National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Assessment of Hybrid Computational Fluid Dynamics-Deep Learning Solvers for Unsteady Problems

Diploma Thesis

Antonis Tzanetakis
Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2024

Preface

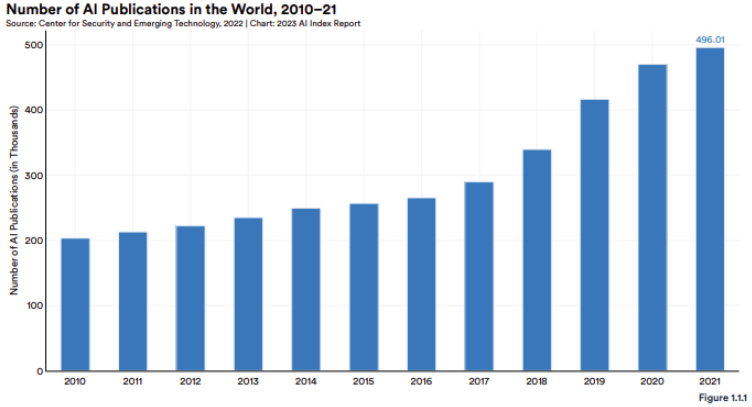
- Surge of AI research

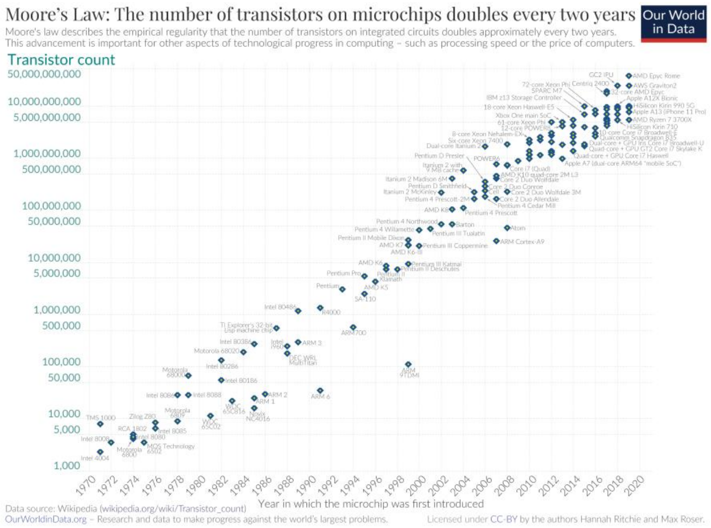
- Availability of compute and data

- AI Hardware developments (TPUs, specialized GPUs)

- CFD is rich in high quality well-structured data (but $)

=> Synergy





Contents

What this thesis is about

- Programming and evaluation of two hybrid CFD-DL methods
- Why go hybrid?
  - Generalization capabilities of numerical solvers
  - Speed of ANNs

- **Learned Interpolation (LI)** method: Production of space and time dependent coefficients by a NN for approximating spatial derivatives based on data
  - Inspired by Sinai-Bar et al. - Proceedings of the National Academy of Sciences, 2019
- **Learned Corrections (LC)** method: While solving numerically on a coarse grid, add corrections to get solutions of finer grids.
  - Inspired by Um et al. - Advances in Neural Information Processing Systems, 2020

Cases specifications

Equation

1D advection

$$\frac{\partial \rho}{\partial t} + \bar{u}\frac{\partial \rho}{\partial x} = 0$$

1D linear acoustics

$$\frac{\partial}{\partial t}\begin{bmatrix} p \\ u \end{bmatrix} + \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix} \frac{\partial}{\partial x}\begin{bmatrix} p \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$K_0 = \rho_0 {c_0}^2$$

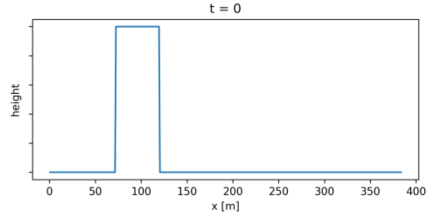
Boundary Conditions

- periodic

- periodic

Initial Conditions

- 30 square waves of different heights and widths



- 10 square waves of different heights in pressure, zero velocity

Discretization

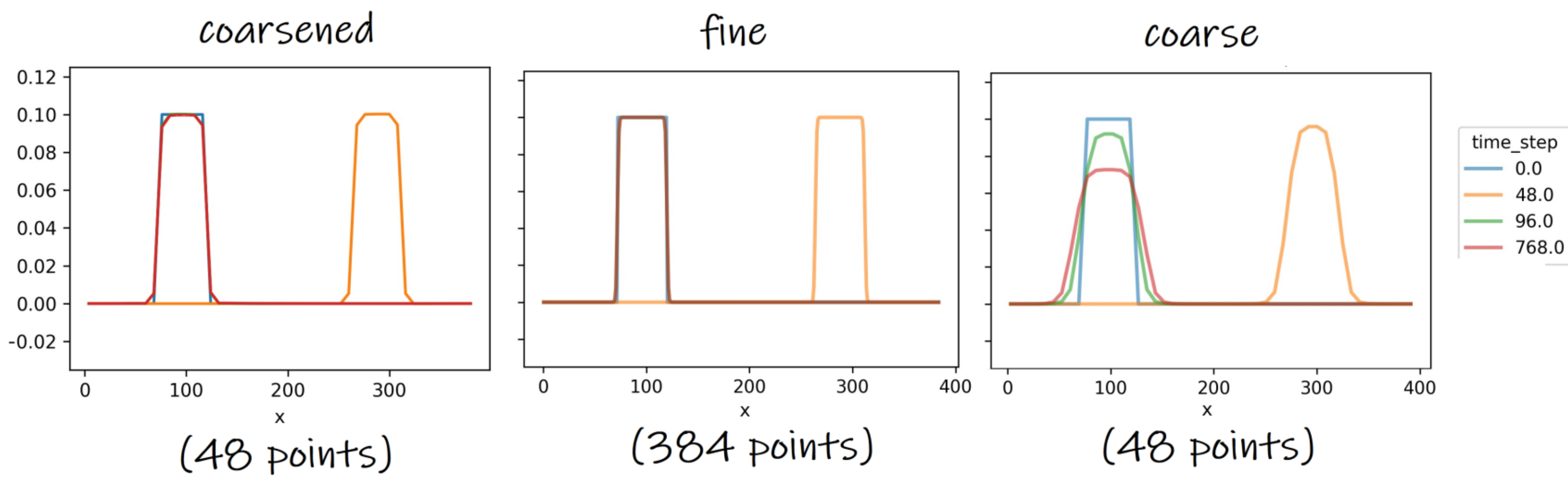
- FVM second-order with superbee

- FVM second-order with Van Leer

Hybrid Solvers (0): Main Observation

- Coarsening Hi-Fi solutions from the fine grid to the coarse grid

$\Rightarrow$ Hi-Fi solutions on the coarse grid



- Coarsened Hi-Fi solutions are not accessible by solving directly on the coarse grid
- Using NNs in the loop, we solve on the coarse grid but obtain Hi-Fi (coarsened) solutions

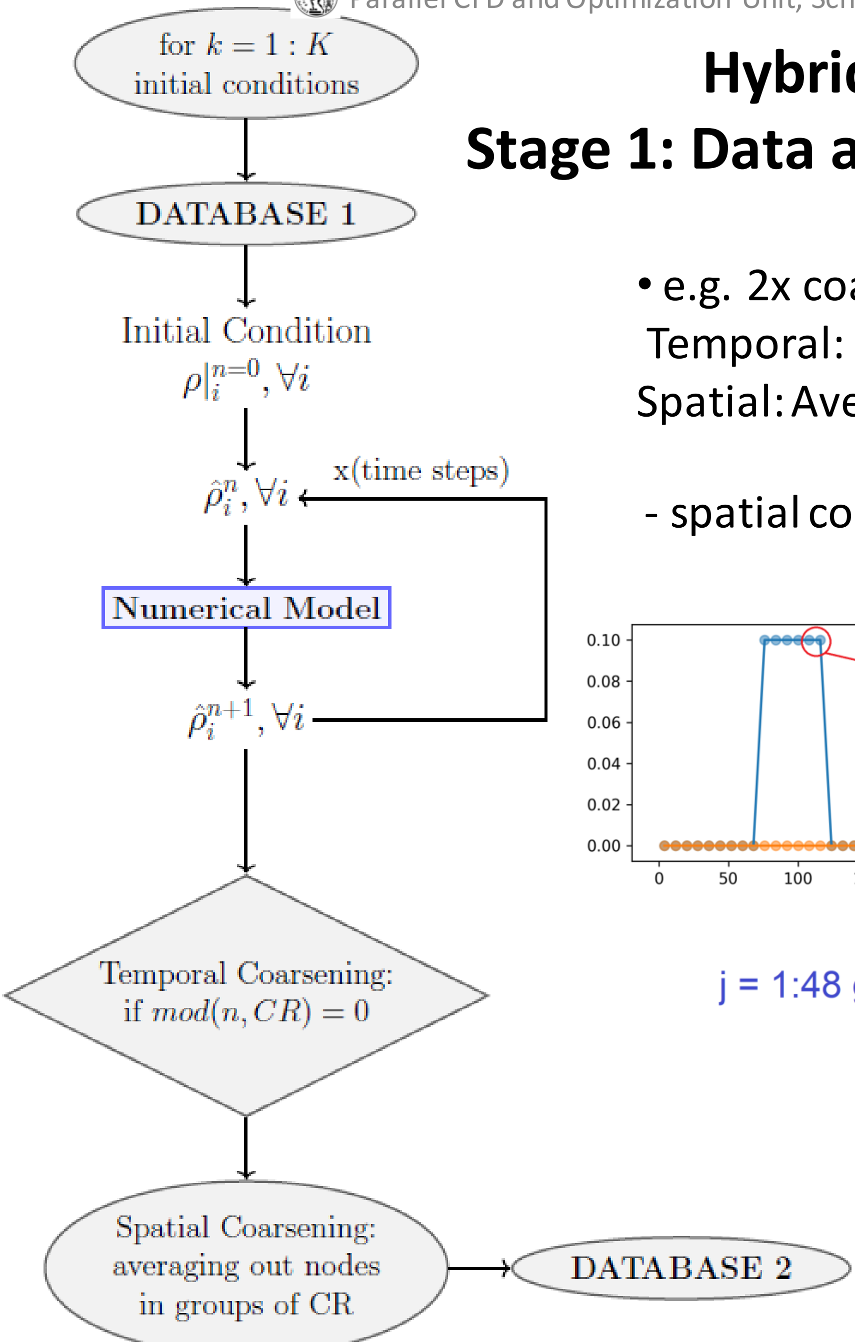
Hybrid Solvers (1): General stages

- Stage 1: Integrate the discretized PDE in time (fine grid) coarsen and save fields into storage on the coarse grid (=> training data)

- Stage 2: Train the CNN on the coarsened data. CNN will produce space and time dependent coefficients (**LI**) or corrections (**LC**)

- Stage 3: Deploy the hybrid solver to out-of-sample initial conditions to produce near-fine solutions while operating on the coarse grid.

Hybrid Solvers (2):
Stage 1: Data accumulation (LI & LC)

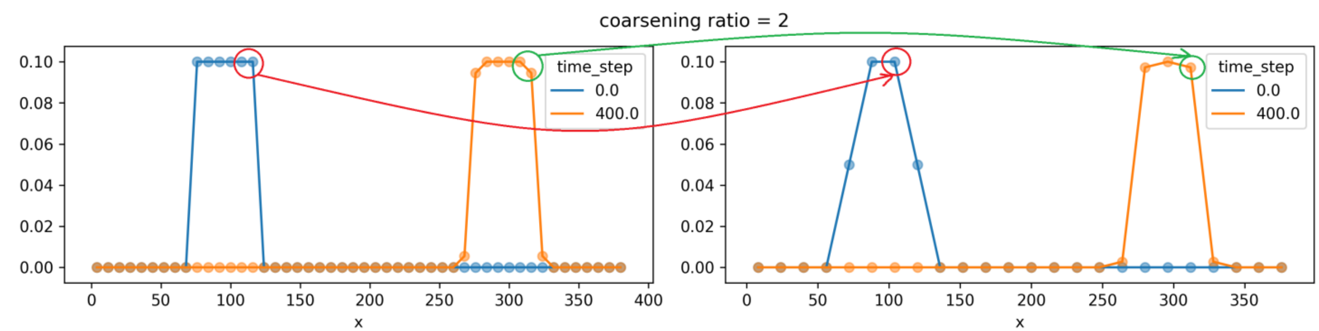


- e.g. 2x coarsening:

Temporal: Keep every 2-th snapshot: 192 → 96 time steps

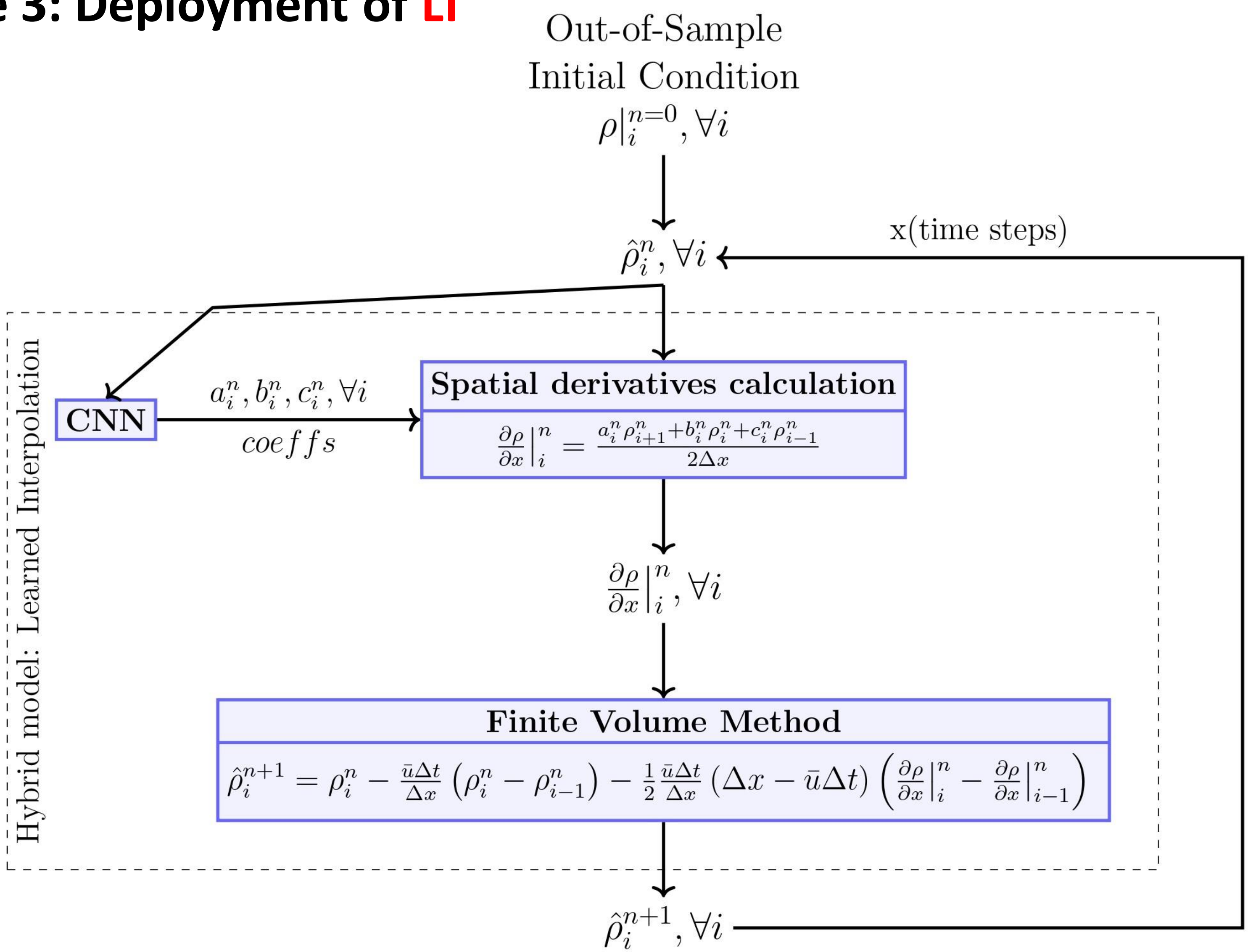
Spatial: Average every group of 2 points: 48 → 24 grid points

- spatial coarsening:

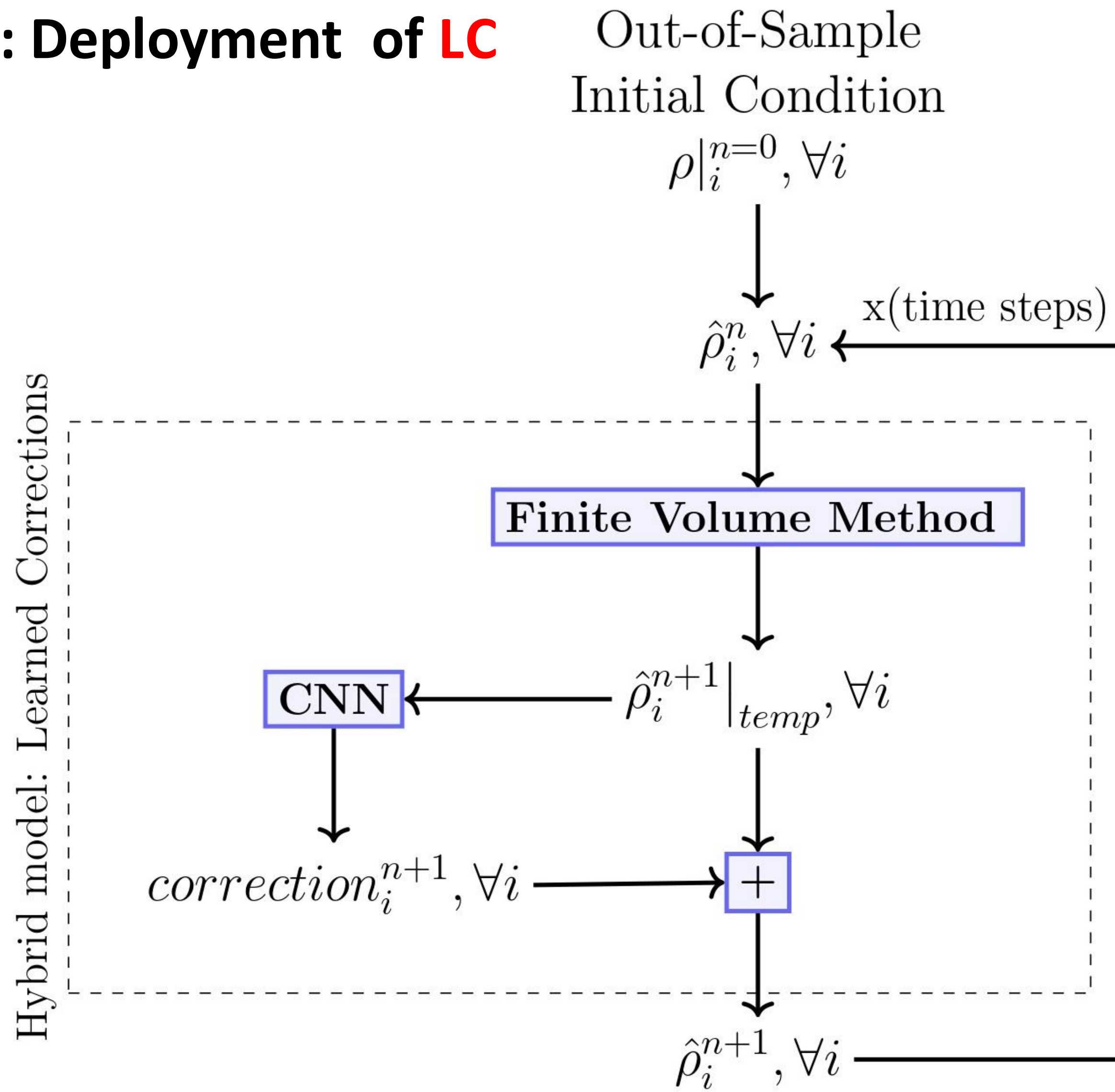


j = 1:48 grid points ⇒ $\bar{\rho}_i^n = \frac{\rho_j^n + \rho_{j+1}^n}{2}$ ⇒ i = 1:24 grid points

Hybrid Solvers (3): Stage 3: Deployment of LI

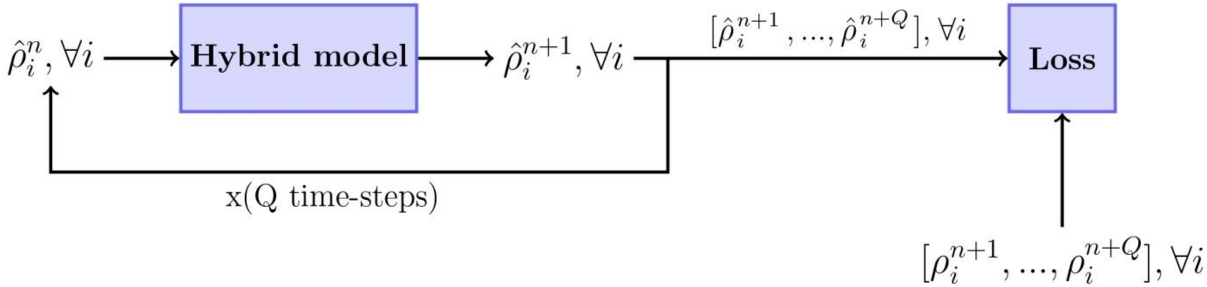


Hybrid Solvers (4):
Stage 3: Deployment of LC

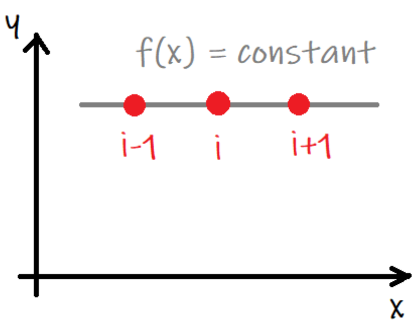


Hybrid Solvers (5): Additional elements

- Recurrent steps or “Q-steps” (**LI** & **LC**)



- Constraint of first-order accuracy (**LI** only)
  - e.g. 1st order derivative in space (3-point stencil):



$$\left.\frac{df}{dx}\right|_i = af_{i-1} + bf_i + cf_{i+1} \qquad (1)$$

$$f = constant \Rightarrow \left.\frac{df}{dx}\right|_i = 0 \qquad (2)$$

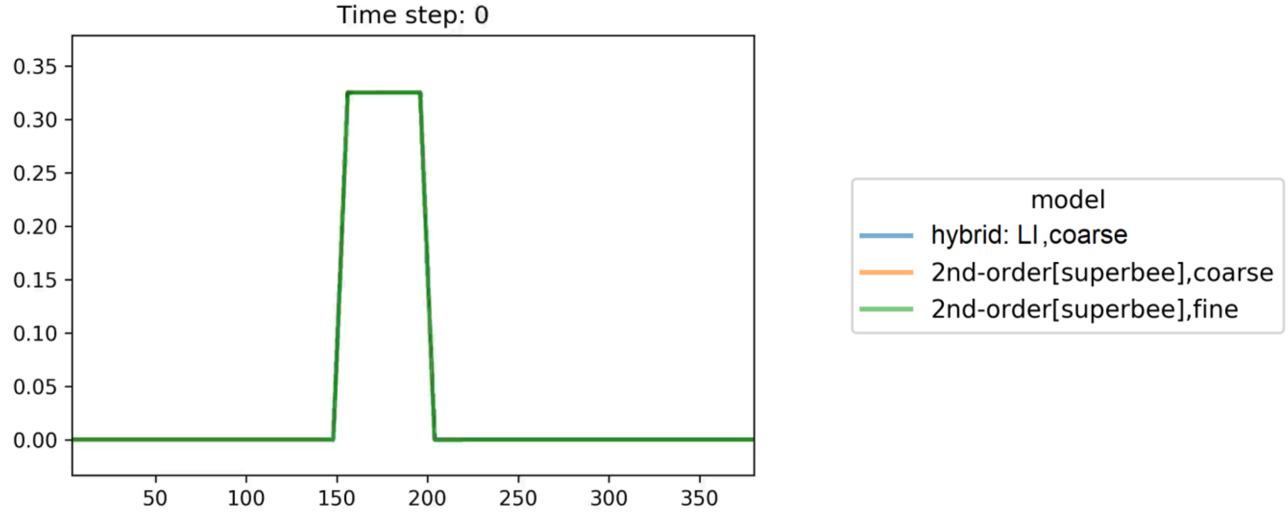
$$f_i = f_{i-1} = f_{i+1} = constant \qquad (3)$$

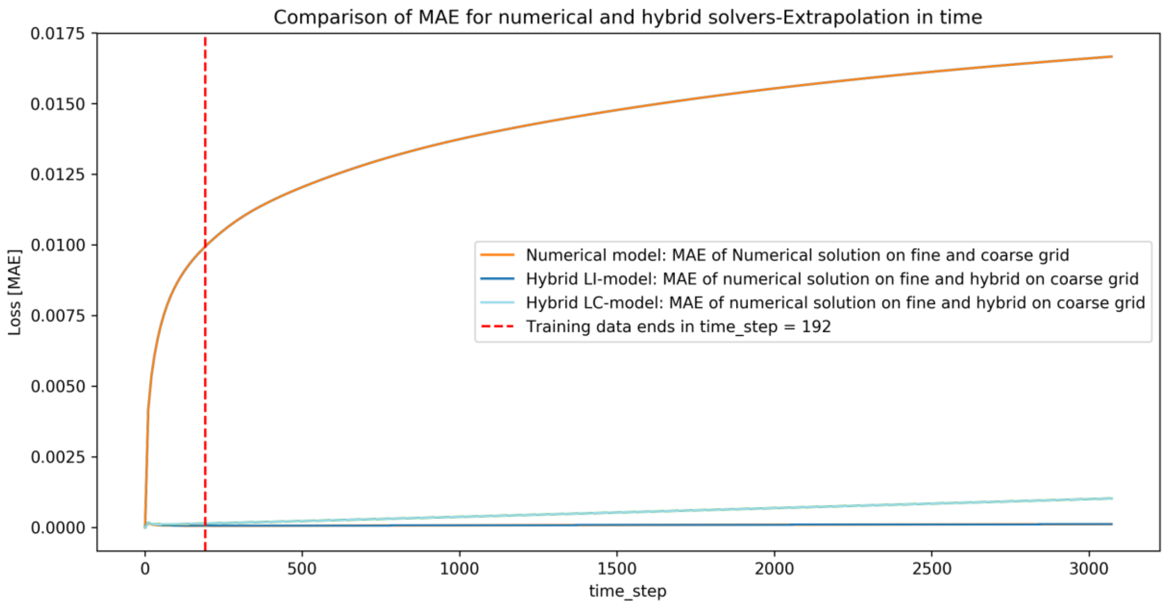
$$(1),(2),(3) \Rightarrow a+b+c=0 \qquad (4)$$

$$\left.\frac{\partial \rho}{\partial x}\right|_i^n = a_i^n \rho_{i-1}^n + b_i^n \rho_i^n + c_i^n \rho_{i+1}^n$$

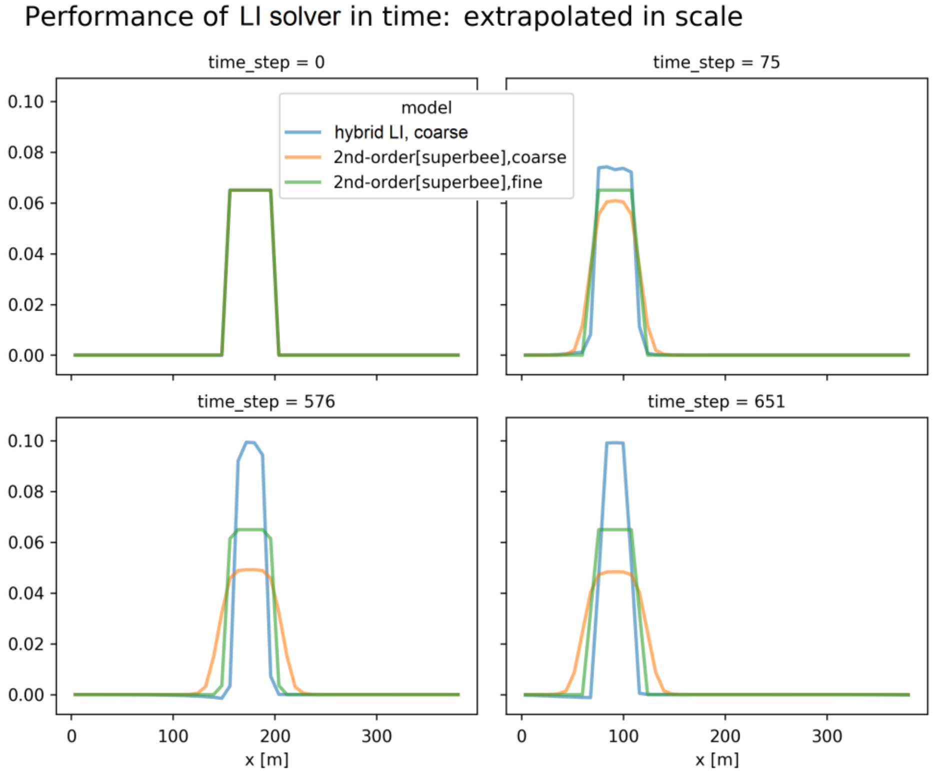
Constraint: $a_i^n + b_i^n + c_i^n = 0$

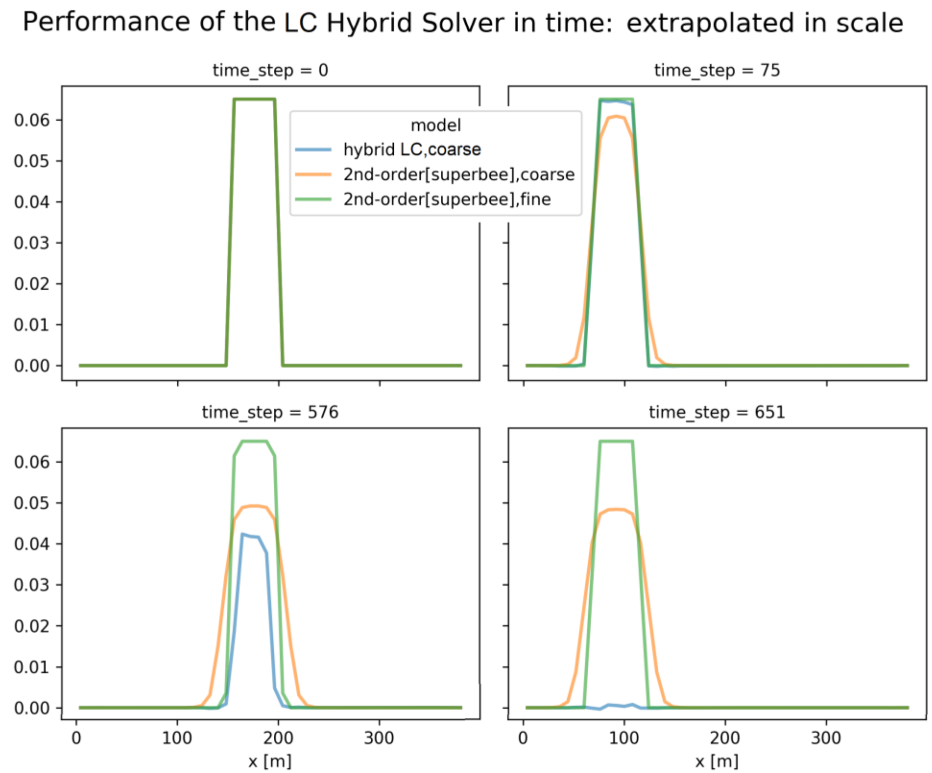
Results (1): Case 1: 1D Advection Equation: $\frac{\partial \rho}{\partial t} + \bar{u}\frac{\partial \rho}{\partial x} = 0$





Results (2): Case 1: 1D Advection Equation: Poor performance for extrapolated scales





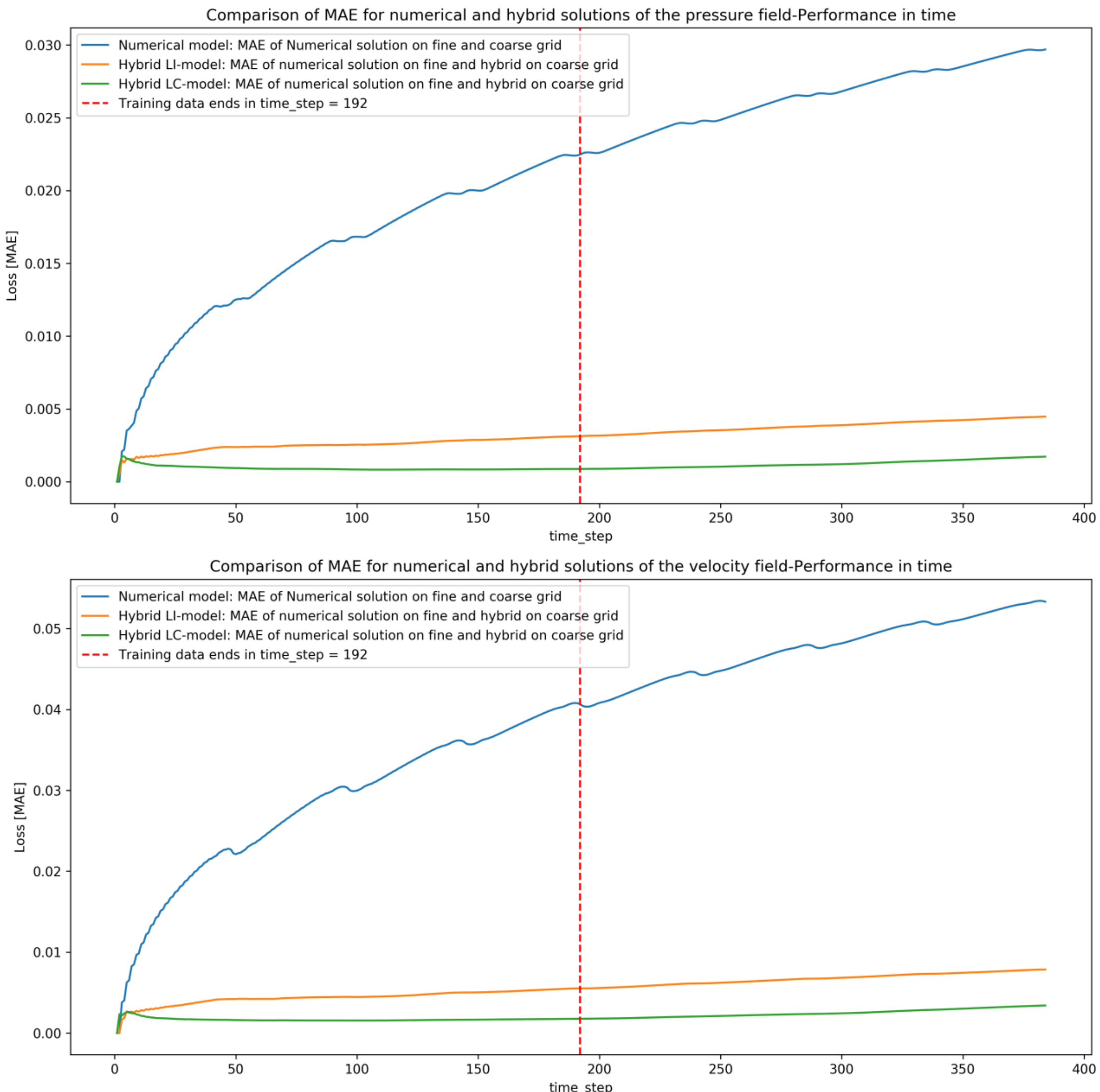
Results (5):
Case 2: 1D Linear Acoustics Equation

- Equation:

$$\frac{\partial}{\partial t}\begin{bmatrix} p \\ u \end{bmatrix} + \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix} \frac{\partial}{\partial x}\begin{bmatrix} p \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

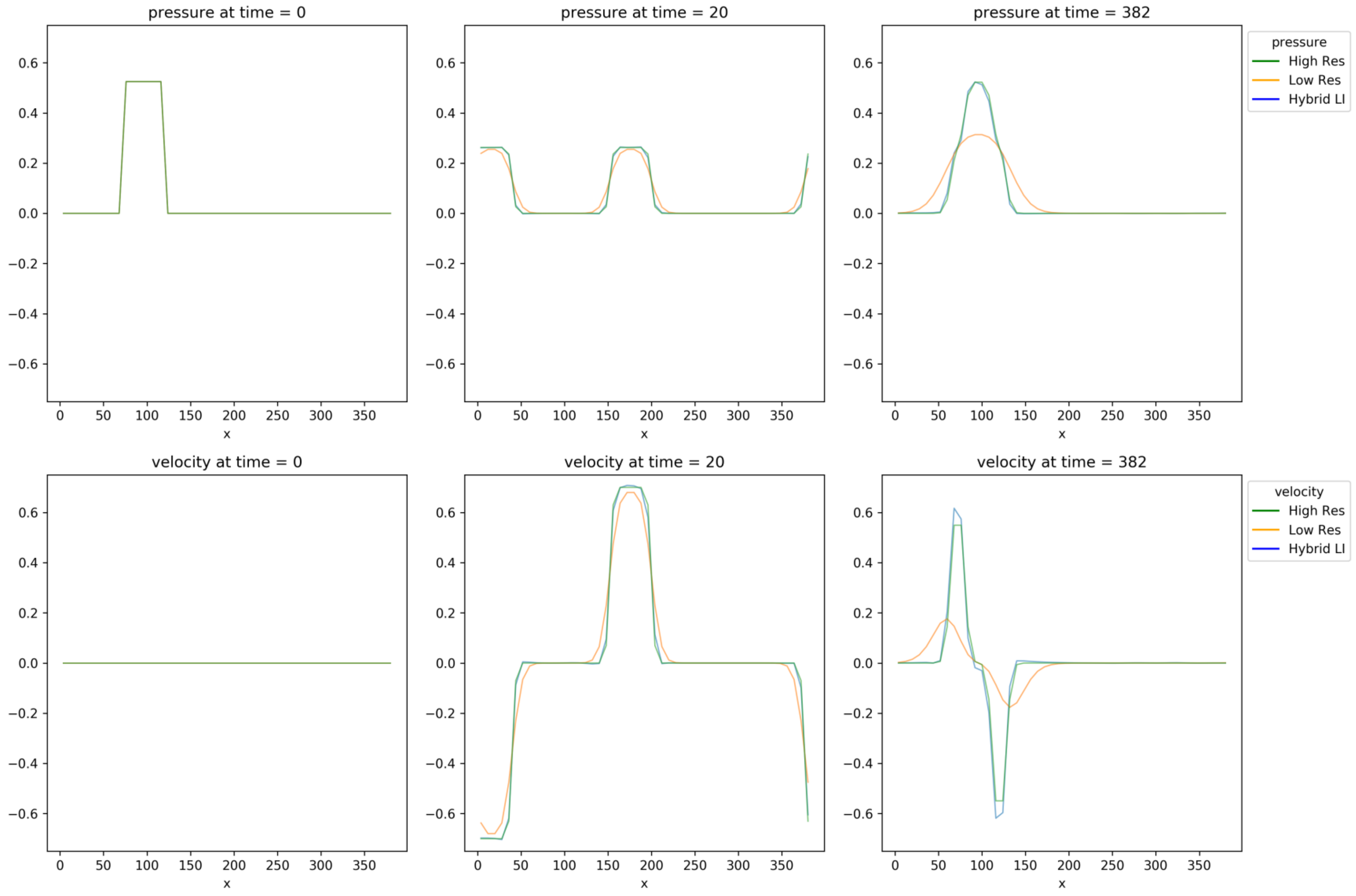
$$K_0 = \rho_0 {c_0}^2$$

- Comparison of LI (orange), LC (green) and numerical (blue) solvers on the coarse grid.



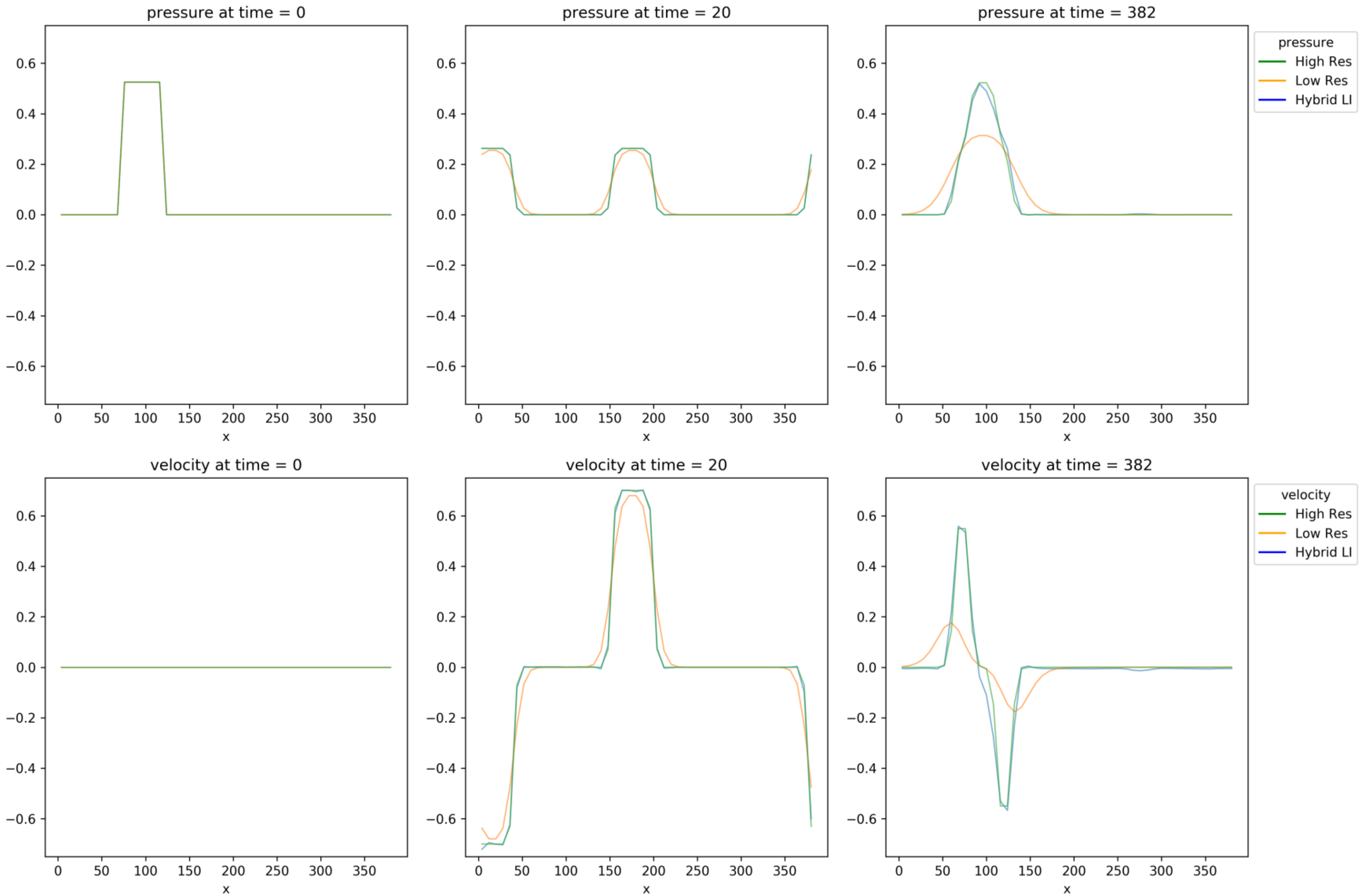
Results (4): Case 2: 1D Linear Acoustics Equation

- Comparison of **LI** on coarse, numerical solver on coarse and fine grids.

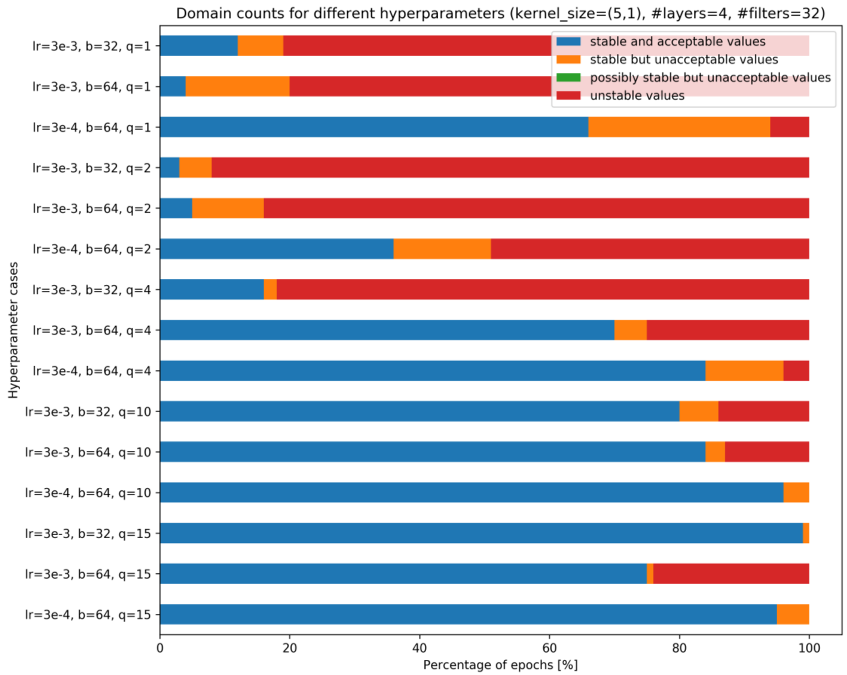


Results (5): Case 2: 1D Linear Acoustics Equation

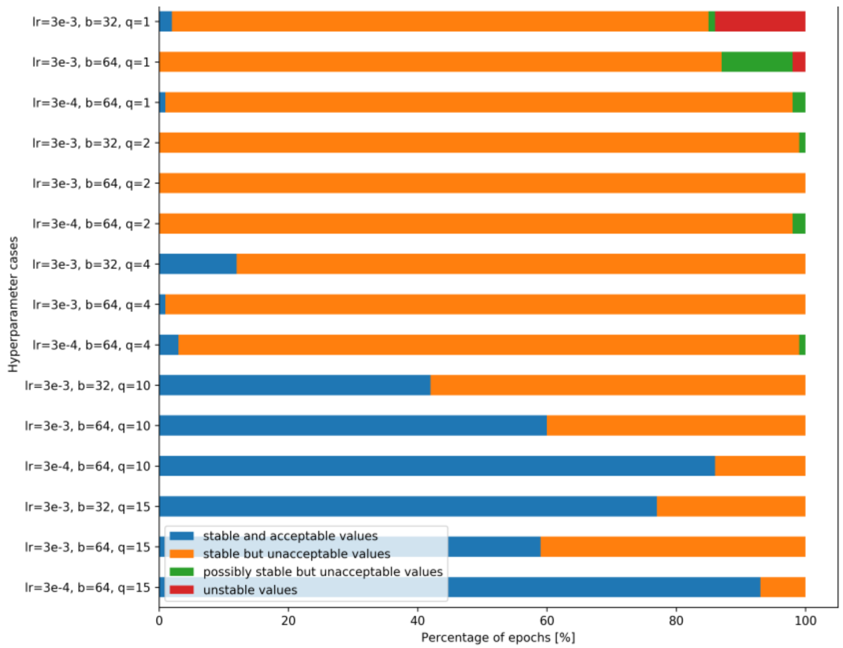
- Comparison of **LC** on coarse, numerical solver on coarse and fine grids.



Parametric study on case 2: LI & LC



LI



LC

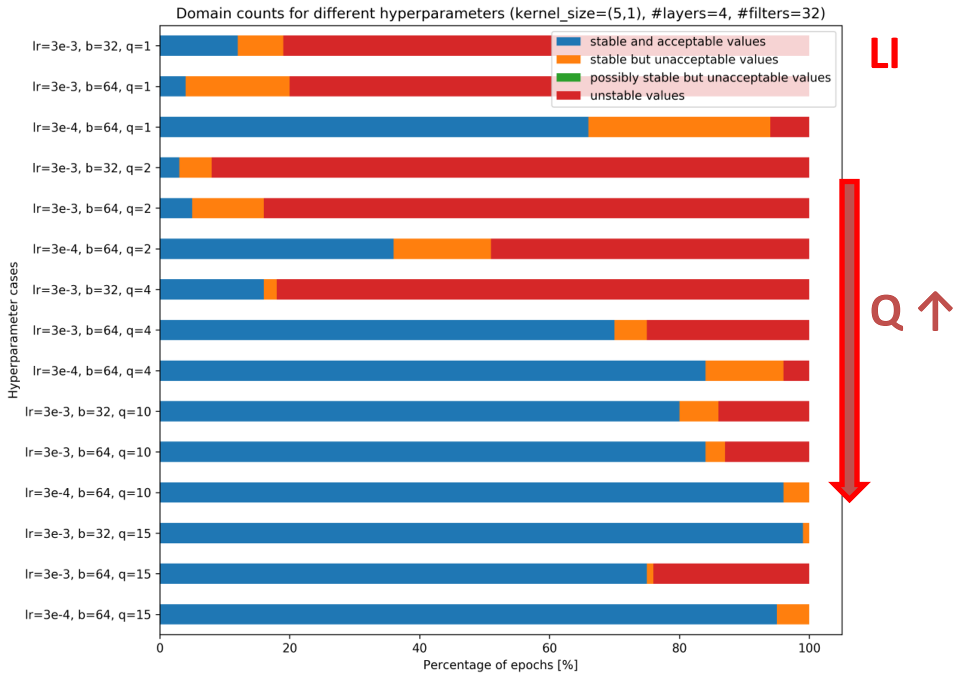
- In what percentage of training epochs are the occurring solvers stable?
- LC solvers are more stable than the LI solvers.
- LI solvers need more recurrent steps to be stable for epochs during training (or more validating runs for early stopping).

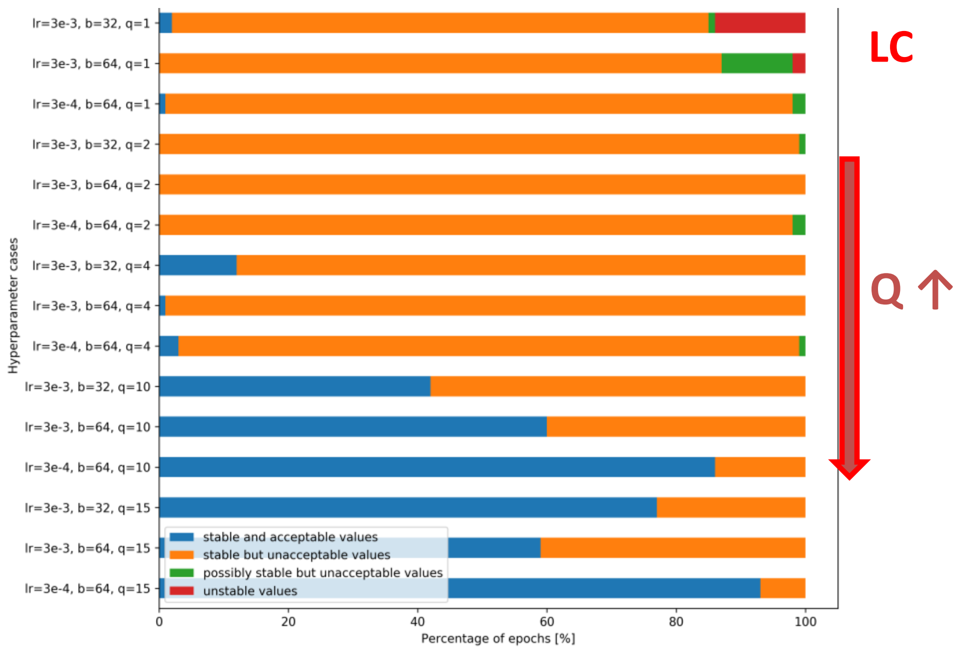
Parametric study on case 2: LI & LC

- In what percentage of training epochs are the occurring solvers stable?

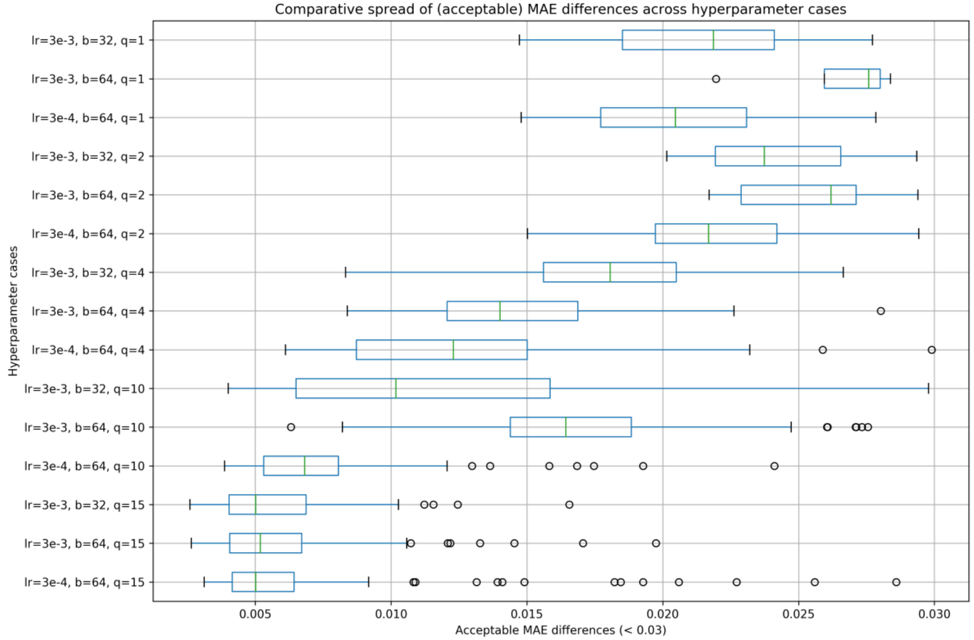
- LC solvers are more stable than the LI solvers.

- LI solvers need more recurrent steps to be stable for epochs during training (or more validating runs for early stopping).

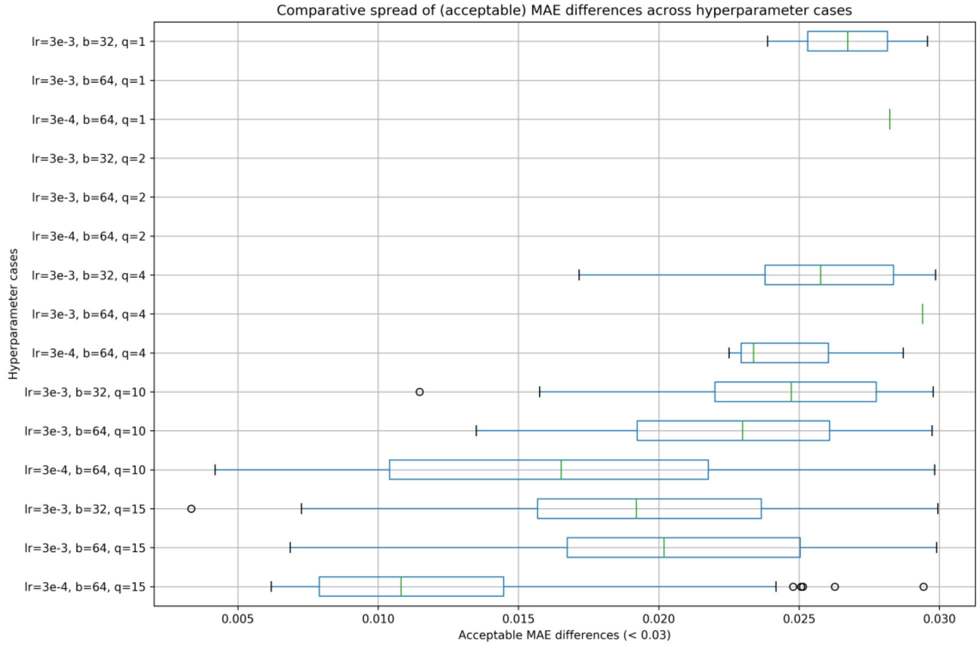




Parametric study on case 2: LI & LC



LI



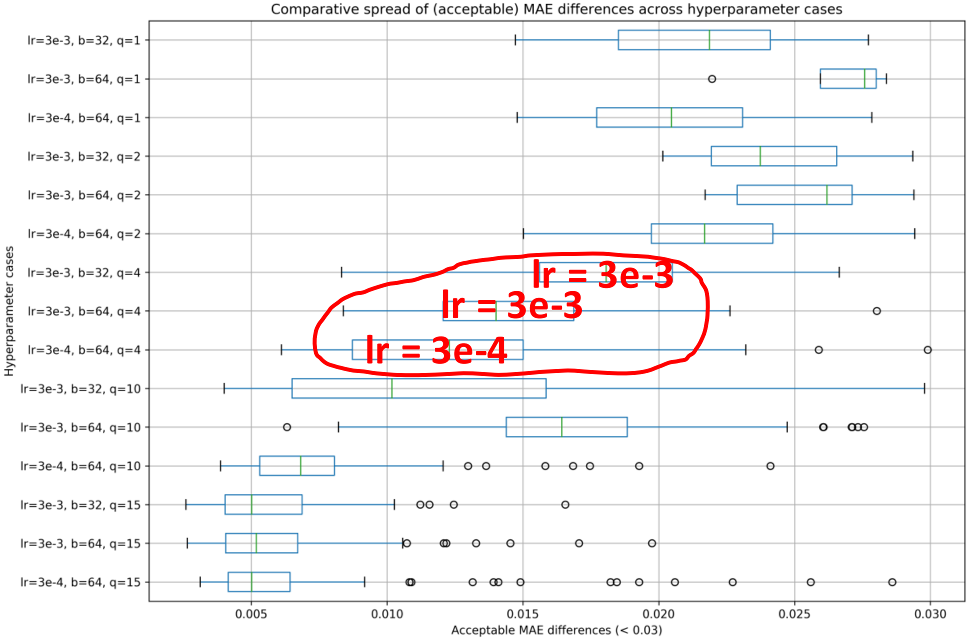
LC

- For the solvers that are stable, what are their statistics?
  - How low is the error?
  - How varied are the results?
- LC solvers need more recurrent steps and more epochs to get low errors.

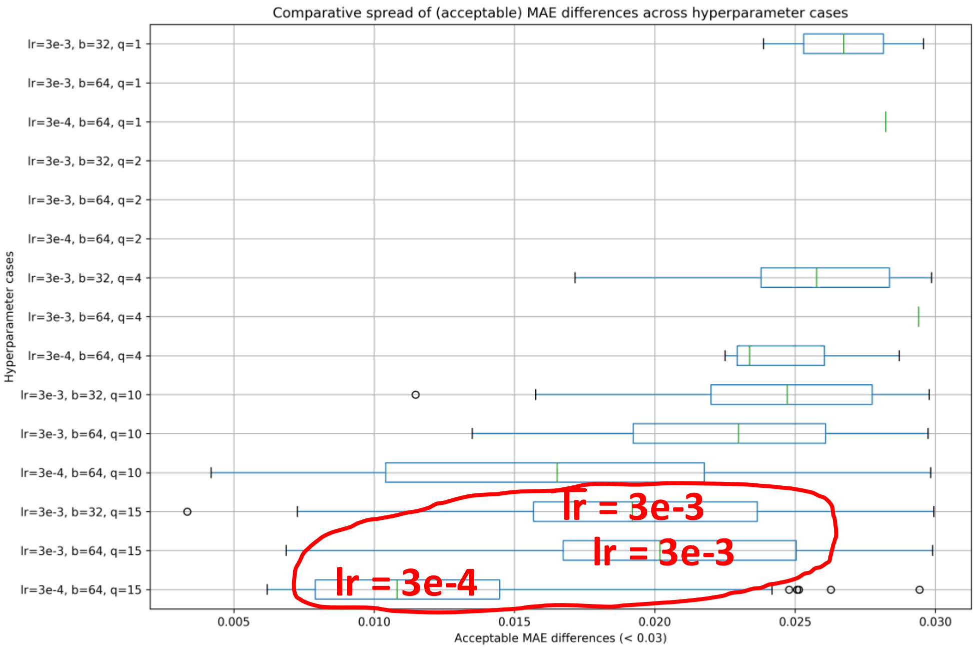
Parametric study on case 2: LI & LC

- For the solvers that are stable, what are their statistics?
  - How low is the error?
  - How varied are the results?

- LC solvers need more recurrent steps and more epochs to get low errors.



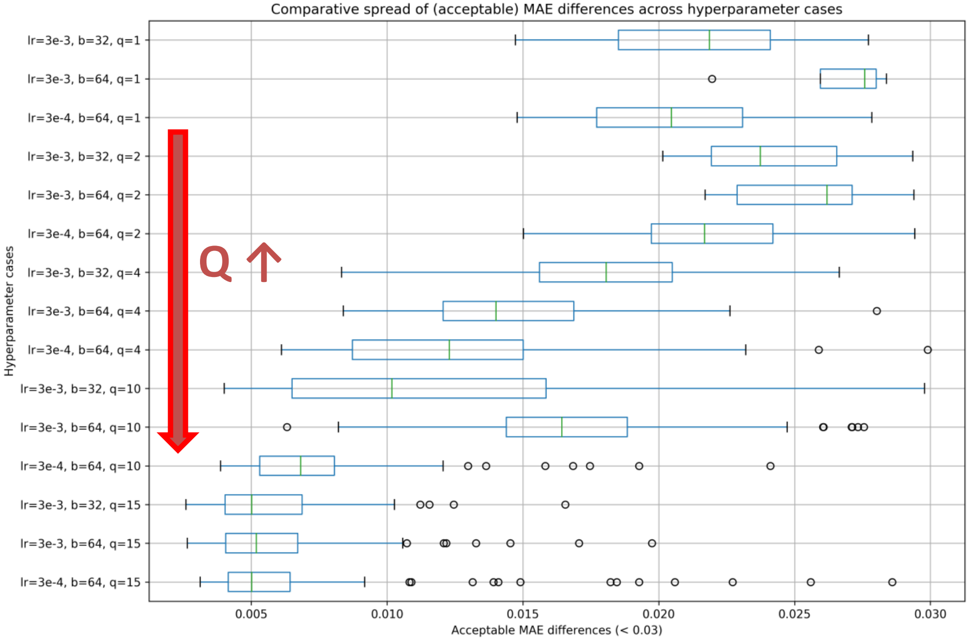
LI



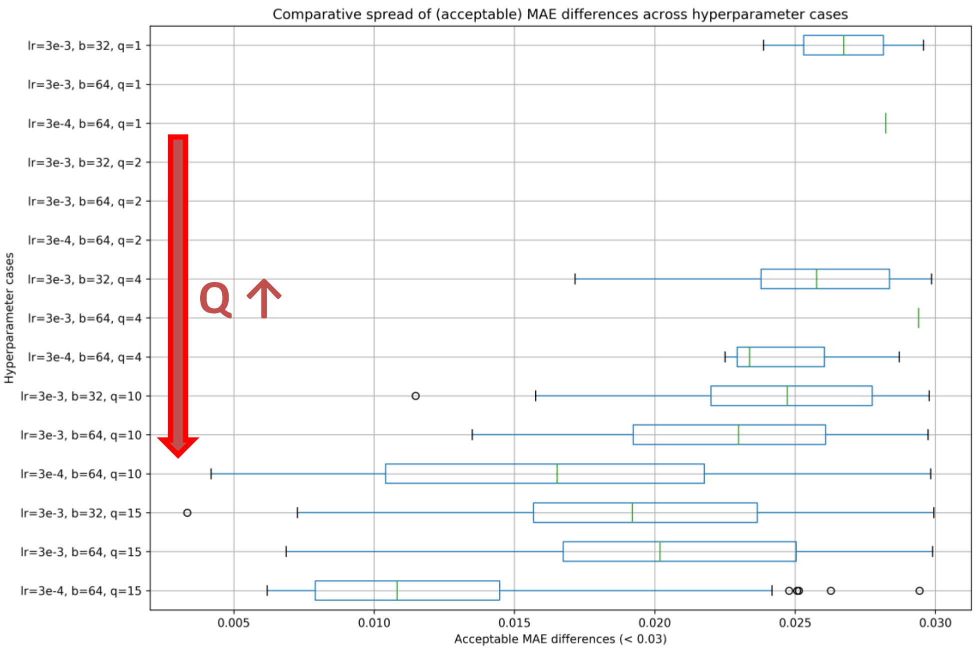
LC

Parametric study on case 2: LI & LC

- For the solvers that are stable, what are their statistics?
  - How low is the error?
  - How varied are the results?
- LC solvers need more recurrent steps and more epochs to get low errors.



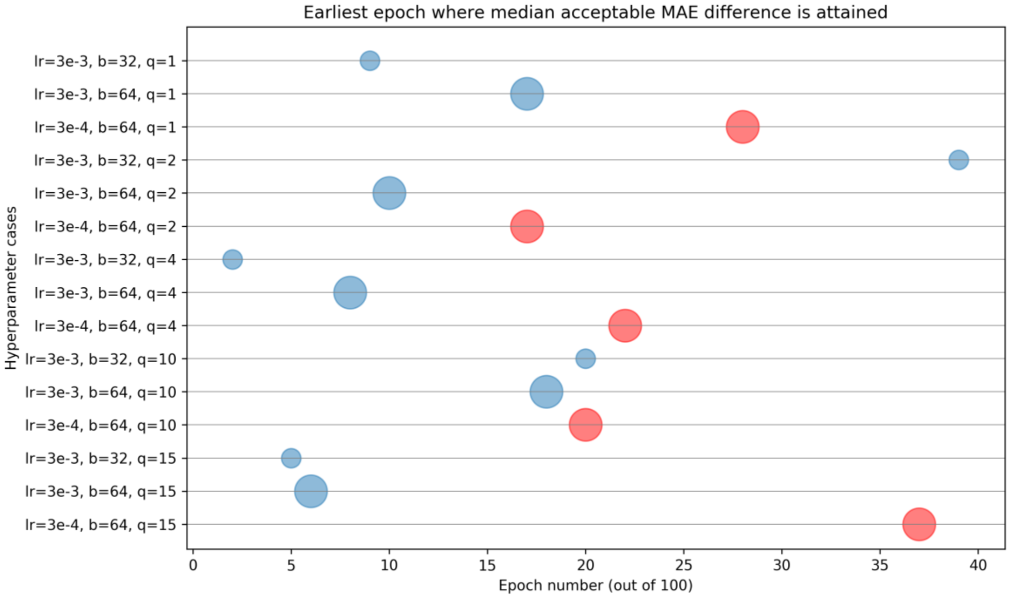
LI



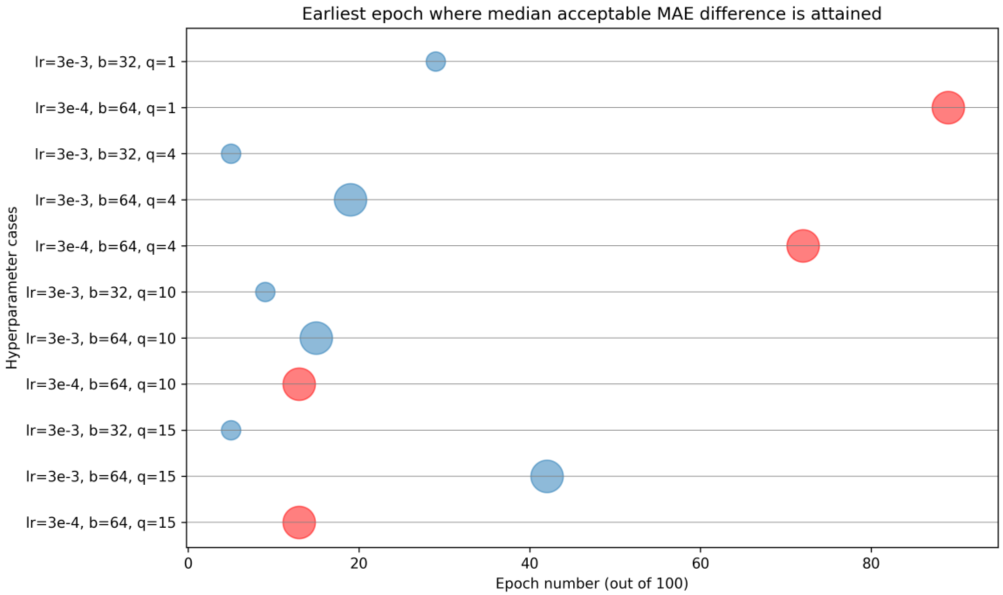
LC

Parametric study on case 2: LI & LC

- How early do they achieve the median of their capability?
  - No correlation with batch size.
  - Smaller learning rates converge slower but at better results for LI, unclear for LC.



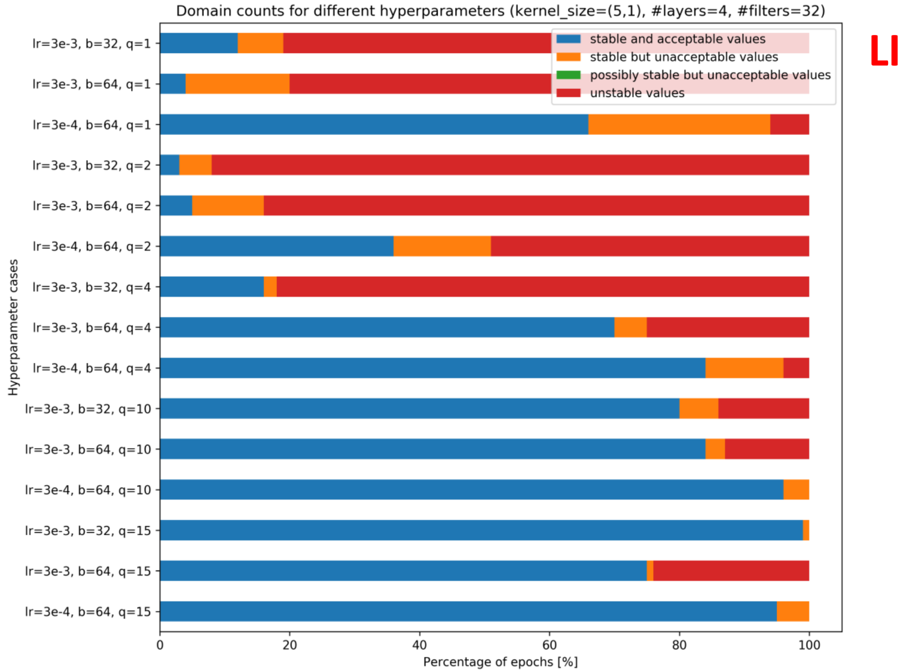
LI

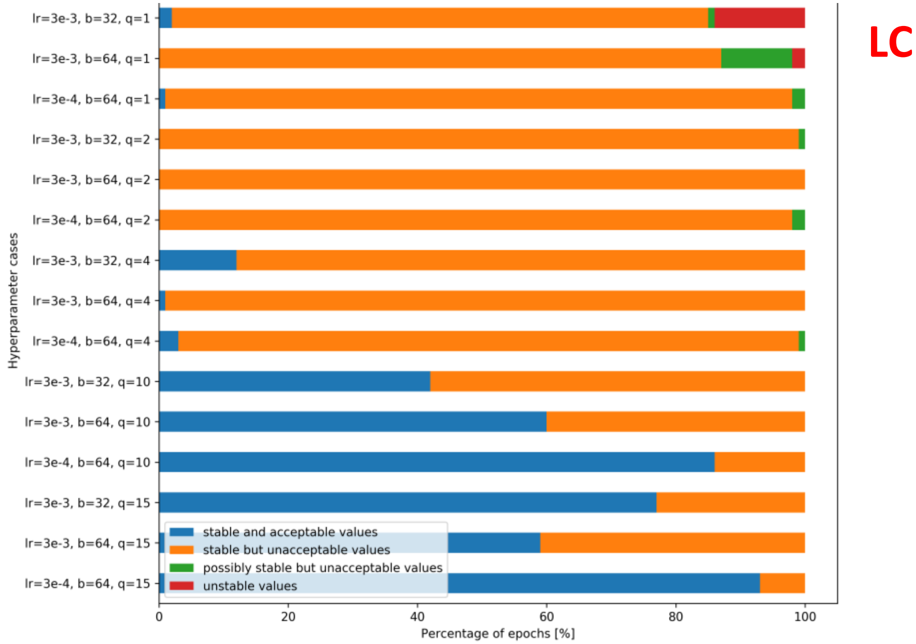


LC

Parametric study on case 2: LI & LC

- In what percentage of training epochs are the occurring solvers stable?
- LC solvers are more stable than the LI solvers.
- LI solvers need more recurrent steps to be stable for epochs during training (or more validating runs for early stopping).

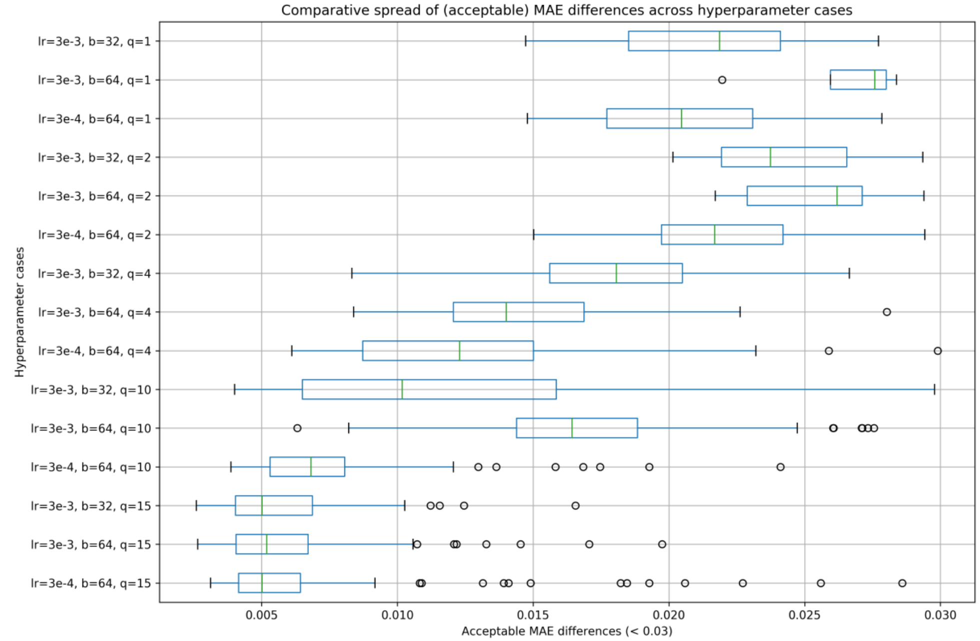




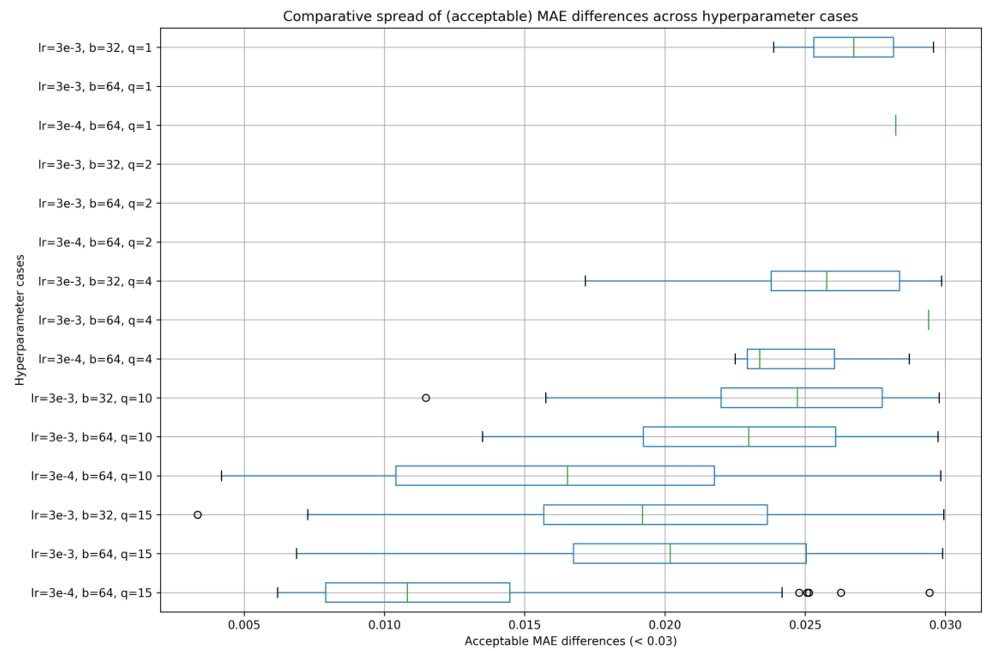
Parametric study on case 2: LI & LC

- For the solvers that are stable, what are their statistics?
  - How low is the error?
  - How varied are the results?

- LC solvers need more recurrent steps and more epochs to get low errors.



LI



LC

Possible reasons for why these methods work (1): Learned Interpolation

→ Why predict coefficients?

- Minimal modification of the numerical solver
- Locality is preserved

$$\left.\frac{\partial \rho}{\partial x}\right|_i^n = NN(\rho_j^n, \forall j)$$

versus

$$\left.\frac{\partial \rho}{\partial x}\right|_i^n = NN(\rho_j^n, \forall j) * \rho_k^n, \quad \text{where } k = \{i-1, i, i+1\}$$

- Working coefficients have small variance.

Possible reasons for why these methods work (2): Learned Corrections

→ Prediction-Correction scheme is a very successful idea

- The numerical parts handle most of the dynamics
- Residual connections' advantages in training NNs
  - No vanishing/exploding gradient problem
  - More optimizable parameters => better performance

→ Why online and not offline corrections?

- Small terms (in less scales) to learn
- On-line corrections => the error does not get

a chance to grow

Summary-Conclusions

- Programming and evaluation of two hybrid methods => technical know-how
- Significant potential acceleration of unsteady problems' solutions (20-80x speedup for 2D-3D flows)
- Challenges for hybrid solvers.
  - Extrapolation
  - Scaling up
  - Integration with existing CFD codebases