Karlsruher Institut für Technologie

Structure Preserving Neural Networks: A Case Study in the Entropy Closure of the Boltzmann Equation

Steffen Schotthöfer, Tianbai Xiao, Martin Frank, Cory D. Hauck

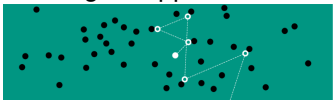
RESEARCH GROUP COMPUTATIONAL SCIENCE & MATHEMATICAL METHODS, KIT KARLSRUHE

Scales of physical modelling

Karlsruhe Institute of Technology

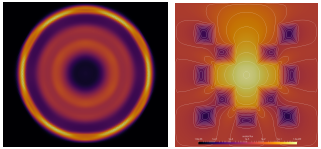
Different physical scales have to be considered, when modelling an application scenario

Newton's laws of motion: $m_i \ddot{x}_i = F_i(t, x, \dot{x})$



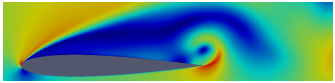
↓ Large number of particles ↓

Kinetic equations: $\partial_t f + v \nabla_x f = S(f)$



↓ Many collisions ↓

Macroscopic equations: $\partial_t \rho + \nabla_x g(\rho) = Q(\rho)$



The Boltzmann moment equation

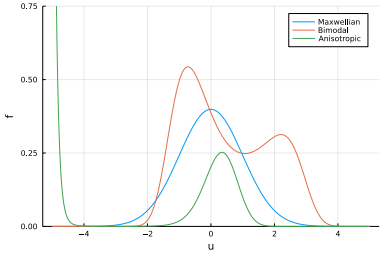
The moment hierachy of the Boltzmann equation is a powerful tool for kinetic systems!

Boltzmann moment system

$$\partial_t \mathbf{u} + \nabla \cdot \langle v f \mathbf{m} \rangle = \langle \mathbf{m} Q(f) \rangle$$

Two variables for one equation!

- $u(t,x) \in \mathbb{R}^N$ moment vector
- $f(t,x,v) \in \mathbb{R}_+$ kinetic density



This system is not closed! Need to find closure for the advection term $\langle v f(t,x,v) m(v) \rangle$.

Moment closures

- Truncation closure, i.e. linear reconstruction in the moment basis

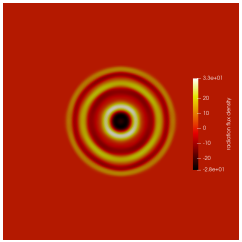
$$f_u(t,x,v) = \mathbf{u}(t,x) \cdot \mathbf{m}(v) \tag{1}$$

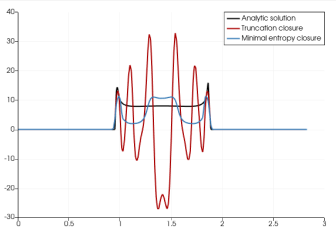
  Cheap to compute, but solution can be negative

- General neural network approximations of f [HMME19, HCCR21]

$$f_u(t,x,v) = \mathcal{N}_\theta(\mathbf{u}(t,x), \mathbf{m}(v)) \tag{2}$$

  Many approaches possible, but hard to preserve the systems structure.





Moment closures

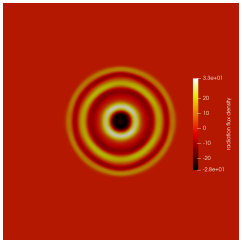
- Truncation closure, i.e. linear reconstruction in the moment basis

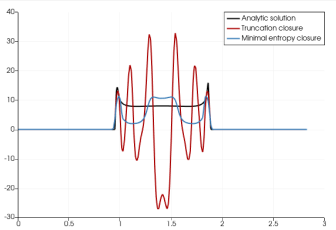
$$f_u(t,x,v) = \mathbf{u}(t,x) \cdot \mathbf{m}(v) \tag{1}$$

  Cheap to compute, but solution can be negative
- General neural network approximations of f [HMME19, HCCR21]

$$f_u(t,x,v) = \mathcal{N}_\theta(\mathbf{u}(t,x), \mathbf{m}(v)) \tag{2}$$

  Many approaches possible, but hard to preserve the systems structure.





Minimal entropy closure

Minimal entropy closure is a nonlinear reconstruction in the moment basis

$$f_u(v) = \eta_*'(\alpha_u \cdot \mathbf{m}(v)) \tag{3}$$

where α_u are the Lagrange multipliers of the convex

Dual minimal entropy closure problem

$$\alpha_u = \text{argmin}_{\alpha \in \mathbb{R}^N} \{h(\alpha)\} \tag{4}$$

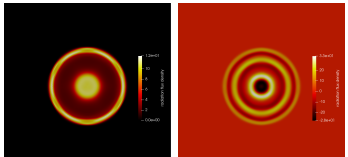
$$h(\alpha_u) = \langle \eta_*(\alpha_u \cdot \mathbf{m}(v)) \rangle - \alpha_u \cdot \mathbf{u} \quad \text{convex} \tag{5}$$

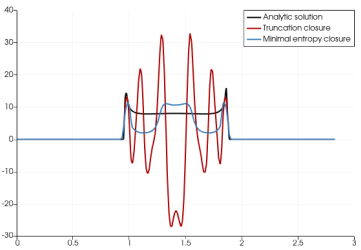
where $\eta(f)$ is the entropy density. The problem is convex, but ill-conditioned and hard to solve!

Entropy closures are good

Entropy functional $h(\alpha, u)$ is convex.
This implies structure preservation!

- Hyperbolicity of the moment system (simulation stability)
- Entropy decay, H-Theorem (physical property)
- Local conservation laws (Mass, Energy conservation)
- Positivity of solution
- Galilean invariance



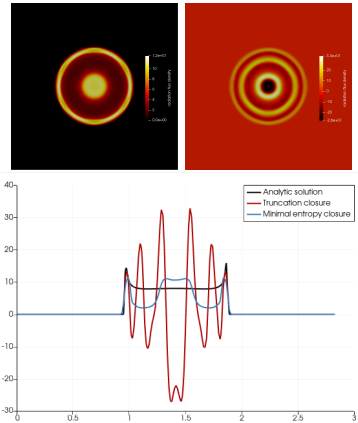


But they are very expensive! Over 90% of the simulation time needed!

Entropy closures are good

Entropy functional $h(\alpha, u)$ is convex.
This implies structure preservation!

- Hyperbolicity of the moment system (simulation stability)
- Entropy decay, H-Theorem (physical property)
- Local conservation laws (Mass, Energy conservation)
- Positivity of solution
- Galilean invariance



But they are very expensive! Over 90% of the simulation time needed!

Structure preserving neural networks

- Substitute optimization problem by trained neural network $\mathcal{N}_\theta$
- Accelerate the simulation
- Preserve the good structure of the minimal entropy closure

Main Idea:

- Preserve convexity of the entropy functional $h = \langle \eta_*(\alpha_u \cdot \mathbf{m}(v)) \rangle - \alpha_u \cdot \mathbf{u}$
- Do not train $\mathcal{N}_\theta$ on closure directly, but on $\mathbf{u} \mapsto h(\mathbf{u})$, obtain $\alpha_u = h'(\mathbf{u})$

Tools:

- Use an input convex neural network [AXK17]
- Or penalize non-monontonicity of the network derivative

Structure preserving neural networks

- Substitute optimization problem by trained neural network $\mathcal{N}_\theta$
- Accelerate the simulation
- Preserve the good structure of the minimal entropy closure

Main Idea:

- Preserve convexity of the entropy functional $h = \langle \eta_*(\alpha_u \cdot \mathbf{m}(v)) \rangle - \alpha_u \cdot \mathbf{u}$
- Do not train $\mathcal{N}_\theta$ on closure directly, but on $\mathbf{u} \mapsto h(\mathbf{u})$, obtain $\alpha_u = h'(\mathbf{u})$

Tools:

- Use an input convex neural network [AXK17]
- Or penalize non-monontonicity of the network derivative

Structure preserving neural networks

- Substitute optimization problem by trained neural network $\mathcal{N}_\theta$
- Accelerate the simulation
- Preserve the good structure of the minimal entropy closure

Main Idea:

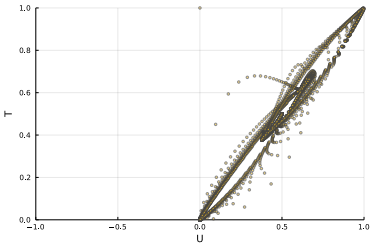
- Preserve convexity of the entropy functional $h = \langle \eta_*(\alpha_u \cdot \mathbf{m}(v)) \rangle - \alpha_u \cdot \mathbf{u}$
- Do not train $\mathcal{N}_\theta$ on closure directly, but on $\mathbf{u} \mapsto h(\mathbf{u})$, obtain $\alpha_u = h'(\mathbf{u})$

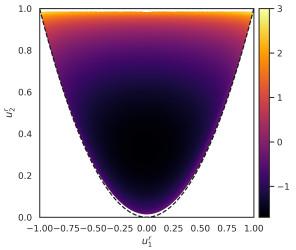
Tools:

- Use an input convex neural network [AXK17]
- Or penalize non-monontonicity of the network derivative

Training data generation

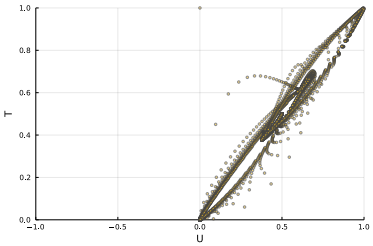
- **Idea 1:** Sample data by running simulations (end-to-end training)
  - Unclear training data distribution
  - Very problem specific
  - Expensive simulation needed
- **Idea 2:** Use the intricate problem structure to generate data
  - Gives strong control over data distribution
  - Allows error analysis
  - Much faster

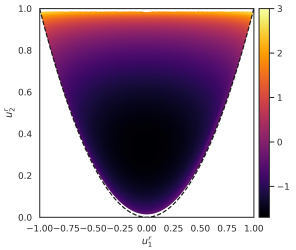




Training data generation

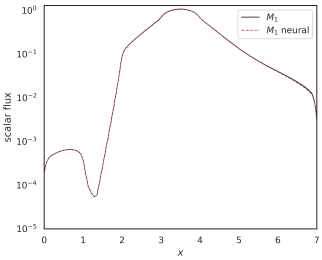
- **Idea 1:**Sample data by running simulations (end-to-end training)
  - Unclear training data distribution
  - Very problem specific
  - Expensive simulation needed
- **Idea 2:** Use the intricate problem structure to generate data
  - Gives strong control over data distribution
  - Allows error analysis
  - Much faster



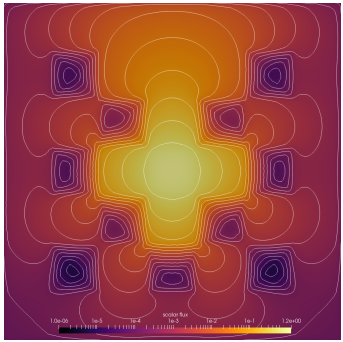


Simulation of a nuclear reactor

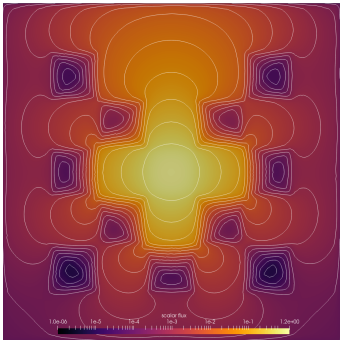
- We achieve the same accuracy as the benchmark simulation!



Cross section comparison



Benchmark solver



Network based solver

Computational Cost

- Over 87% of computational time saved!
- Makes entropy closure methods fast and physically accurate!

compute cores	Newton [s]	$\mathcal{N}_\theta$ [s]	Ratio [%]
4	757.88	80.81	89.33
12	258.64	33.60	87.01

Thank you for your attention!

You have questions or want to stay in touch?
Let me know and write me an email!
steffen.schotthoefer@kit.edu

(link to codebase)

References I

- Brandon Amos, Lei Xu, and J. Zico Kolter, *Input convex neural networks*, Proceedings of the 34th International Conference on Machine Learning (Doina Precup and Yee Whye Teh, eds.), Proceedings of Machine Learning Research, vol. 70, PMLR, 06–11 Aug 2017, pp. 146–155.
- Juntao Huang, Yingda Cheng, Andrew J. Christlieb, and Luke F. Roberts, *Machine learning moment closure models for the radiative transfer equation iii: enforcing hyperbolicity and physical characteristic speeds*, 2021.
- Jiequn Han, Chao Ma, Zheng Ma, and Weinan E, *Uniformly accurate machine learning-based hydrodynamic models for kinetic equations*, Proceedings of the National Academy of Sciences **116** (2019), no. 44, 21983–21991.