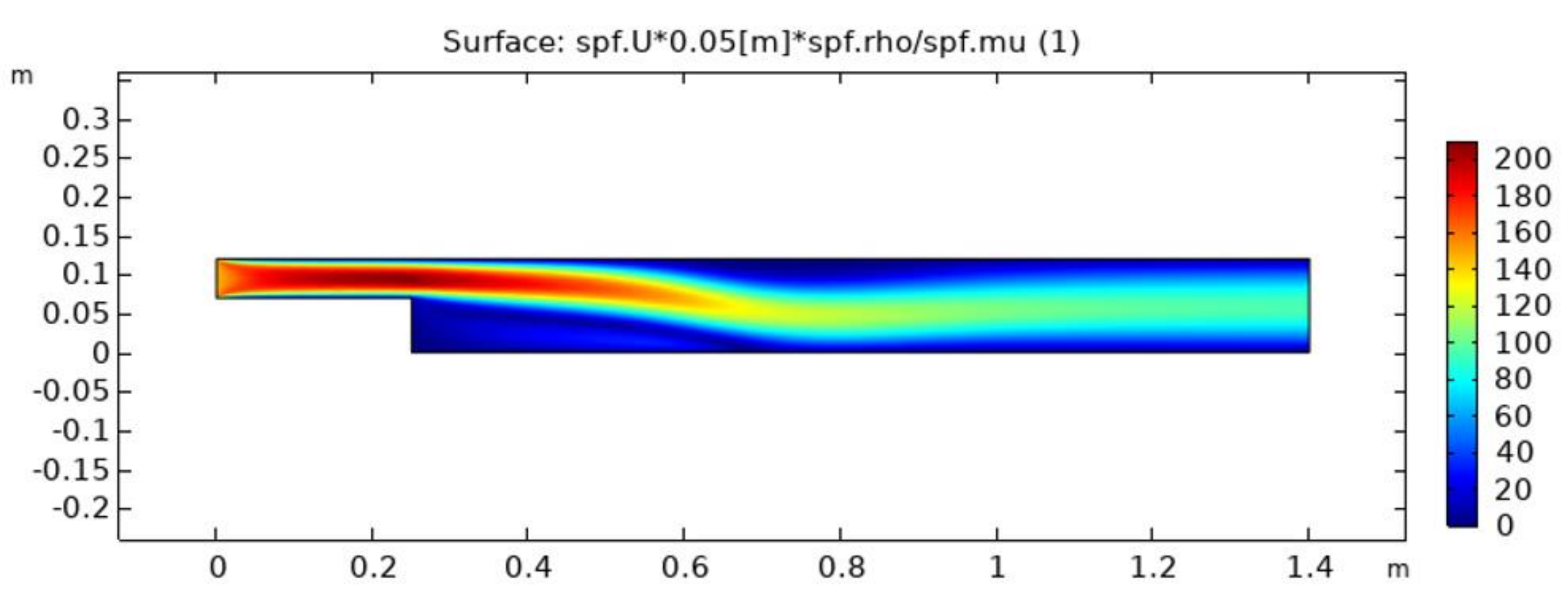
Predicting the Optimal Solver Settings with Machine Learning in the COMSOL CFD Module

25-05-2022

TU Delft COMSOL



Solver: Automatic Newton

Inflow velocity: 0.03 m/s

Initial velocity: 0 m/s

How can you find the optimal CFL number for pseudo time-stepping with the use of machine learning?

Content

- Literature
- CFL and pseudo time-step
- Neural networks
  - Predict target optimal CFL number
  - Minimize the residuals
- Further research

TU Delft

Literature

- Neural network multigrid solver [1]
  - Network corrects interpolated solutions on fine grids
  - Uses patches and local data → generalizable
  - Reduces computational time with same or higher accuracy

Literature

- Detecting troubled-cells [2]
  - Troubled-cells → Gibbs oscillations
  - Existing TVB methods require problem-dependent parameters
  - Neural network does not
  - Input contains local information
  - Output flags troubled cells
  - Outperforms traditional methods in term of solution accuracy

TU Delft

CFL and Pseudo time-step

- Pseudo time-step
  - Initial guess not so important

$$\frac{\partial \mathbf{u}}{\partial t} = F(\mathbf{u}) \qquad \Delta\tilde{t}_n = \mathrm{CFL}_{\mathrm{loc}}(n)\frac{h}{||\mathbf{u}||}$$

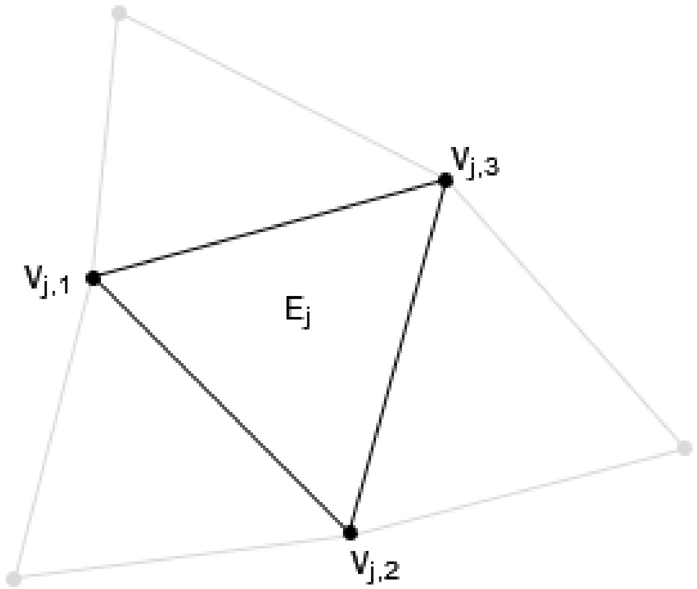
- Two different CFL numbers in COMSOL
  - One uses iteration count of Newton step
  - Other uses nonlinear error estimate
  - Both are global values

Neural networks

- Local data for generalization
  - Data from one element only
  - Data from a patch of four elements
- Two networks
  - With optimized CFL target
  - With objective to minimize residuals

Local data: single element

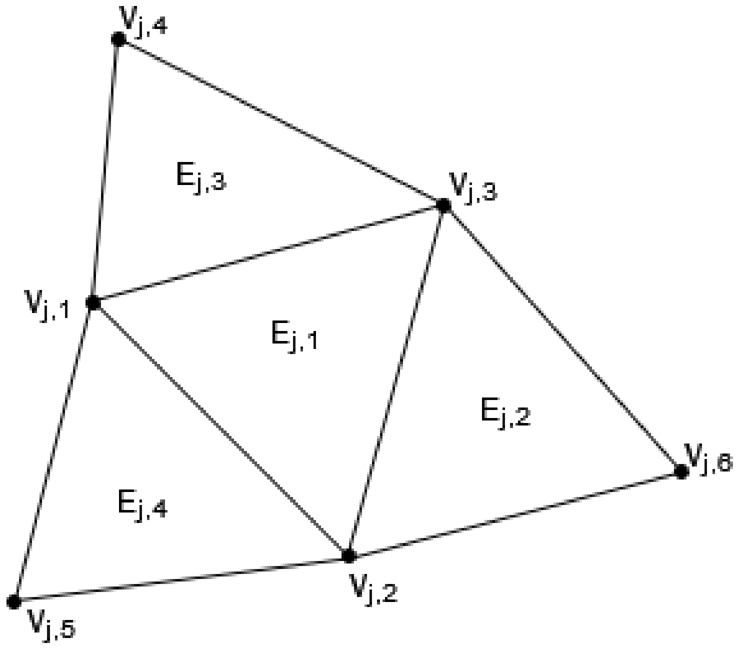
- Vertices
  - u, v, p, residuals
- Centroid
  - Reynolds number
- Edge lengths
- Iteration count
- 32 inputs



TU Delft

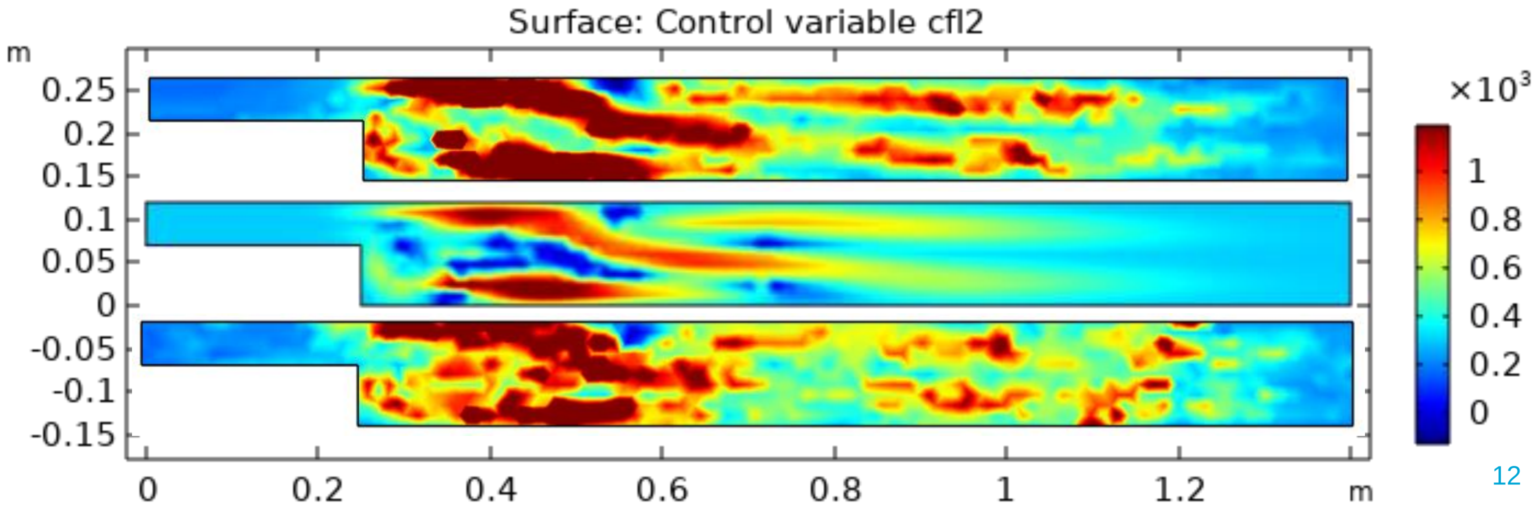
Local data: patch

- Vertices
  - u, v, p, residuals
- Centroids
  - Reynolds number
- Edge lengths
- Iteration count
- 125 inputs



Network: optimized CFL target

- Optimized CFL number in COMSOL
- Loss is MSE
- 6 hidden layers; 256 neurons

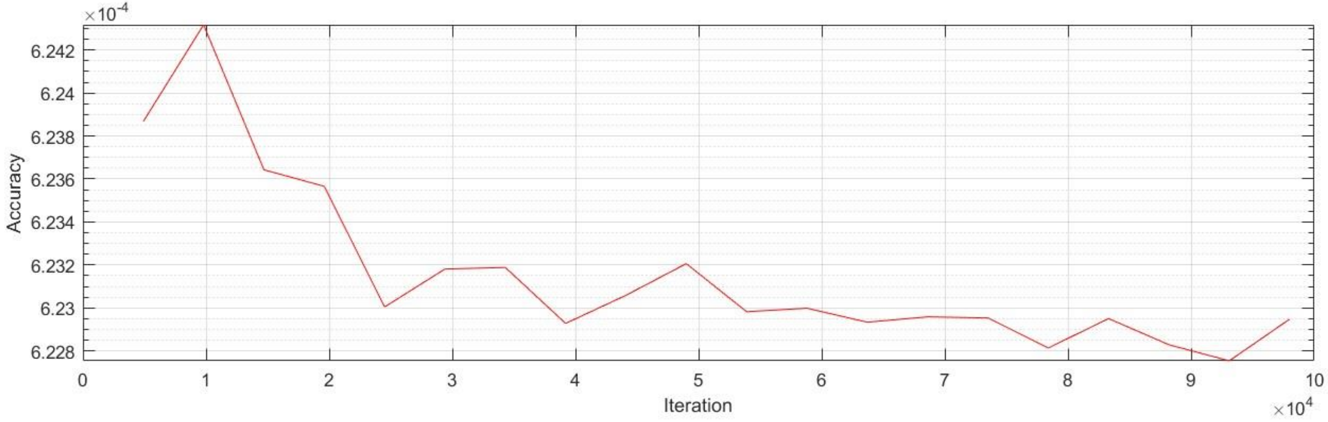


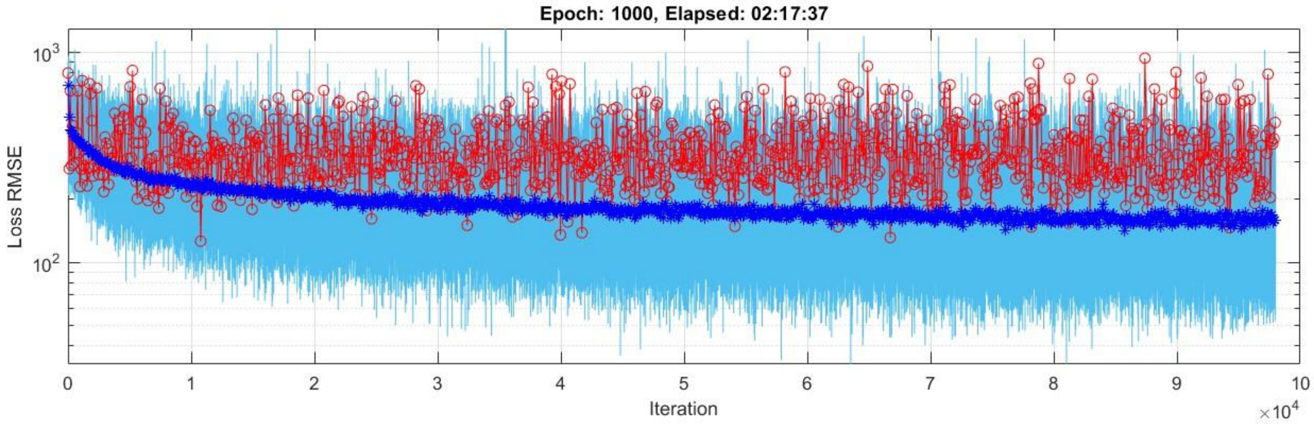
Literature

CFL and pseudo time-step

Neural networks

Further research





TU Delft

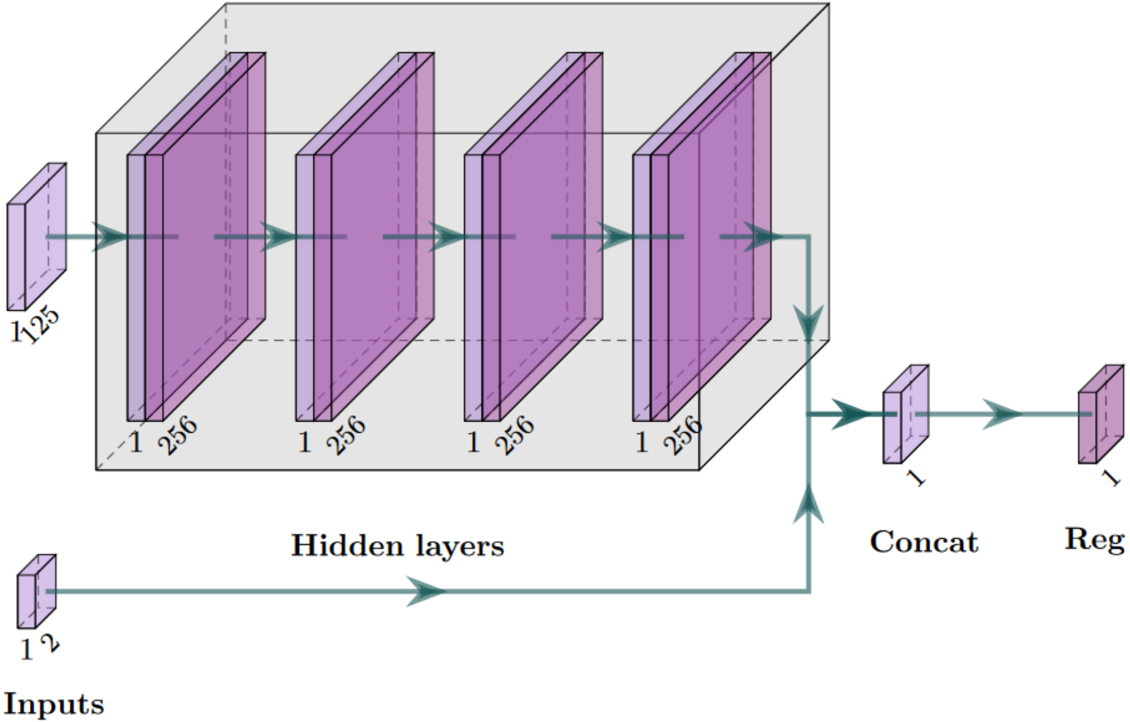
Network: minimize the residuals

- Loss computed in COMSOL:

$$loss = \left(\int_\Omega (u - u_{ex})^2 + (v - v_{ex})^2 \, \mathrm{d}\Omega\right)^{\frac{1}{2}} + 1\text{E-}6 \cdot \sum \frac{1}{\mathrm{CFL} + 1\text{E-}8}$$

- Gradient computed in COMSOL:
  - Sensitivity analysis
- Custom output layer

Network: minimize the residuals



TUDelft

Network: minimize the residuals

- Results
  - Output is always the same for every element/input
  - During training a lot of oscillation
  - And it is very slow due to the constant interface with COMSOL

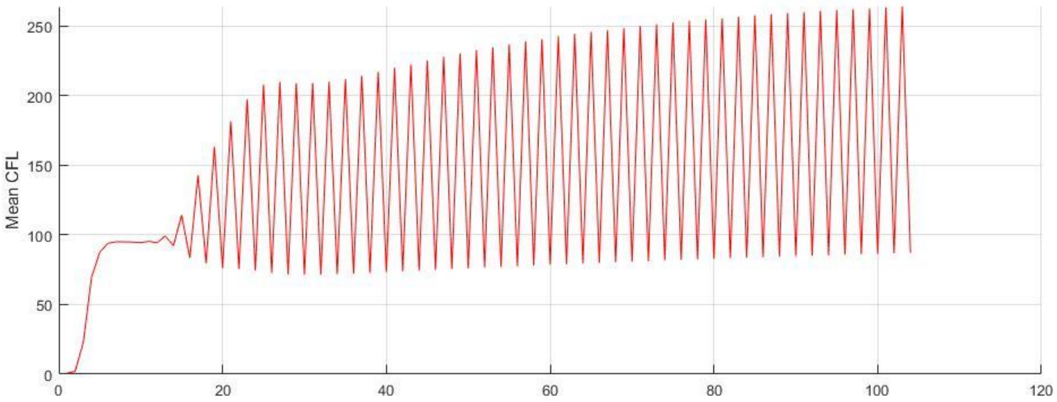
TUDelft

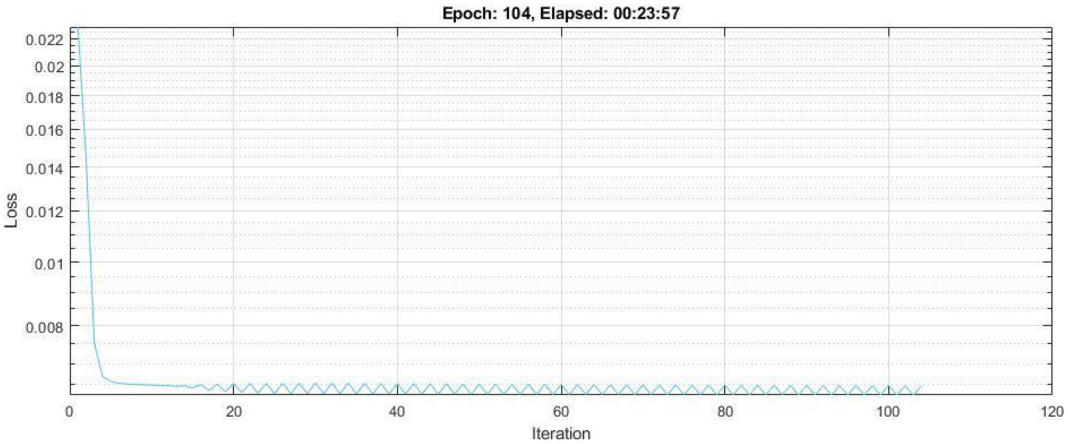
Literature

CFL and pseudo time-step

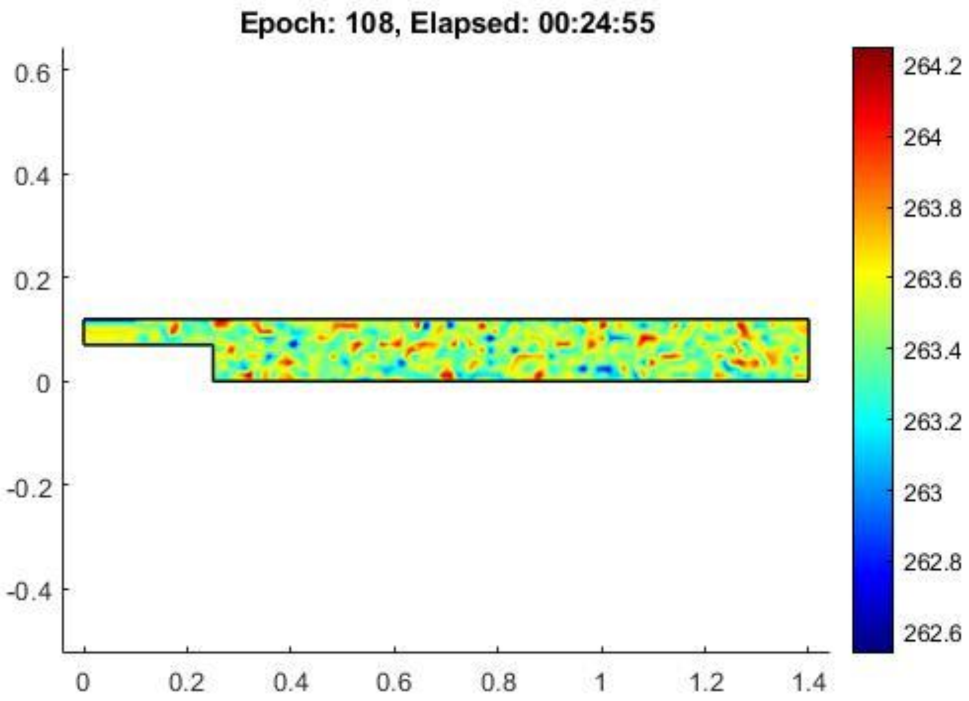
Neural networks

Further research





TU Delft


Epoch: 108, Elapsed: 00:24:55
0.6
0.4
0.2
0
-0.2
-0.4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
264.2
264
263.8
263.6
263.4
263.2
263
262.8
262.6

Further Research

- Neural Network with optimized CFL target
  - Grid search
  - Have one Newton step that is not included to plot the prediction and compute the accuracy
  - Compare with default CFL and Newton Constant

- Neural network with objective to minimize the residuals
  - Grid search
  - Still investigate how to obtain good results
    - Larger data set? Larger patch? Grid search?

TU Delft

Predicting the Optimal Solver Settings with Machine Learning in COMSOL CFD Module

25-05-2022

TU Delft

COMSOL

References

[1] Nils Margenberg et al. “A neural network multigrid solver for the Navier-Stokes equations”. In: *Journal of Computational Physics* (Jan. 2022), p. 110983

[2] Deep Ray and Jan S. Hesthaven. “Detecting troubled-cells on two-dimensional unstructured grids using a neural network”. In: *Journal of Computational PhysicsI* 397 (2019), p. 108845.

TU Delft