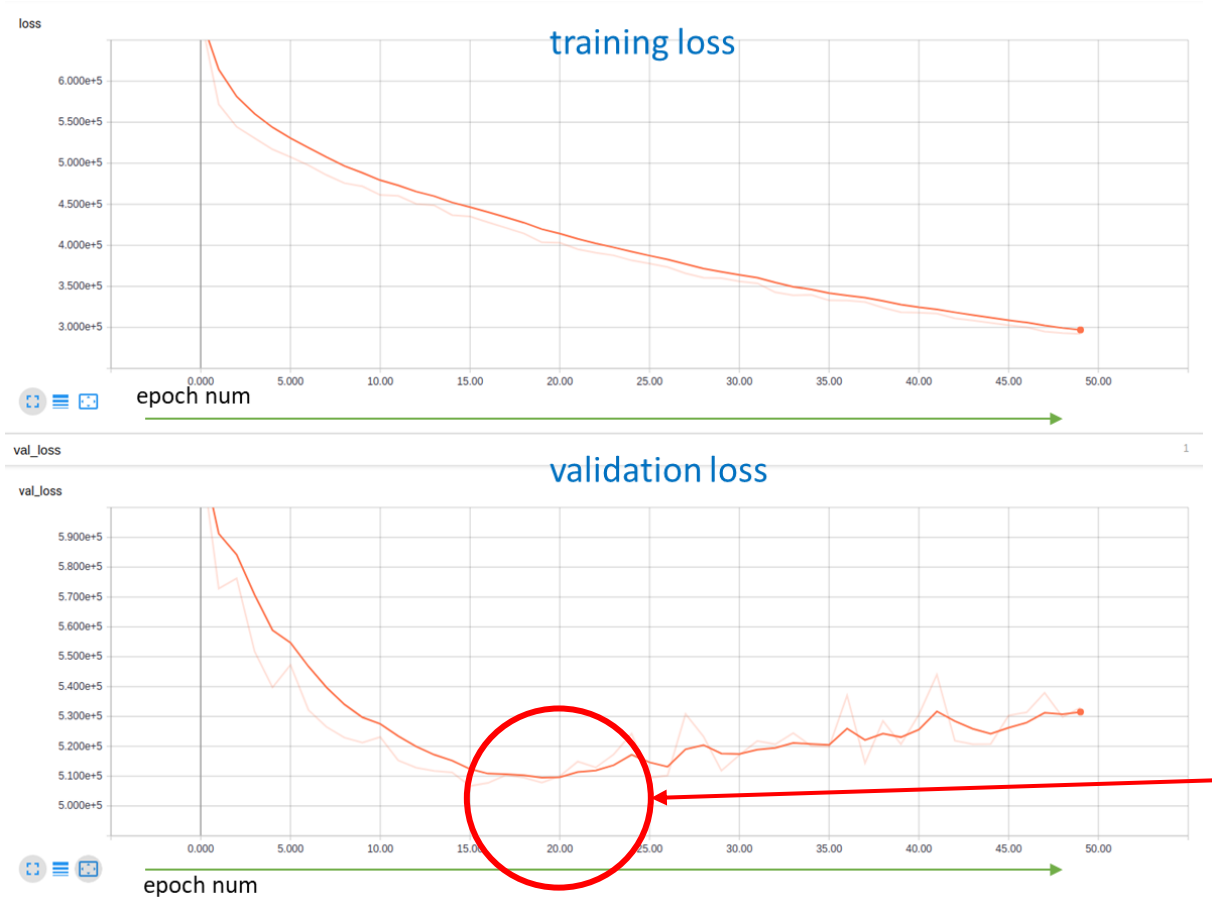
Each seismogram is converted to a **1250x512x2 tensor**:

1250 time samples
512 traces per shot
2C receiver

Then each tensor is splitted onto **23 patches** using the **sliding window 512x512 pixels with 32 pixel stride**

Finally, the input of NDM-net is **512x512x2 patch**

NDM-net training

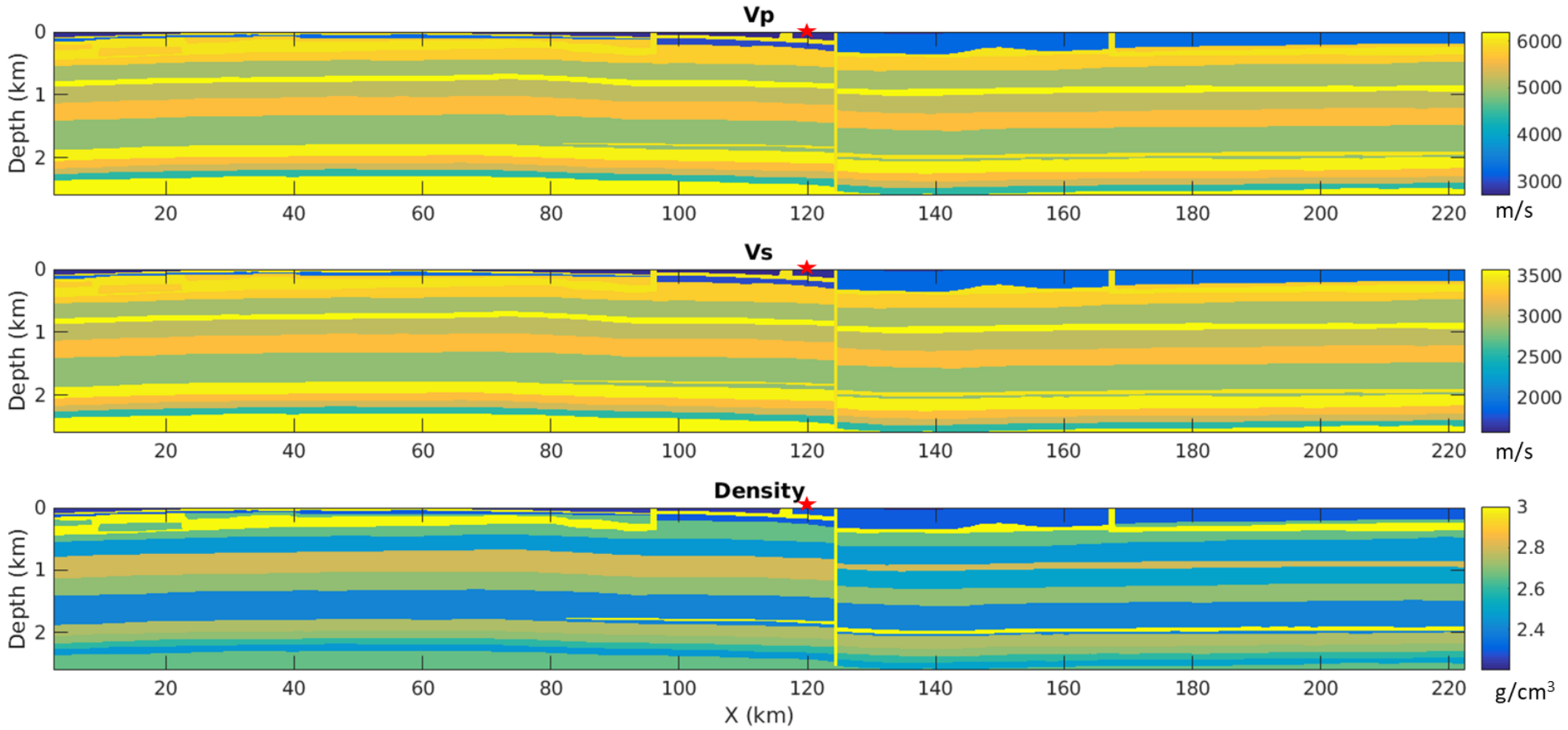


The training took about **40min**

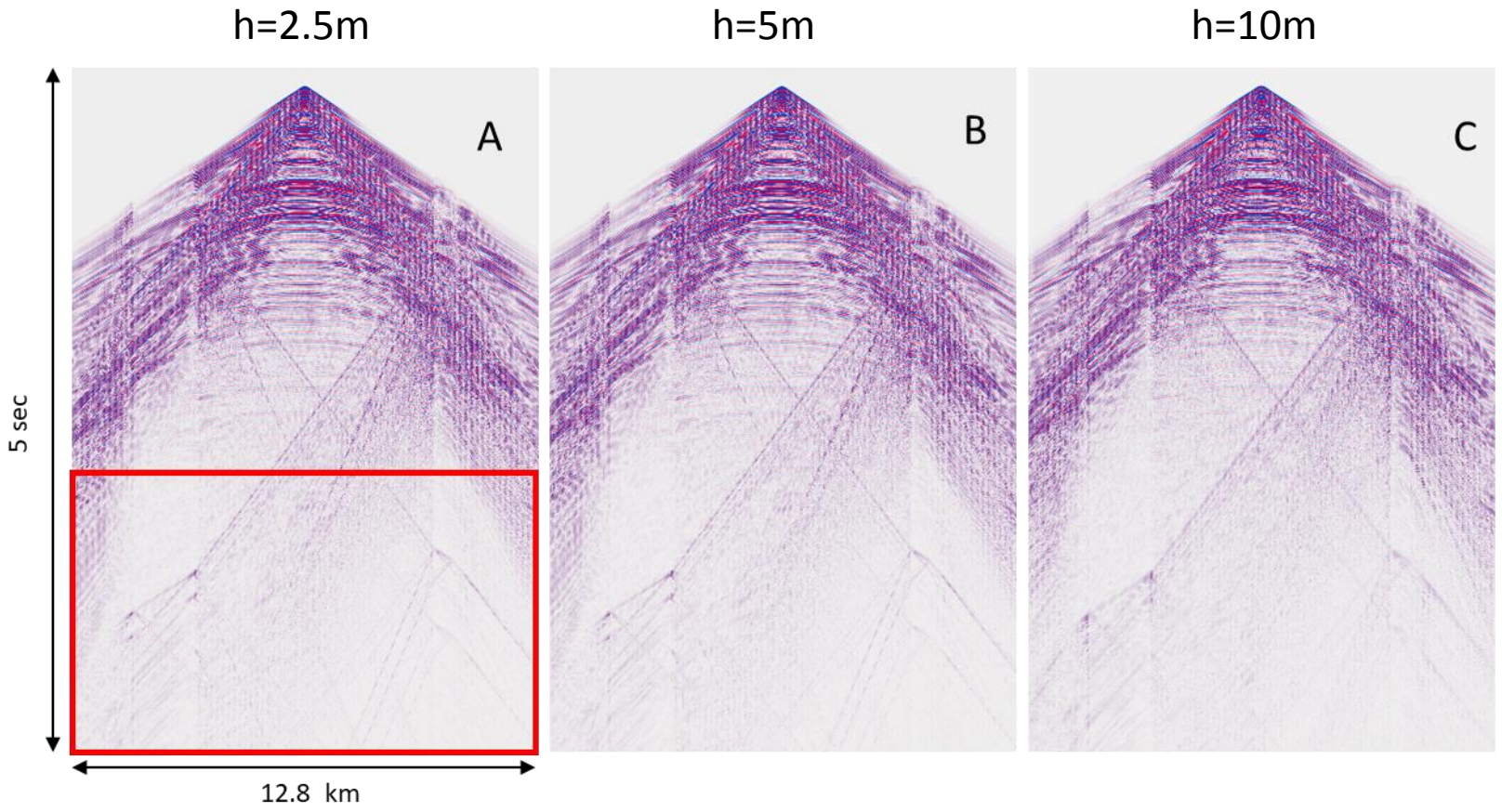
Finally, the DNN weights after 20th epoch were used for data prediction

Early stopping should be applied here

Model and acquisition!

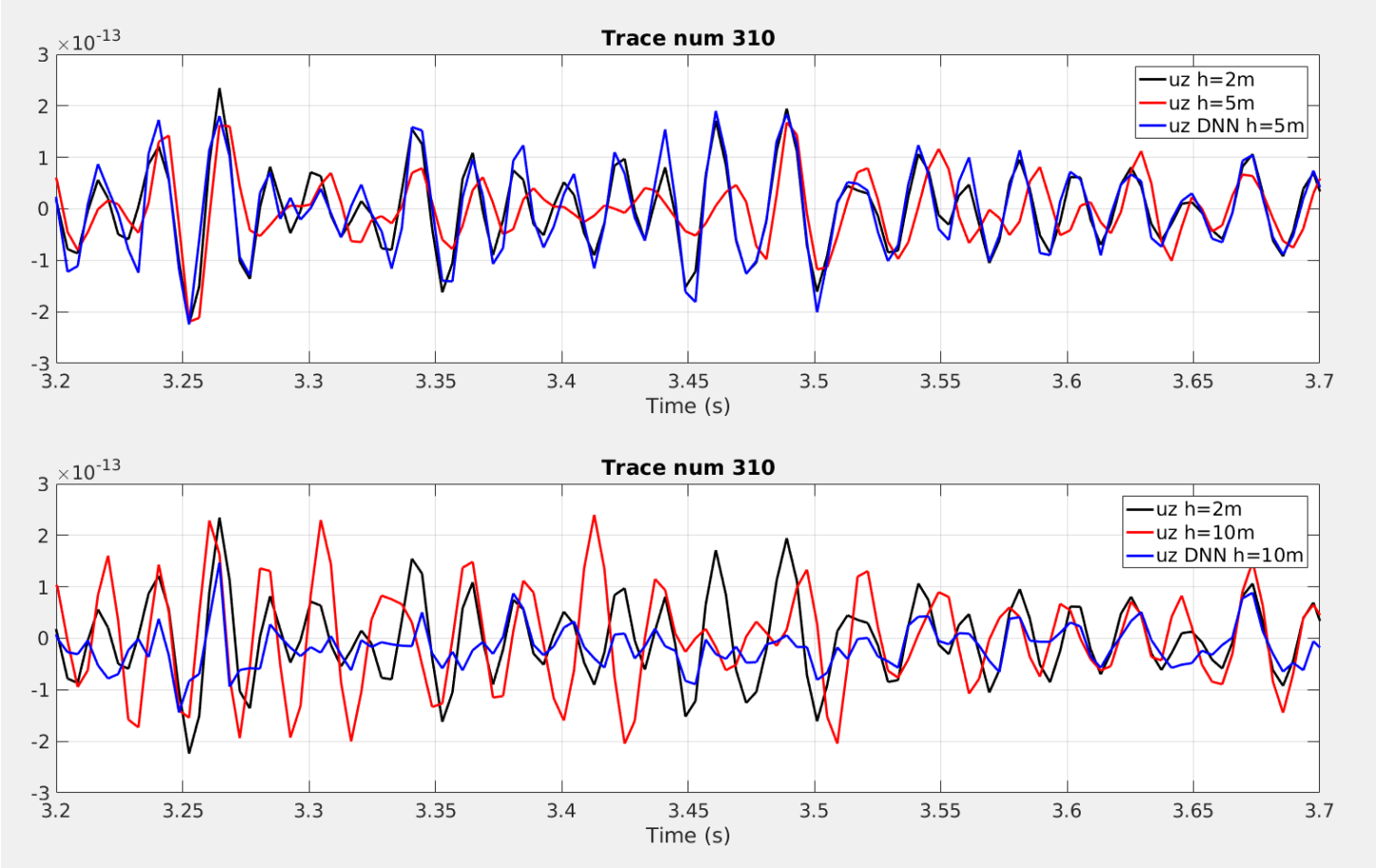


Data (vertical component)



Mitigating the numerical dispersion

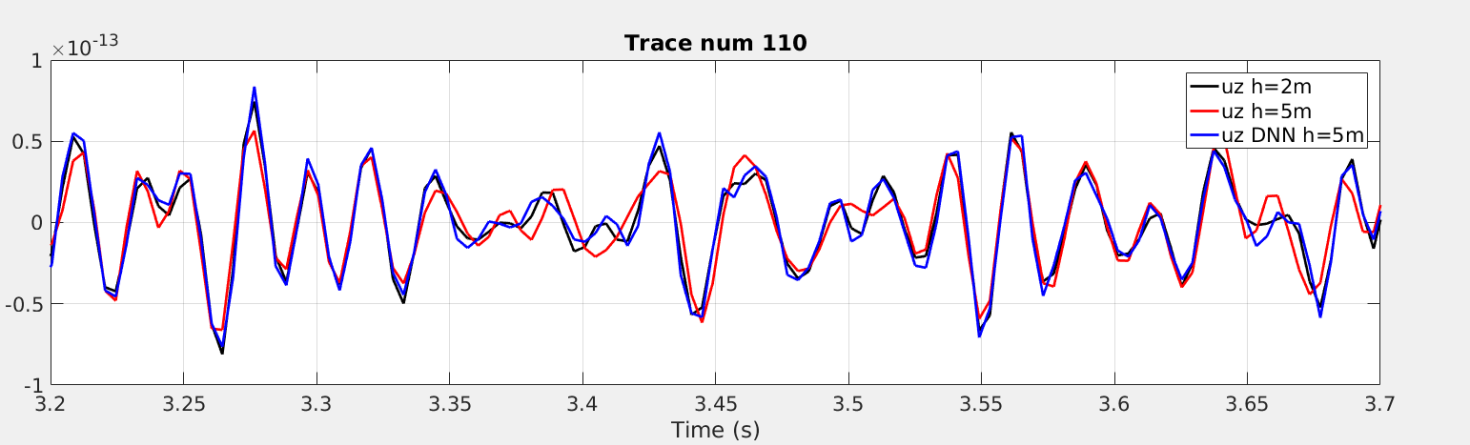
h=5 to h=2



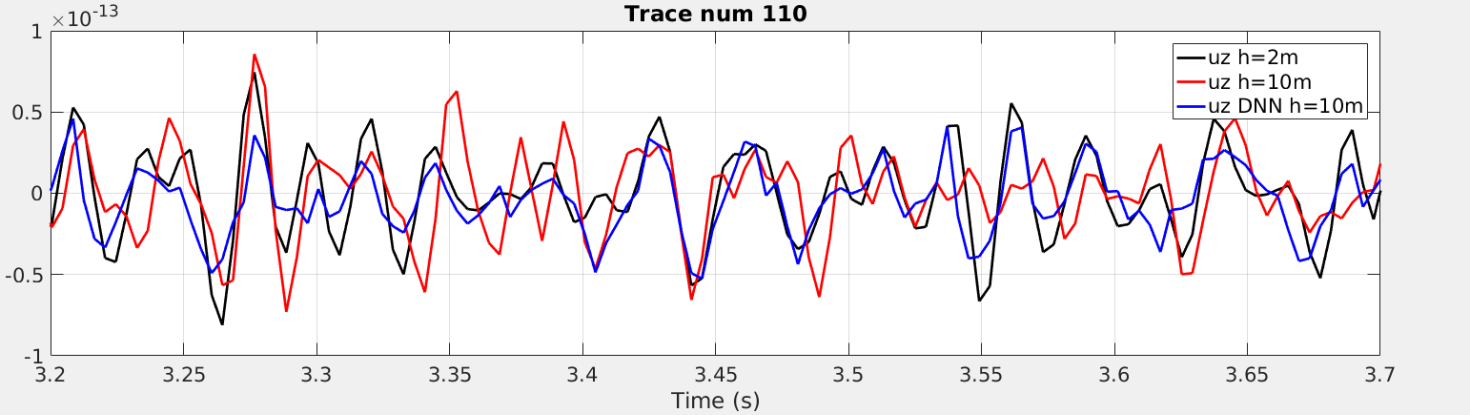
h=10 to h=2

Mitigating the numerical dispersion

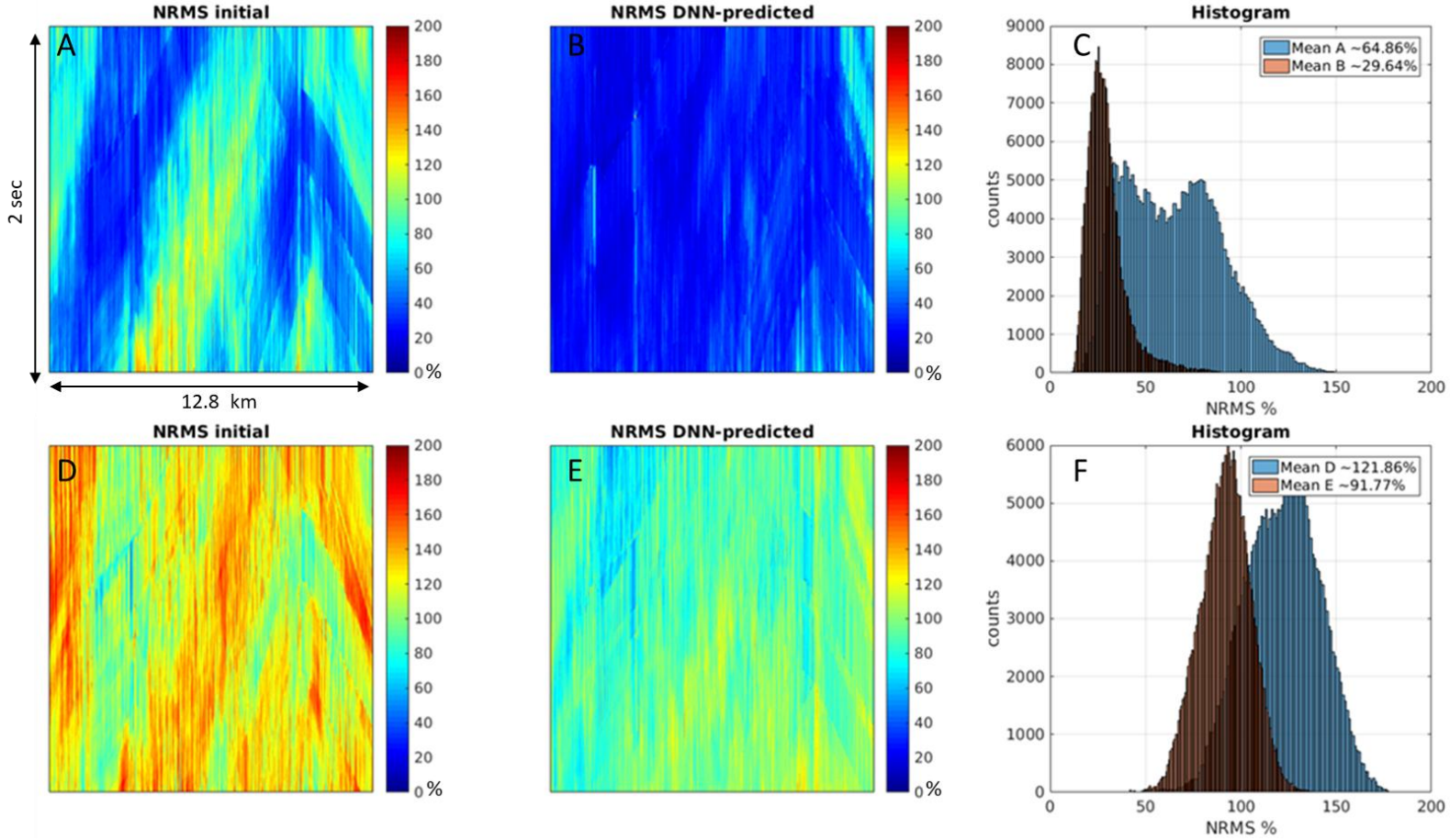
h=5 to h=2



h=10 to h=2

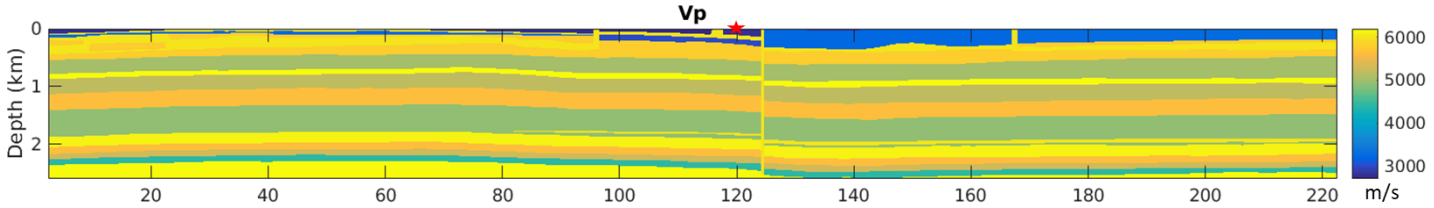


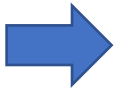
NRMS plots

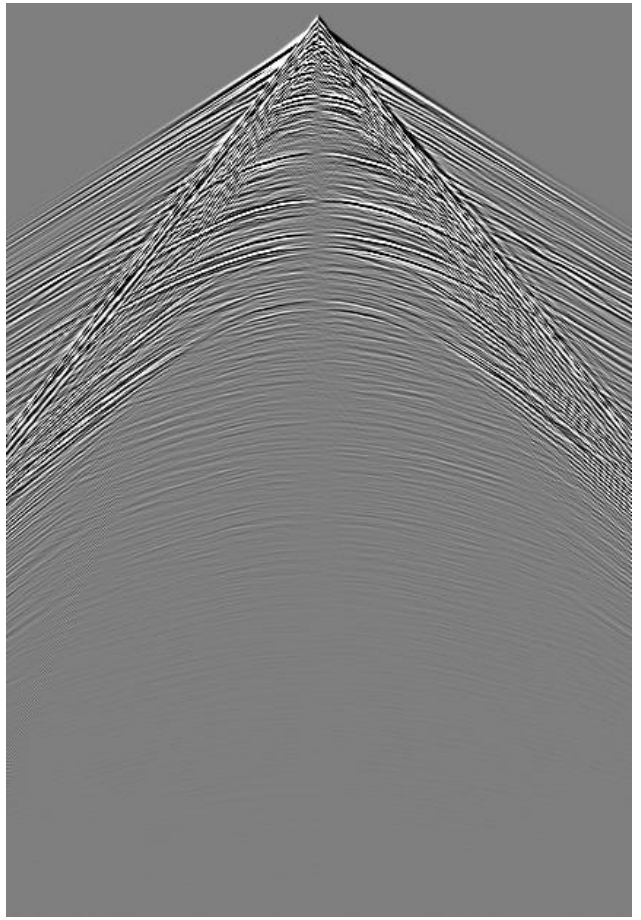

NRMS initial
A
2 sec
12.8 km
NRMS DNN-predicted
B
Histogram
C
Mean A ~64.86%
Mean B ~29.64%
counts
NRMS %
NRMS initial
D
NRMS DNN-predicted
E
Histogram
F
Mean D ~121.86%
Mean E ~91.77%
counts
NRMS %

Performance

1900 sources





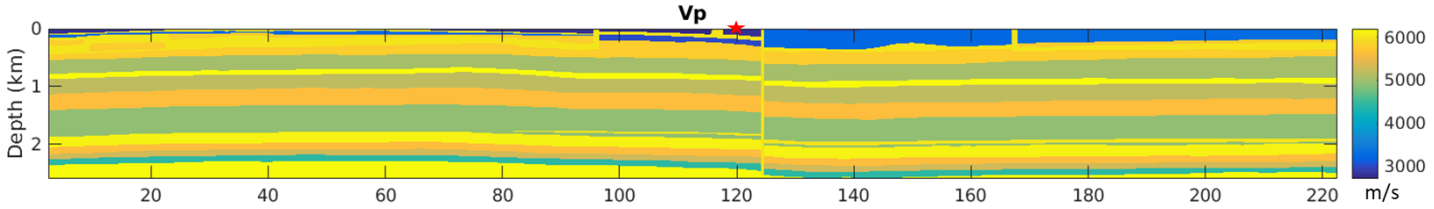


Time to compute ONE common-shot gather:
2x2 m – 43 seconds
5x5 m – 6 seconds
10x10m – 1.2 seconds

Time to compute ALL common-shot gather:
2x2 m – 23 hours
5x5 m – 2.3 hours
10x10m – 0.6 hours

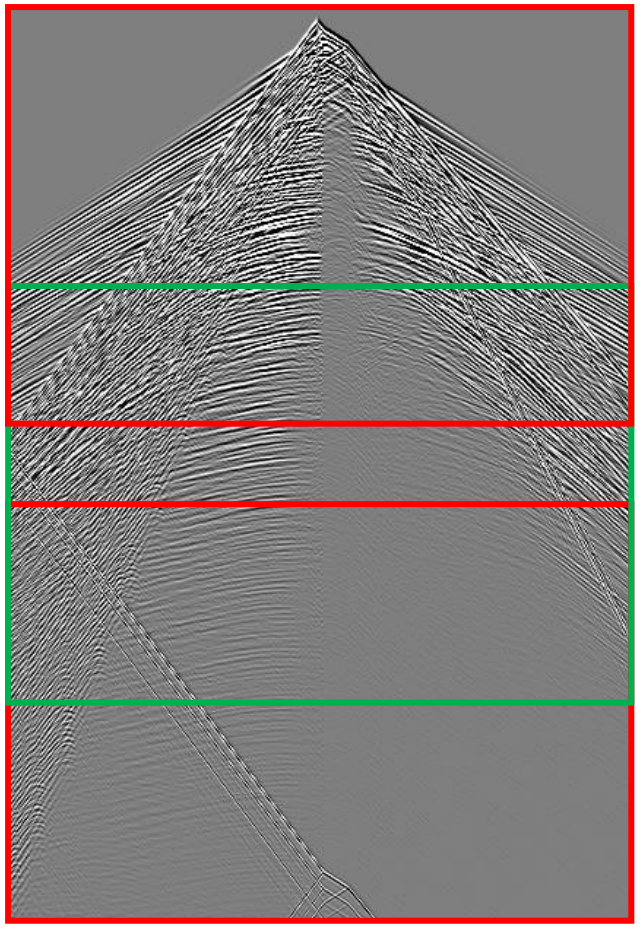
Performance

1900 sources



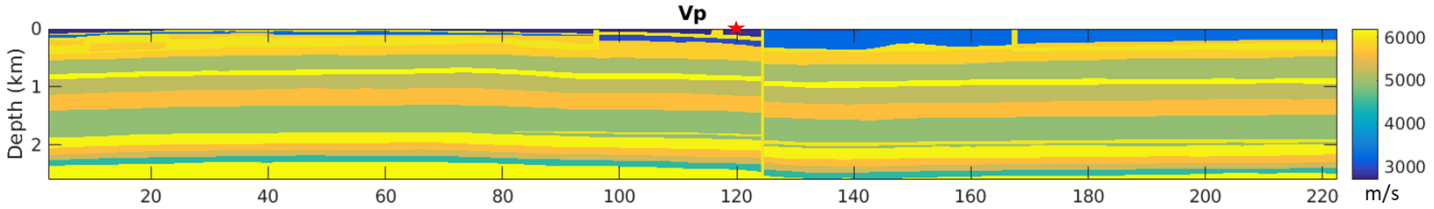
Gathers used for training – 190
Training time (total) – 40 min
Correction time – 0.1 second per gather

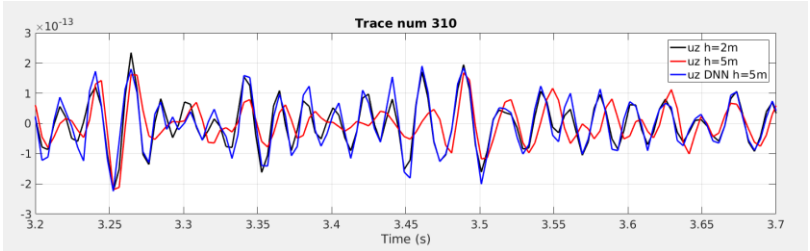
Training

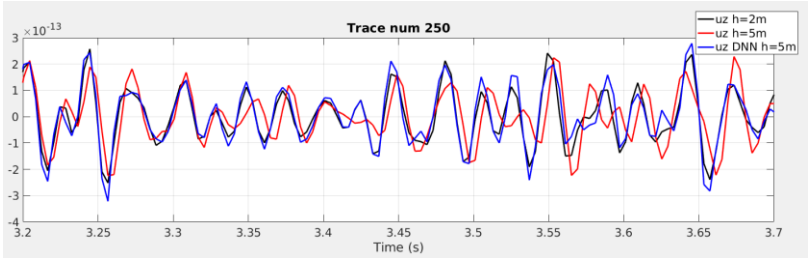


Performance

1900 sources







Total time **finite difference modelling**:
Entire acquisition at fine grid: **23 hours**

Total time **f-d + NDM-net**:
Entire acquisition at 5 m grid: 3 hours
Training set generation: 2.3 hours
Training: 0.6 hours
Total: 6 hours

Conclusions and Road map

- The combined **DNN-based approach** for seismic modelling using the **NDM-ne**t is proposed
- The **proof of a concept** was done using a complex 2D elastic model
- The presented results demonstrate the ability of NDM-net to make a **high-quality** seismic **data prediction** using the synthetics generated on a coarse grid
- Our future investigations are focused on **3D seismic modelling**, where the computational time becomes critical

Thank you for your attention!