KU LEUVEN

Kinetic-Diffusion Monte-Carlo

based on work with G. Samaey, W. Dekeyser, T. Baelmans, B. Mortier, E. Loevbak, O. Lappi, V. Maes

Thijs Steel

KU Leuven

August 22nd 2025

1 Outline

1 KDMC: recap

- Fully Monte-Carlo fluid-kinetic hybrid method.
- Alternate between normal kinetic increment and brownian motion.
- Automatic determination of how much weight to give to kinetic and brownian increment based on collisionality.

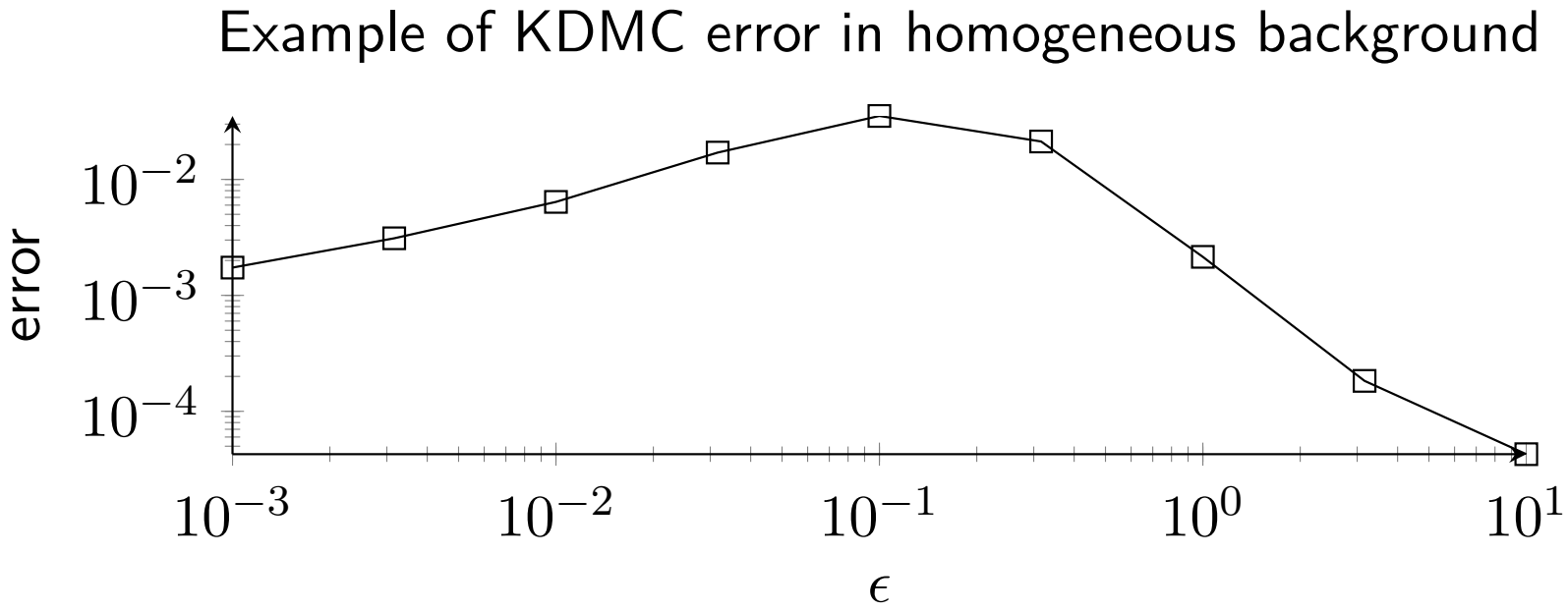
1 KDMC: pseudocode

Simulation of one timestep Δt

1. Sample time till background collision τ_c
2. Sample new velocity V^{n+1}
3. Determine coefficients for brownian motion μ, σ
4. $X^{n+1} = X^n + \tau_c V^n + (\Delta t - \tau_c)\mu + \sqrt{\Delta t - \tau_c}\sigma n$

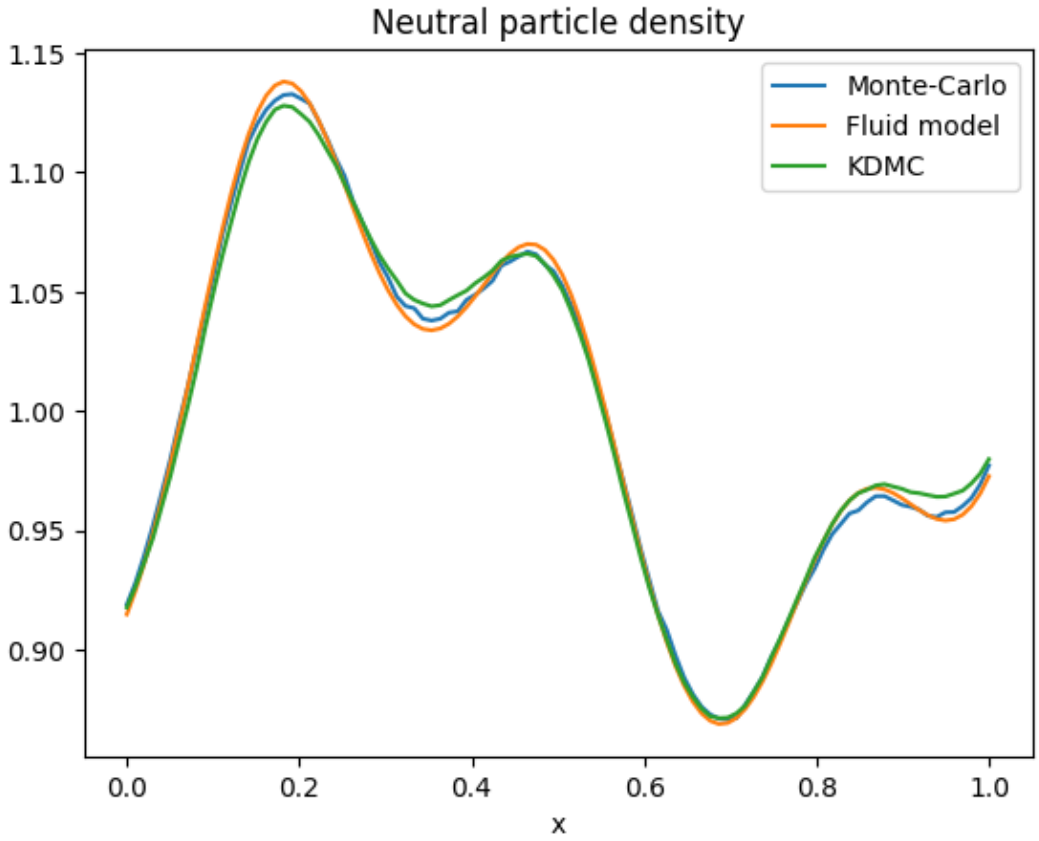
Steps 1 and 2 are kinetic, steps 3 and 4 are diffusive.

1 Errors in KD



- Error is smallest in low collisional regime (fully kinetic) and high collisional regime (fluid approximation is accurate)
- low and high collisional is relative to the chosen timestep $\rightarrow$ error can be controlled.
- (Convergence is slower than normal here because of MC noise)

1 Example of KDMC in non-homogeneous background



1 Extra features

- Fluid estimator: extract QOI using a fluid model with the KDMC positions as initial condition
- Multilevel KDMC: use a hierarchy of timesteps to significantly improve efficiency. Potentially achieving results consistent with fully kinetic code at a reduced cost.

2 Outline

2 Boundary conditions

Original KDMC:

- if a particle crosses a boundary, switch to fully kinetic for that time step.
- Easy to implement
- expensive
- not (always) accurate

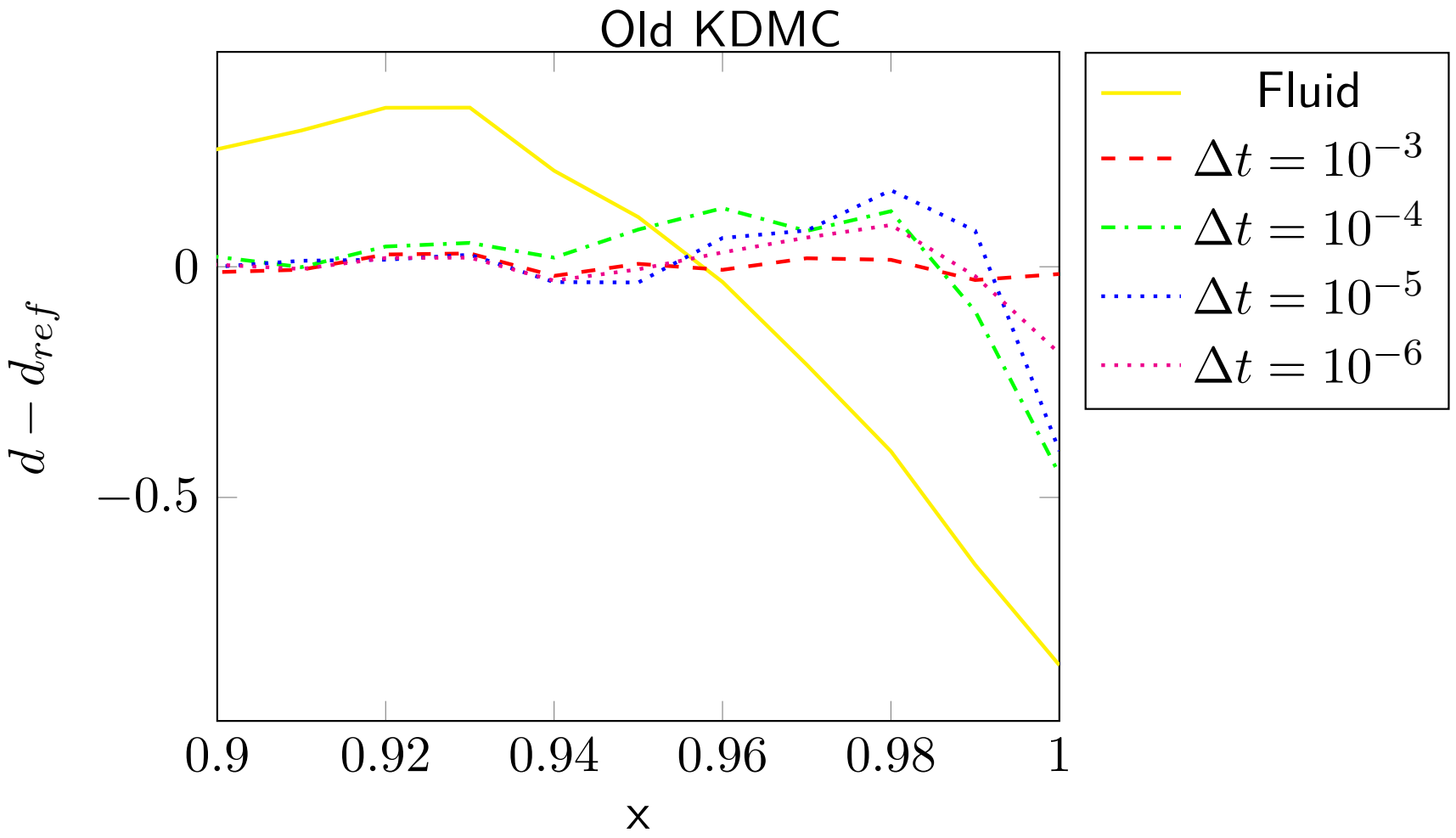
New idea:

- Take the boundary into account when solving the diffusive SDE
- Requires separate fluid boundary implementation
- cheap
- more accurate

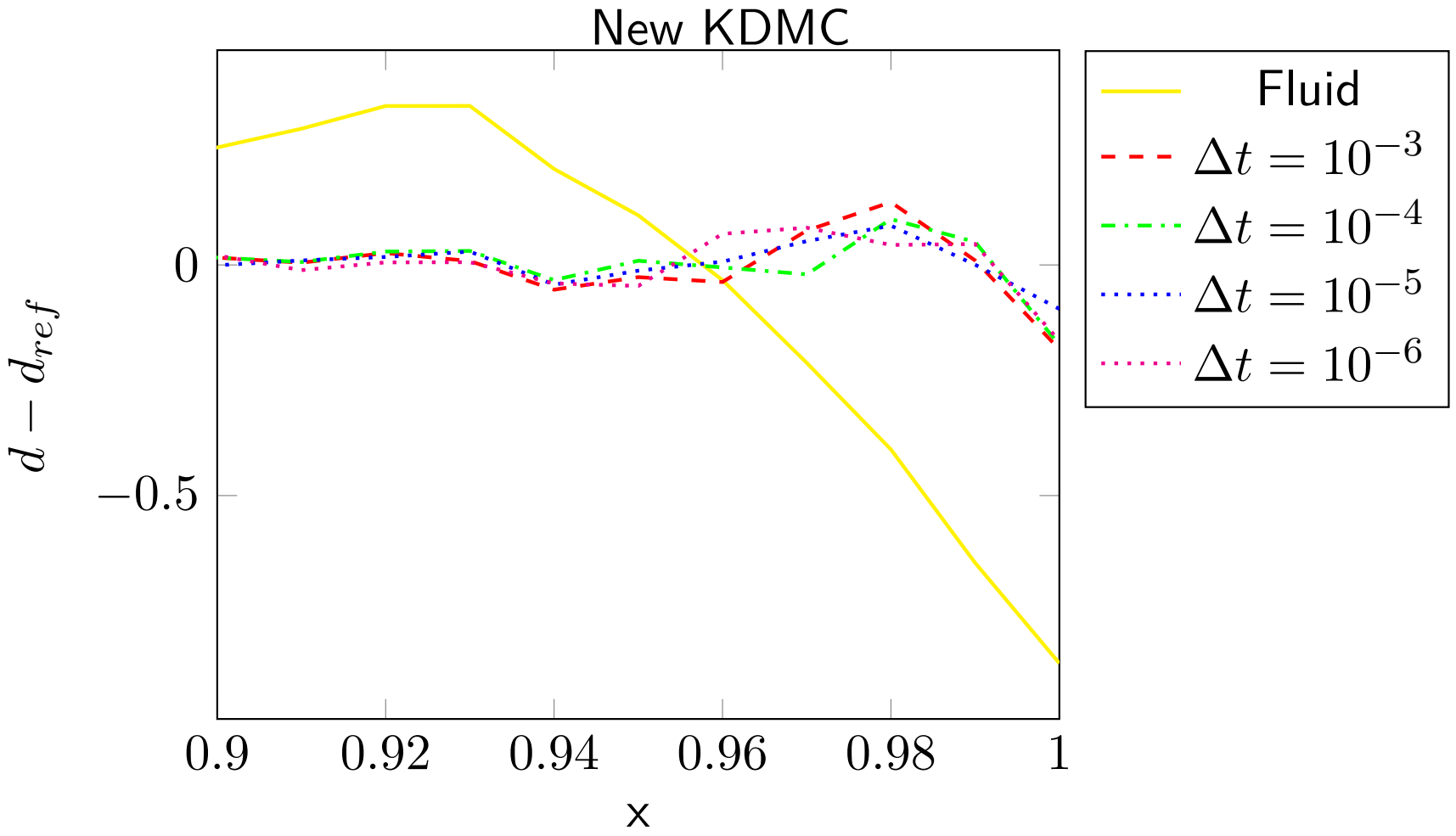
2 Boundary condition test 1

- Reflecting boundary
- Rate: $10^7 s^{-1}$
- BGK stddev: 10^7
- BGK mean: 0

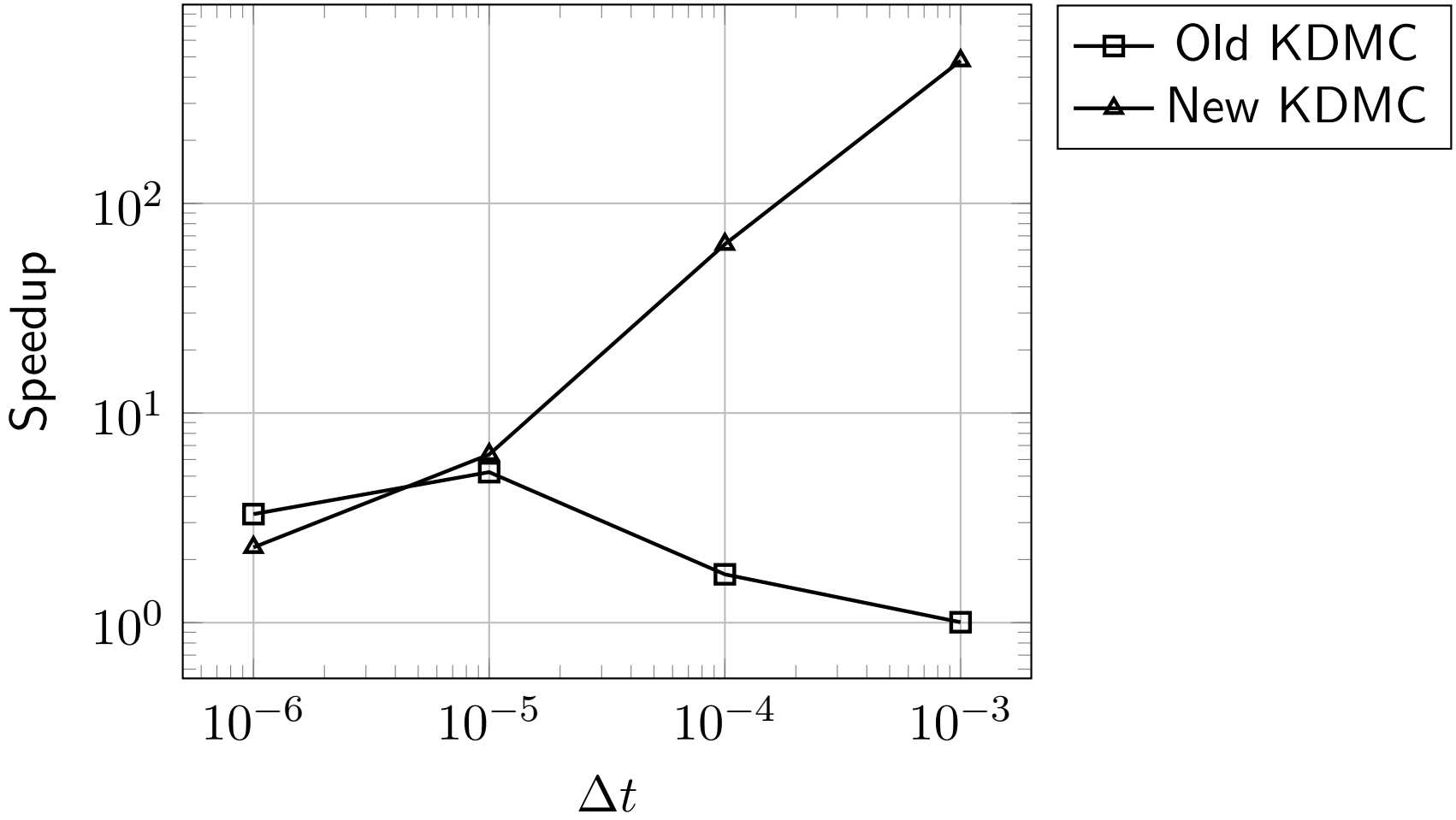
2 Boundary condition test 1



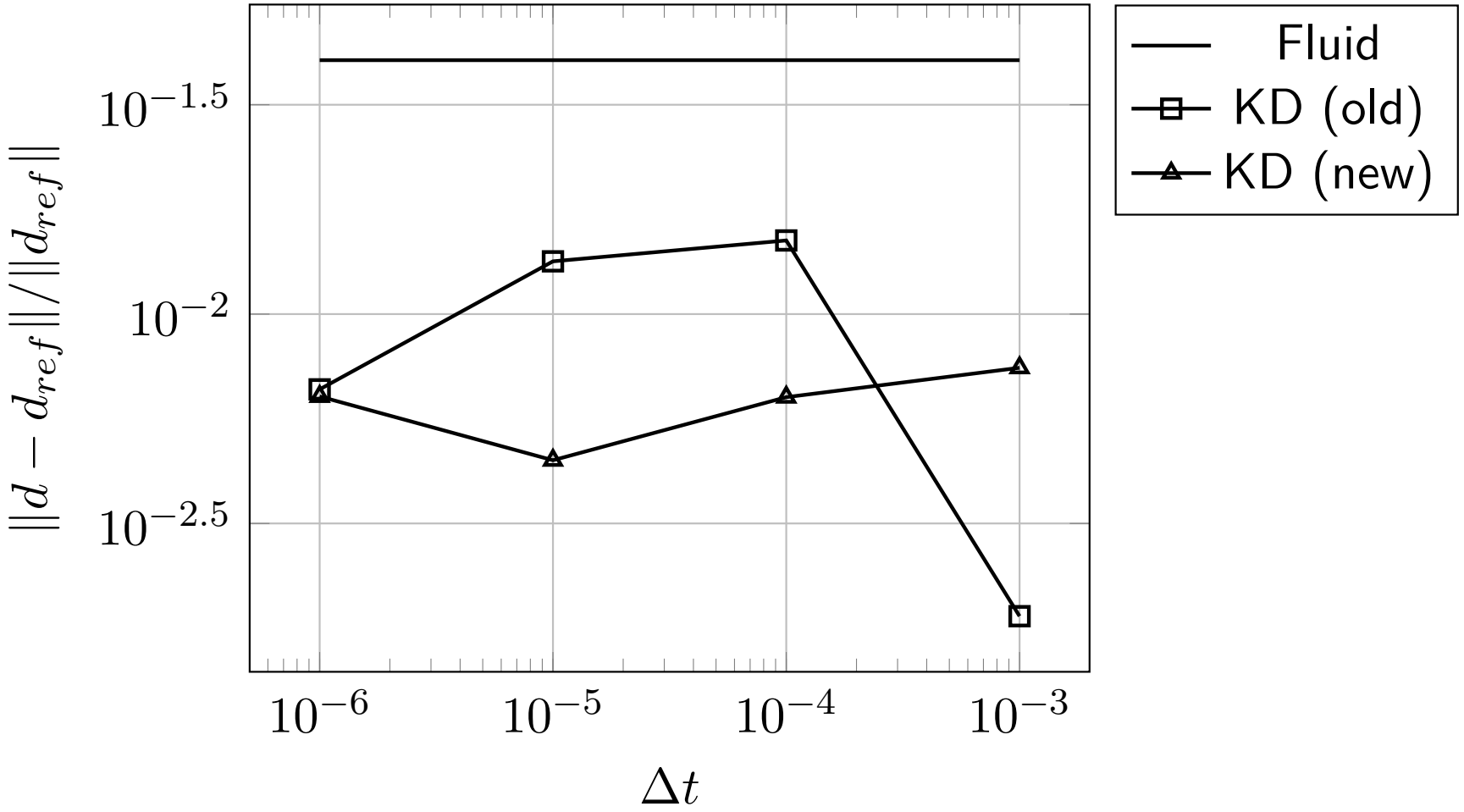
2 Boundary condition test 1



2 Boundary condition test 1



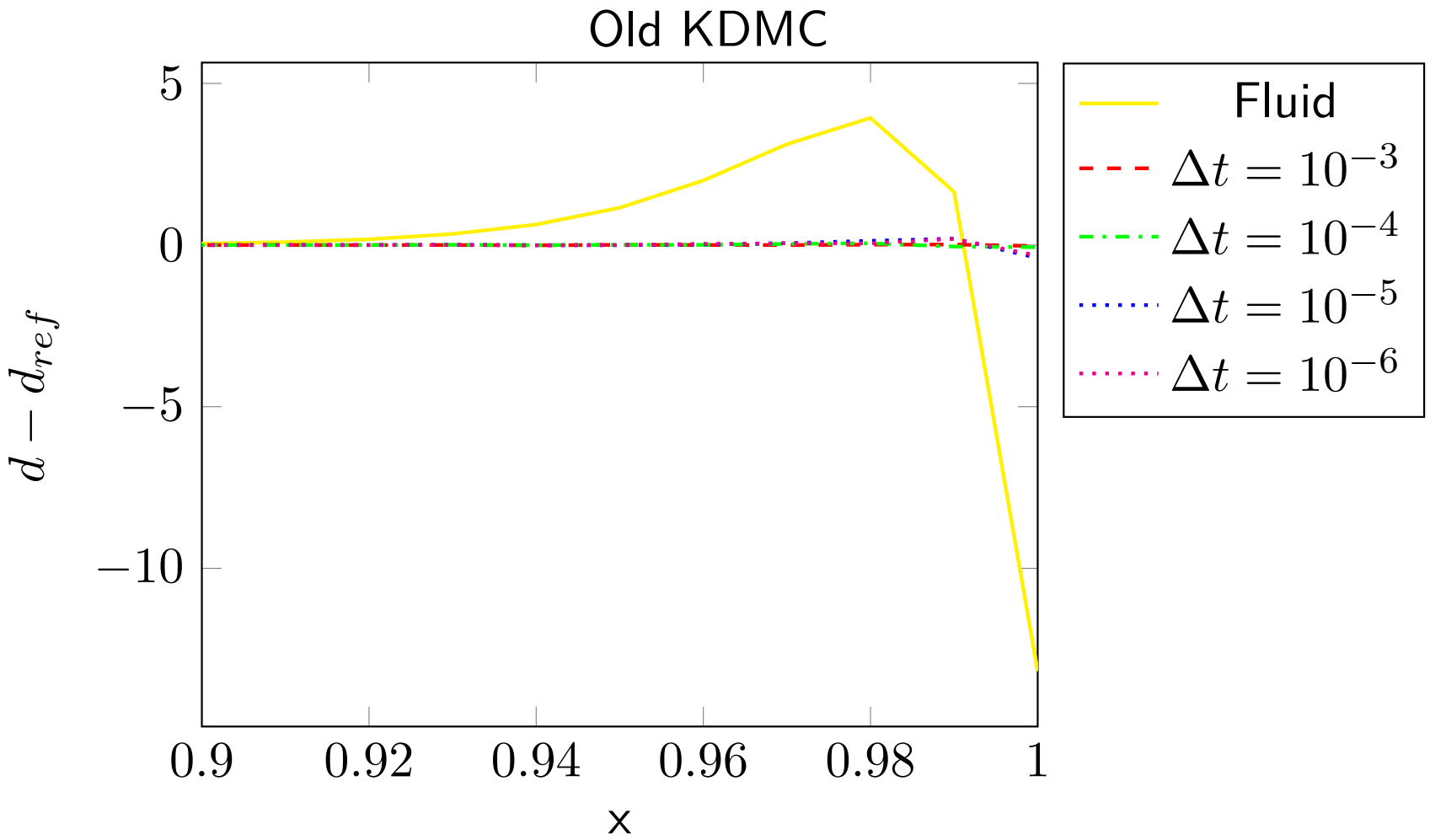
2 Boundary condition test 1



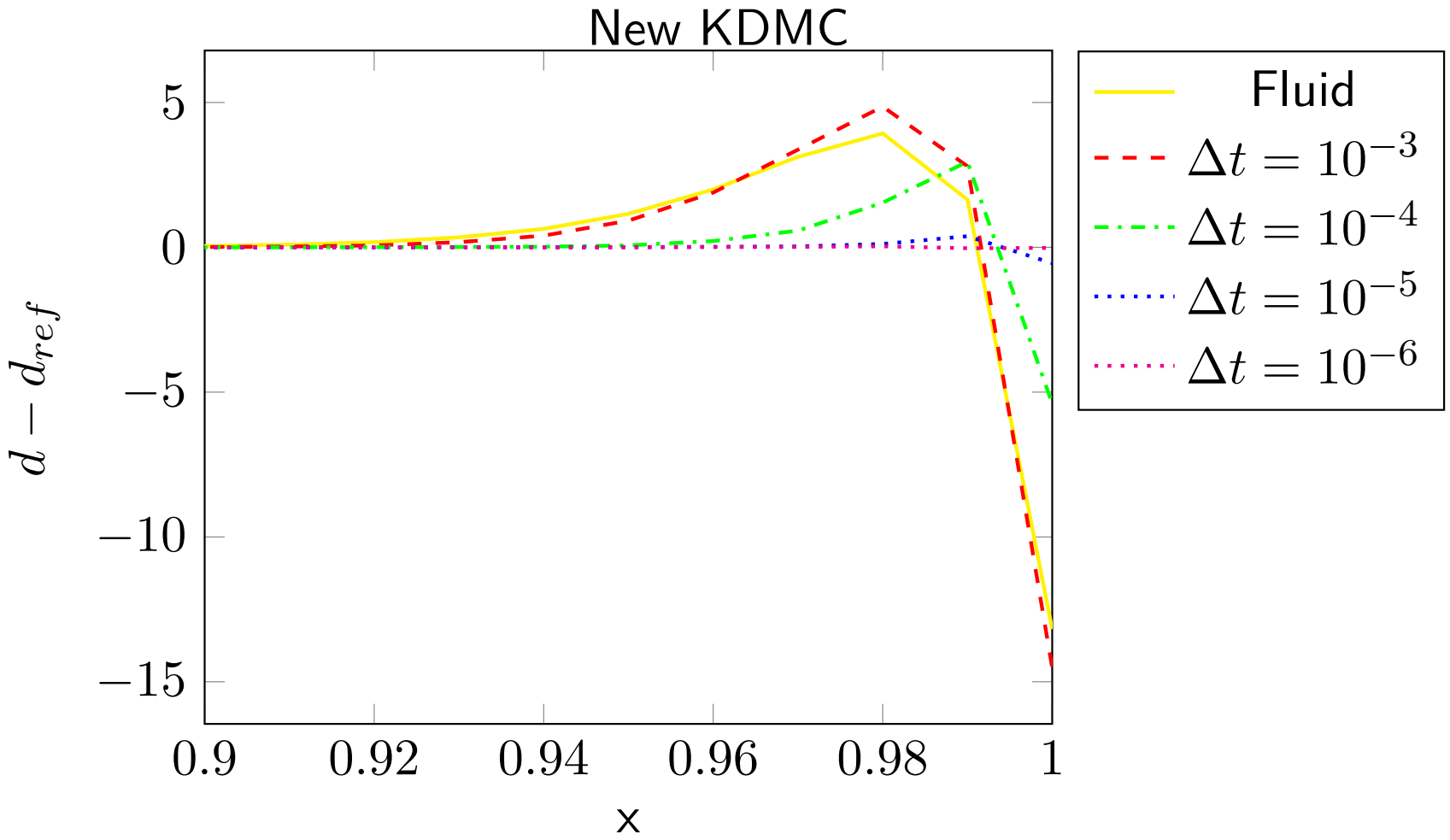
2 Boundary condition test 2

- Reflecting boundary
- Rate: $10^7 s^{-1}$
- BGK stddev: 10^7
- BGK mean: 100

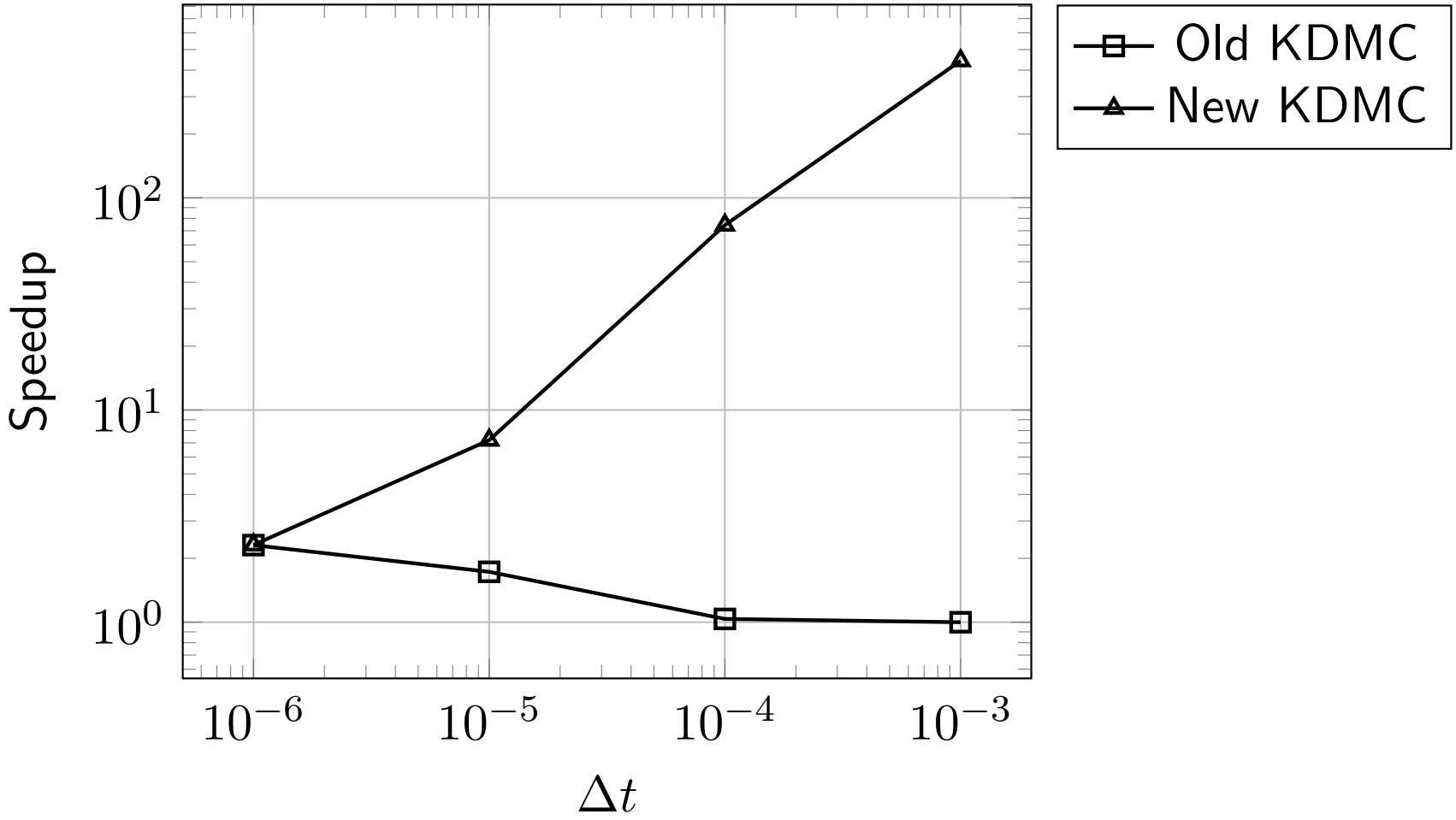
2 Boundary condition test 2



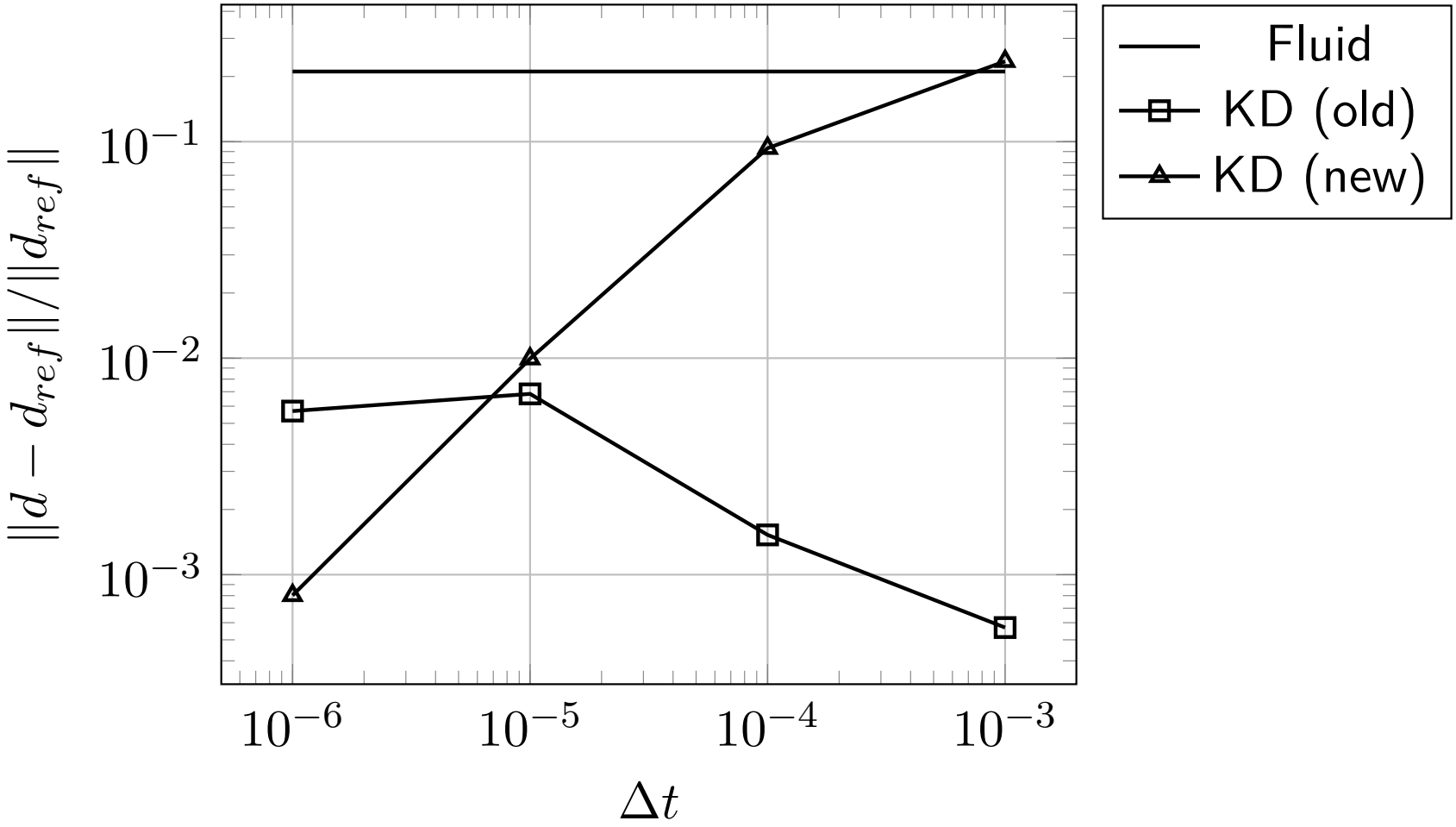
2 Boundary condition test 2



2 Boundary condition test 2



2 Boundary condition test 2



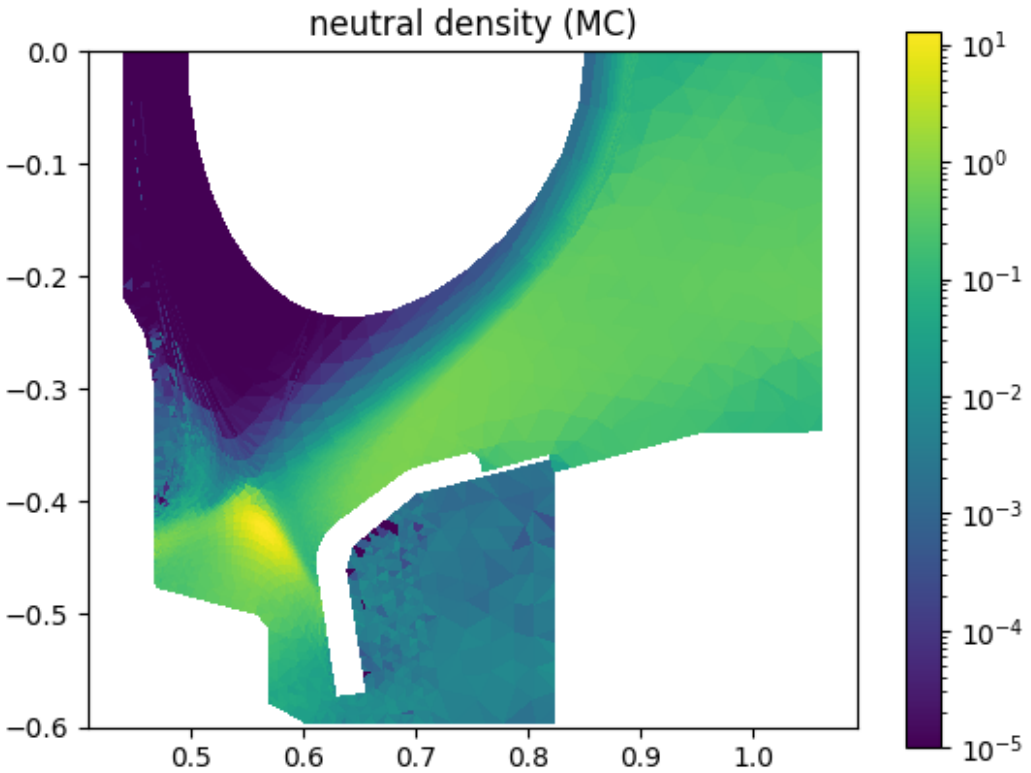
2 Conclusion

- New KDMC can accurately simulate the boundary conditions fluidly.
- But... fluid approximation is particularly inaccurate close to the boundary.
- Old KDMC is accurate (though slow) if there is drift towards the wall, which is the case in all of Bert's fusion relevant test cases. Is this just common in fusion?

3 Outline

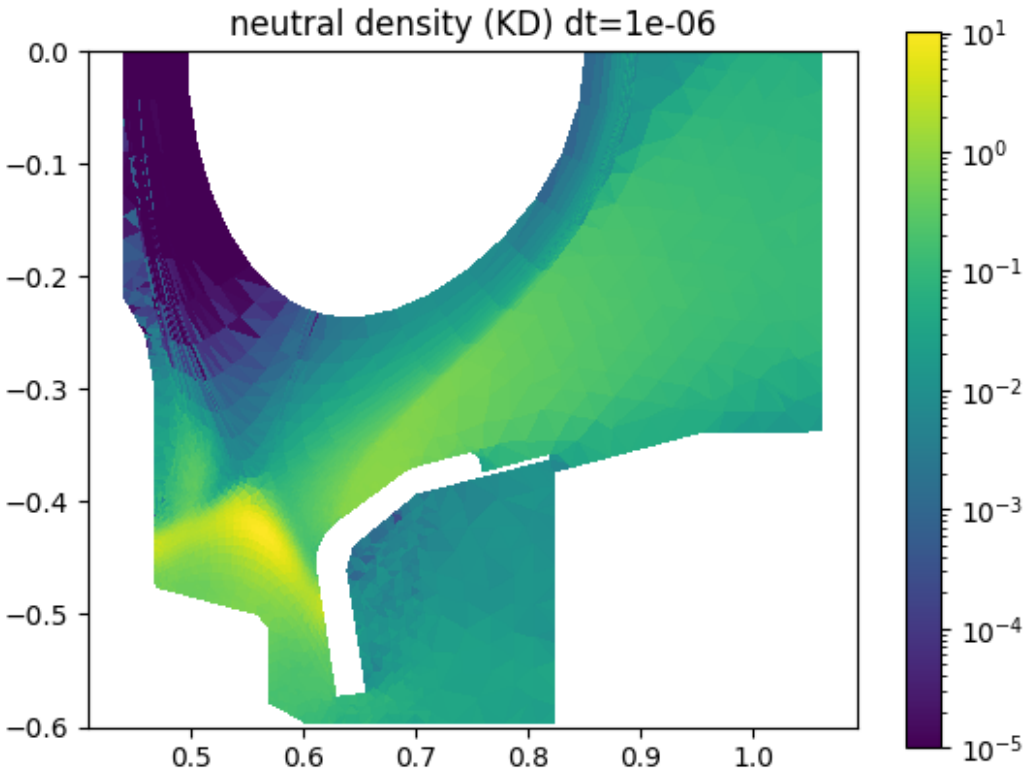
3 Early result: unstructured mesh KDMC

Alcator-CMOD mesh
runtime: 81s



3 Early result: unstructured mesh KDMC

Alcator-CMOD mesh
runtime: 44s



3 Implementations

	Bert Mortier	Eiron	Neptune
Language	Python	C++	C++
Dimensions	1D	2D	3D
Unstructured mesh	×	×	✓
Domain decomp.	×	✓*	×
Heterogeneous background	✓	×	✓
Fluid estimator	✓*	×	✓
Multilevel	✓*	×	×
Fluid BC	×	×	✓

Table: Comparison of features in different KDMC implementations.

Neptune is available at
https://gitlab.kuleuven.be/numa/software/neptune-mc

3 Outlook (possibilities)

- Multilevel KDMC
- Higher order SDE simulator
- Advanced fluid model for estimator
- Collection of smaller improvements (better BC sampler, optimized implementation for geometry calculations, ...)
- More realistic physics/grid, finally use KDMC on a "real" case.