CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

Fast Evaluation of the Boltzmann Collision Operator Using Data Driven Reduced Order Models

Alex Alekseenko
California State University Northridge

NSF DMS-2111612, SFFP, XSEDE

Solving the Boltzmann Equation for Neutrino Transport in Relativistic Astrophysics

July 11, 2024

Overview

Development of efficient and accurate algorithms for simulating multidimensional flows of non-continuum non-equilibrium gases reflecting real gas effects

$$\frac{\partial}{\partial t} f_i(t,\vec{x},\vec{v}) + \vec{v}\cdot\vec{\nabla}_x f_i(t,\vec{x},\vec{v}) + \frac{1}{m_i}\vec{F}_i\cdot\vec{\nabla}_v f_i(t,\vec{x},\vec{v}) = \sum_{ij} Q[f_i,f_j](t,\vec{x},\vec{v})$$

Advantage of BE: Physics modelled on level of individual particles. Most accurate description.
Disadvantage: high dimensionality and high computational costs, esp. to account for particle interactions

- DSMC methods have been largely successful in providing scalable simulations
- Adaptive/high order methods (Arslanbekov, Kolobov, & Frolova, 2013; Taitano, Chacón, & Simakov, 2018, Boscheri & Dimarco, 2021)
- Low rank tensor representation of solutions (Boelens, Venturi, & Tartakovsky, 2020; Guo & Qiu, 2022; Chikitkin, Kornev, & Titarev, 2021, Taitano and Araki, 2024)
- Use of machine learning (Xiao & Frank, 2021, Miller, Roberts, Bond, & Cyr, JCP 2022)
- Methods of moments (Struchtrup & Frezzotti, 2022; Claydon, etal, 2017; Lockerby, B Collyer 2016; Djordjić, Pavić-Čolić, & Torrilhon, 2021)

This talk will focus on use of low rank/compressed approximations of solutions/equations and machine learning approaches to accelerate solutions of BE/collision operator

The Collision Operator

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

$$\frac{\partial}{\partial t} f_i(t, \vec{x}, \vec{v}) + \vec{v} \cdot \vec{\nabla}_x f_i(t, \vec{x}, \vec{v}) + \frac{1}{m_i} \vec{F}_i \cdot \vec{\nabla}_v f_i(t, \vec{x}, \vec{v}) = \sum_{ij} Q[f_i, f_j](t, \vec{x}, \vec{v})$$

Rarefied Gas Dynamics:

- Boltzmann collision integral

$$Q_{ij}[f_i, f_j](v) = \int_{R^3} \int_0^{2\pi} \int_0^b (f_i' f_j' - f_i f_j) B_{ij}(|g|, b, \epsilon) db d\epsilon \, du$$

- Relaxation models (BGK, ES-BGK, Shakhov)

$$Q_{ij}[f_i, f_j](v) = \frac{1}{\epsilon_{ij}} (f_{ij}^* - f_i)$$

Simulation of Plasma:

- Fokker-Plank with Rosenbluth potential

$$Q_{ij} = \Gamma_{ij} \nabla_v \cdot [D_j \nabla_v f_i - \frac{m_i}{m_j} A_j f_i], \ D_j = \nabla_v \nabla_v G_j, \ A_j = \nabla_v H_j,$$

$$\Delta_v H_j = -8\pi f_j, \ \Delta_v G_j = H_j$$

Fast Evaluation of the Collision Operator

CSUN CALIFORNIA STATE UNIVERSITY NORTHRIDGE

$$Q[f,f](v) = \int_{R^3}\int_0^{2\pi}\int_0^{b}(f(v')f(u') - f(v)f(u)B(|g|,b,\epsilon)dbd\epsilon\, du$$

- Advantage: accounts to interactions of individual molecules, most accurate physics
- Difficulty: $O(n^8)$ evaluation at one point in space. n is # of points in each velocity/mom. dim.
- Fast spectral methods: (Wu, etal. 2013; Mouhot & Pareschi, 2006; Gamba, Haack, Hauck, Hu, 2017)
- Nodal discontinuous Galerkin: $O(n^6)$ (A. & Josyula, 2014, A., Nguyen & Wood, 2015, A. & Limbacher, 2019)

$$Q_\varphi(v) = \int_{R^3}\int_{R^3} f(u-v)f(w-v)A_\phi(u,v)\, du \qquad A(u,v) = c\int_{S^2}(\phi(u') + \phi(w') - \phi(u) - \phi(w))B(|g|,\theta)d\theta$$

$$Q_k = \sum_{i,j} f_{i-k} f_{j-k} A_{ij}$$

(will become a single sum after a discrete Fourier transform)

Can an approximate low rank/fast collision model be learned for an application at hand?

- Optimal basis/moments to represent the solution are learned from data.
- Reduced models is a Galerkin discretization with some additional tricks to address stability and truncation errors
- Approach is suitable when many similar simulations need to be performed, e.g., in a grid parameter search.

Low Rank of a Tensor

T. Kolda and B. Bader, Tensor Decompositions and Applications SIAM Review 2009

- Tensor is a multi-index array: e.g., discrete kernel of collision operator A_{ijklmn} is an order 6 tensor (not using the Einstein notations)
- Rank One tensors:

$$a_i b_j c_k$$

$$x = \begin{bmatrix} .81 \\ .90 \\ .12 \\ .91 \end{bmatrix} \qquad y = \begin{bmatrix} .63 \\ .09 \\ .27 \\ .54 \end{bmatrix}$$

- CP Decomposition:

$A_{ijk} = \sum_{\sigma=1}^{r} a_i^\sigma b_j^\sigma c_k^\sigma$ smallest r is rank of the tensor

$$xy^T = \begin{bmatrix} .51 & .07 & .22 & .44 \\ .57 & .08 & .25 & .49 \\ .08 & .01 & .03 & .06 \\ .57 & .08 & .25 & .49 \end{bmatrix}$$

- CP: storage rdn vs. n^d; vector multiplication rn vs. n^d

Determining Rank of a Tensor

CP decompositions:

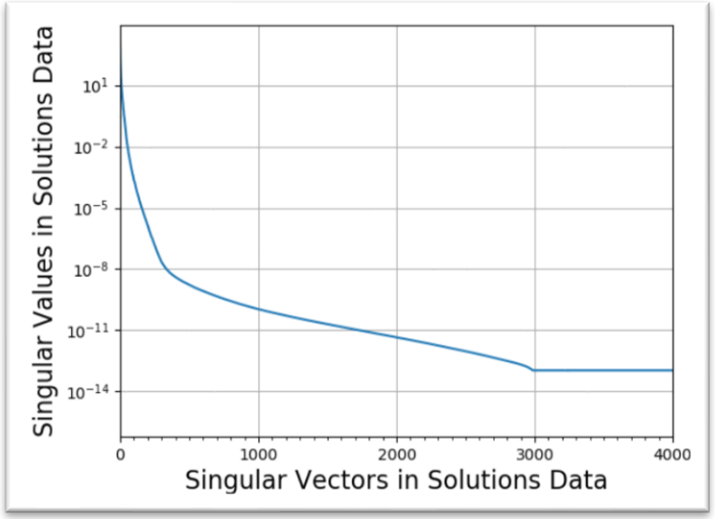
- Order two Matrices, SVD:

$$A_{ij} = \sum_{k=1}^{r} \sigma^k u_i^k v_j^k$$

- Can implement SVD compression

$$A_{ij}^* = \sum_{k=1}^{r^* \ll r} \sigma^k u_i^k v_j^k$$

$$\|A - A^*\|_2 \leq \sigma^{r^*+1}$$



Compressible if singular values decrease exponentially

Table 1
Typical rank values R for $p \times q \times q$ arrays

	$q = 2$	$q = 3$	$q = 4$	$q = 5$
$p = 2$	$\{2, 3\}$	$\{3, 4\}$	$\{4, 5\}$	$\{5, 6\}$
$p = 3$	3	$3 \leqslant R \leqslant 6$	$4 \leqslant R$	$5 \leqslant R$
$p = 4$	4	$4 \leqslant R \leqslant 6$	$4 \leqslant R$	$5 \leqslant R$
$p = 5$	4	$4 \leqslant R \leqslant 6$	$4 \leqslant R$	$5 \leqslant R$
$p = 6$	4	6	$6 \leqslant R$	$6 \leqslant R$
$p = 7$	4	7	$7 \leqslant R$	$7 \leqslant R$
$p = 8$	4	8	$8 \leqslant R$	$8 \leqslant R$

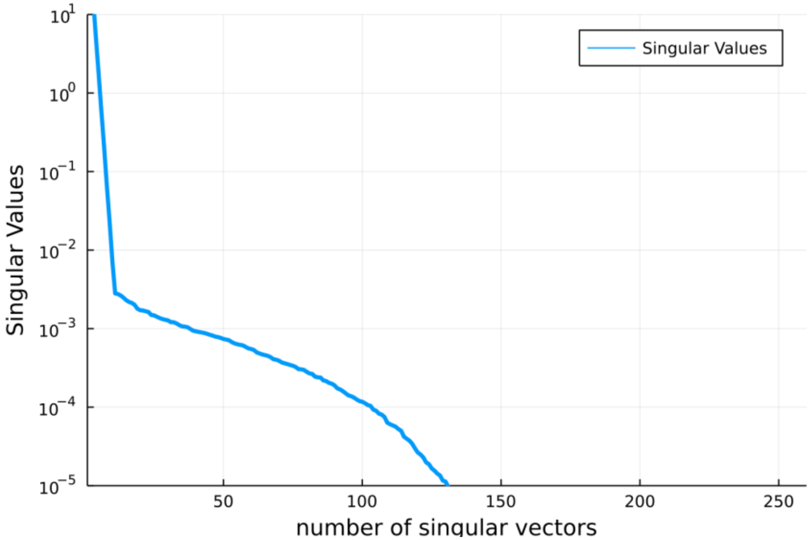
Berge and Stegeman, Linear Algebra and its Applications (2006)

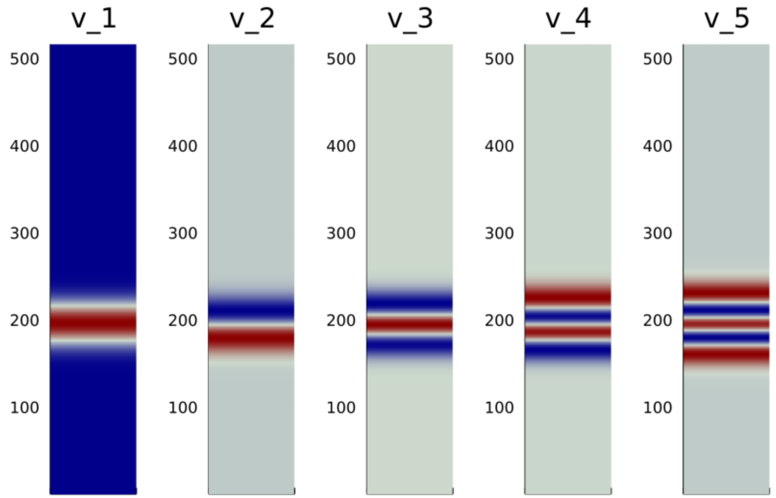
- For tensors of order >2 finding CP decomposition is NP complete problem
- Instead, use Tucker, High Order SVD, hierarchical Tucker, tensor train

$$A_{ijk} = \sum_{a,b,c=1}^{r_a, r_b, r_c} G_{abc} U_i^a V_j^b W_k^b$$

- Higher Order SVD: savings if ranks of kernels are low

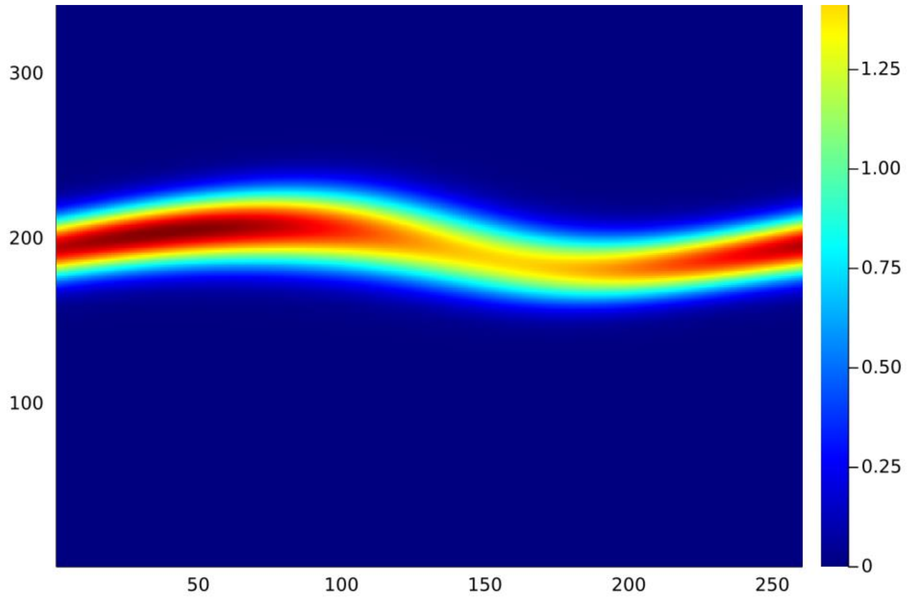
Compression of a Kinetic Solution

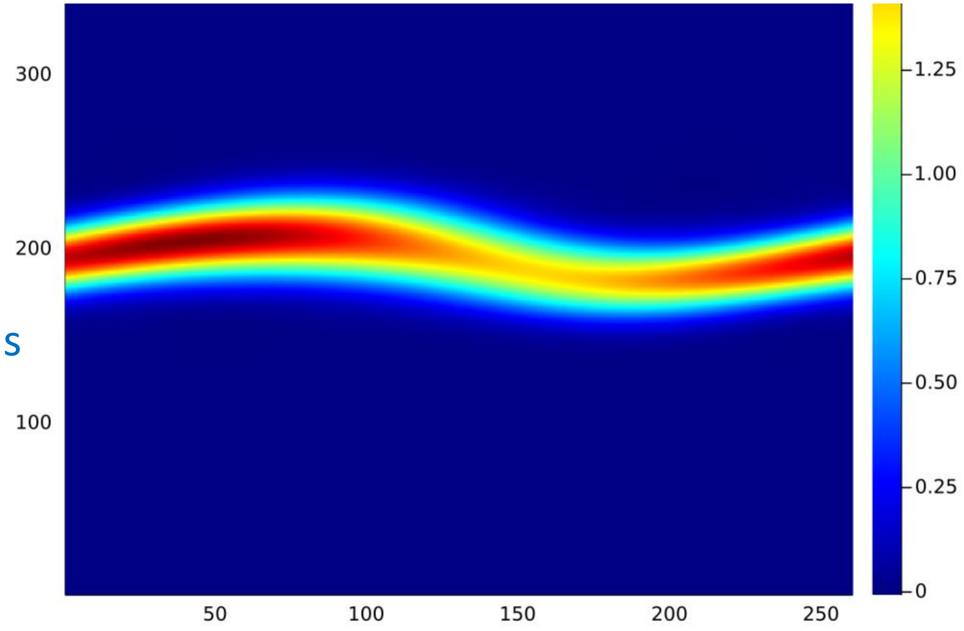




A snapshot of 1D1V ionic shock problem. Joint work with W. Taitano

Full vs. rank 5





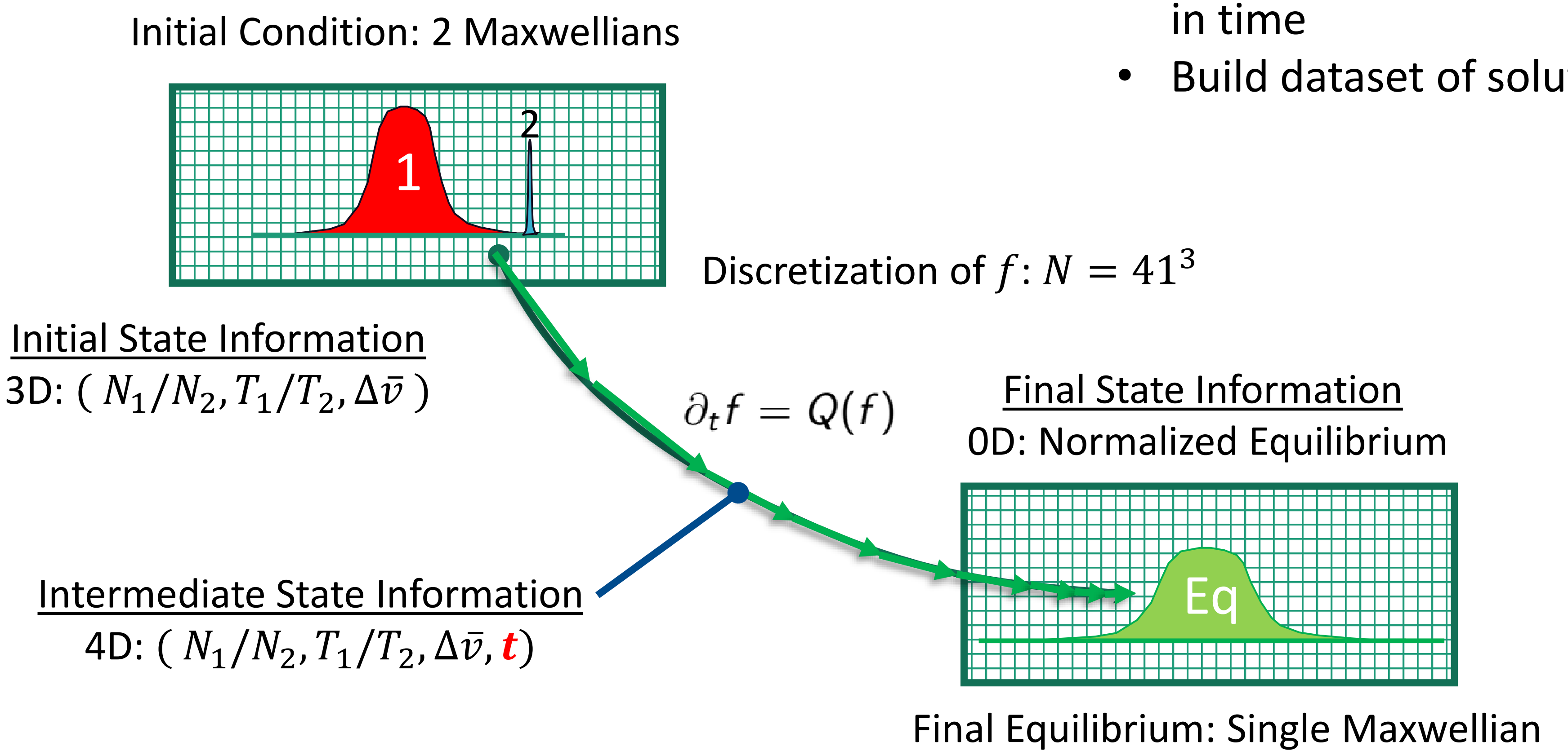
Idea:

- take bunch of solutions
- build compact basis
- use in discretization

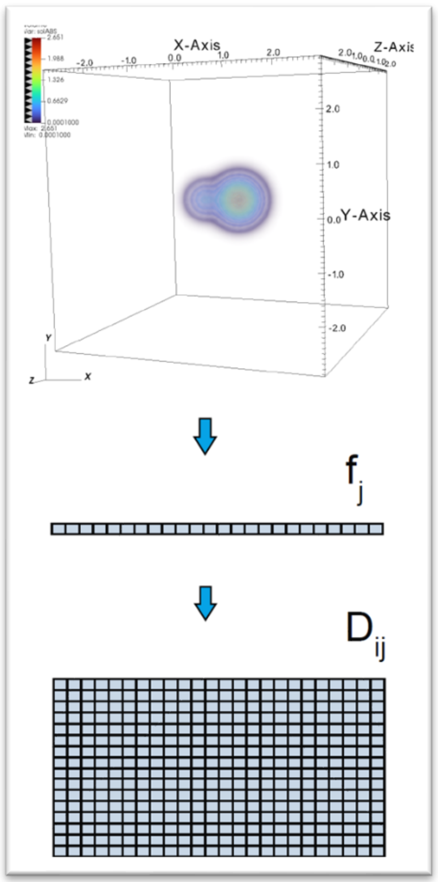
Data Driven ROM: Homogenous Relaxation of 2 Maxwellians

Joint work with R. Martin, U.S. Army Research Office, A. Wood, AFIT

- Generate initial two Maxwellians (random)
- Solve Sp. Hom. B.E.
- Sample Solutions at intervals in time
- Build dataset of solutions



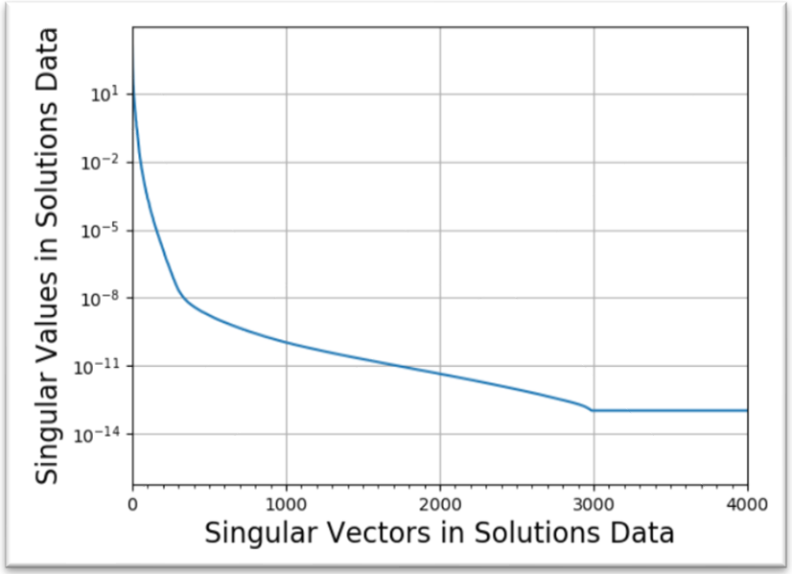
$\partial_t f = Q(f)$



Data points are time saves of solutions

"Naïve" SVD-Based ROM

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE



Singular values of D_{ij}

$$\left\| D_{ij} - \sum_{i=1}^{K} \sigma^k u_i^k v_j^k \right\|_2 \leq 10^{-3} \qquad \text{for } K \geq 40$$

ROM is a Galerkin discretization of the BCI using basis of first *K<100* right singular vectors v_i . Let H be matrix with columns v_i

DG convolution formulation:

$$\partial_t f_j = \sum_{j'=1}^{N} \sum_{j''=1}^{N} f_{j'-j} f_{j''-j} A_{j',j''}$$

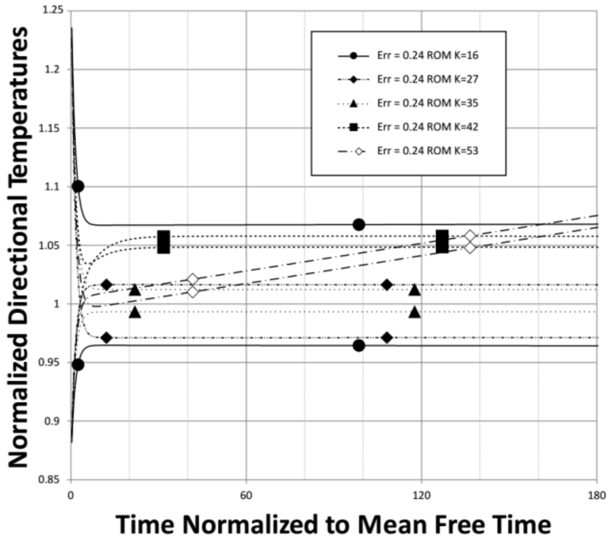
SVD ROM for Boltzmann collision operator:

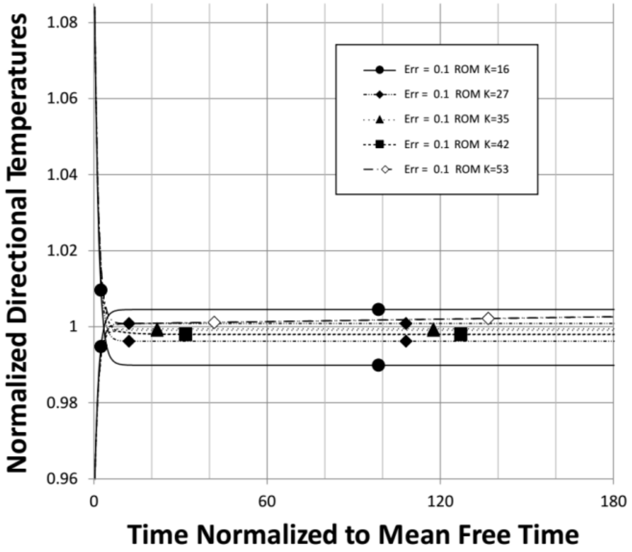
- Projection: $y=H^T f$. Recovery $f=Hy$
- $\partial_t(y)_k = \partial_t(Hf)_k = \sum_{k'=1}^{K} \sum_{k''=1}^{K} y_{k'} y_{k''} \hat{A}_{k',k'',k}$
- $\partial_t f_j = (H \partial_t y)_j$

(off-line stage) $$\hat{A}_{k',k'',k} = \sum_{j=1}^{N} \sum_{j'=1}^{N} \sum_{j''=1}^{N} H_{kj} H_{k',j'-j} H_{k'',j''-j} A_{j',j''}$$

Complexity: $O(K^3)$ as compared to $O(n^3), n = 41$
More than 10^2 speedup for K<40

Failure of the “Naïve” SVD-Based ROM



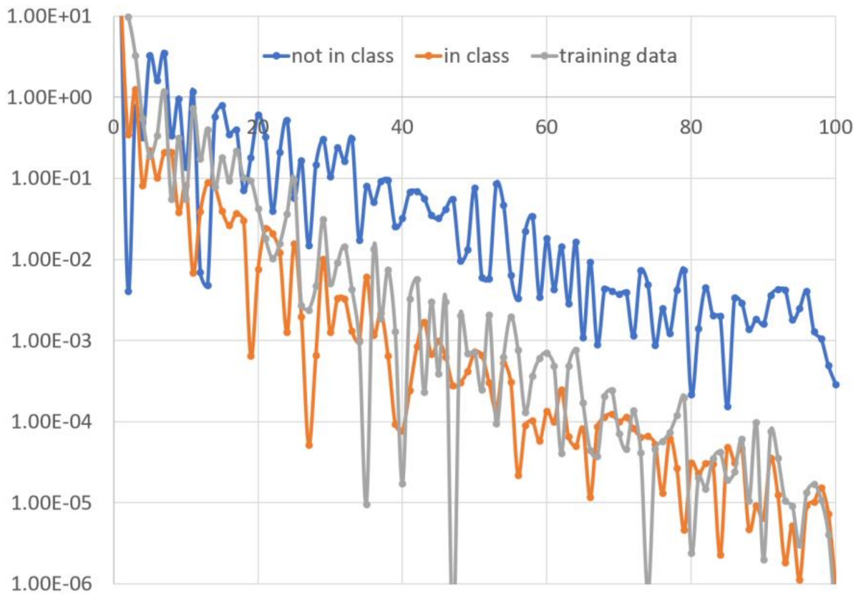


K	CPU time , s	Speedup vs. $O(n^6)$ Method (151 sec)
27	.19	7.9e2
35	.21	7.2e2
42	.27	5.6e2
53	.35	4.3e2

Time to evaluate collision operator equiv. 41^3 mesh for different sizes of ROM basis

- non-physical moments in the steady state regime
- Instabilities for large number of basis functions

Interpolation vs. Extrapolation:



Magnitudes of Coefficients in SVD ROM basis

The instability was related to having the steady state solution in the ROM while non-physical steady states are caused by residuals of ROM projection in initial data.

SVD ROM V2.0

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

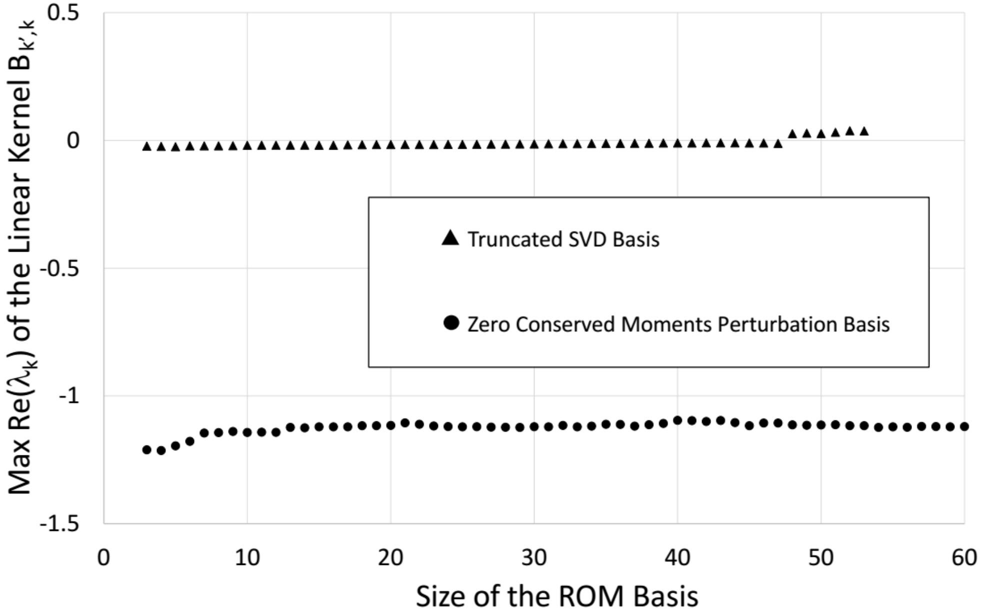
Improvements to the ROM:

- Remove steady state Maxwellian from the ROM
- Make ROM basis functions free form conservative moments.
- Re-write the model in terms of $df = f - f^M,\ \epsilon = H^T[df]$

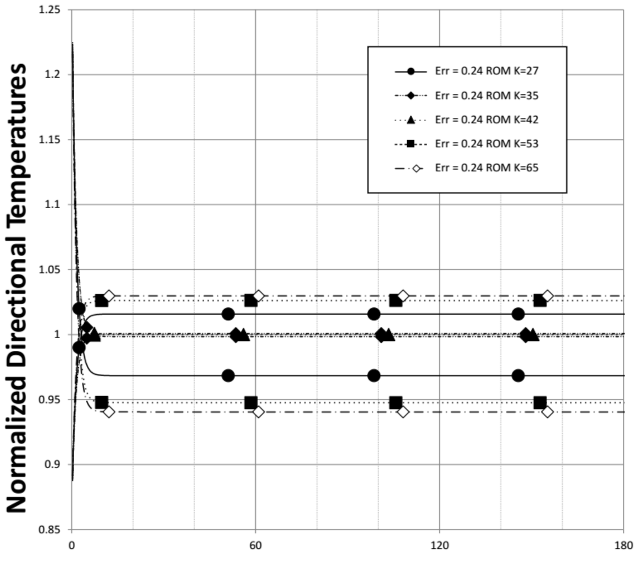
$$\partial_t \epsilon_k = \sum_{k'=1}^{K} \hat{B}_{k',k}\epsilon_{k'} + \sum_{k',k''=1}^{K} \hat{A}_{k',k'',k}\,\epsilon_{k'}\epsilon_{k''}$$

- Where

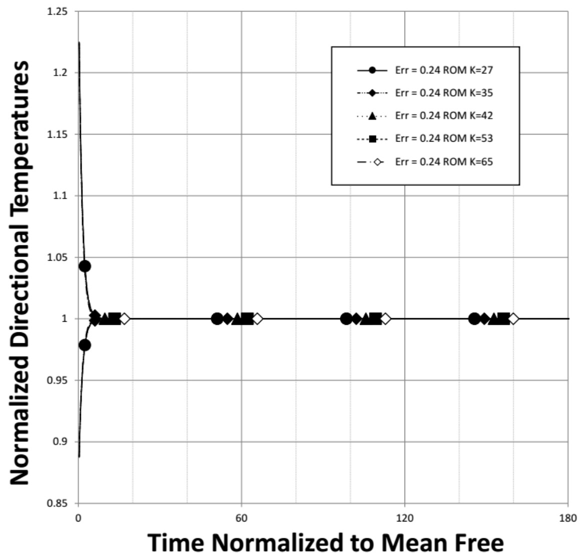
$$\hat{B}_{k',k} = 2\sum_{j,j',j''=1}^{n^3} H_{kj}H_{k',j'-j}\,f^M_{j''-j}A_{j',j''}$$



Largest real parts of eigenvalues of $B_{k',k''}$. Naïve ROM has a positive eigenvalue for $K > 47$. Eigenvalues of Maxwell-free ROM are below -1.



No residual damping



Residual damping

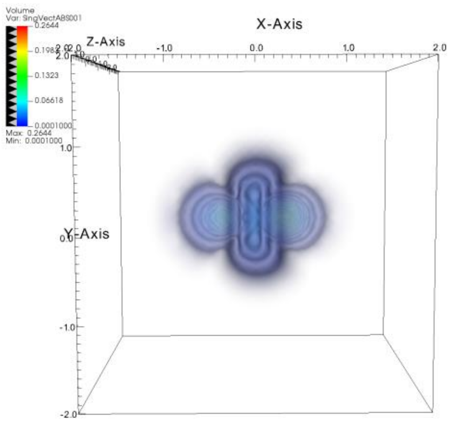
- Damping ROM Residual (Wong & Cai, 2019):
- Define

$$\epsilon^{\perp} = (I - HH^T)f$$

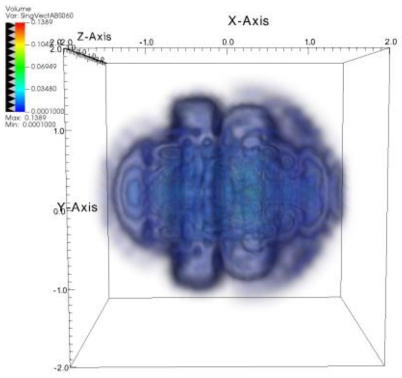
- Evolve

$$\partial_t \epsilon^{\perp} = -\nu\epsilon^{\perp}$$

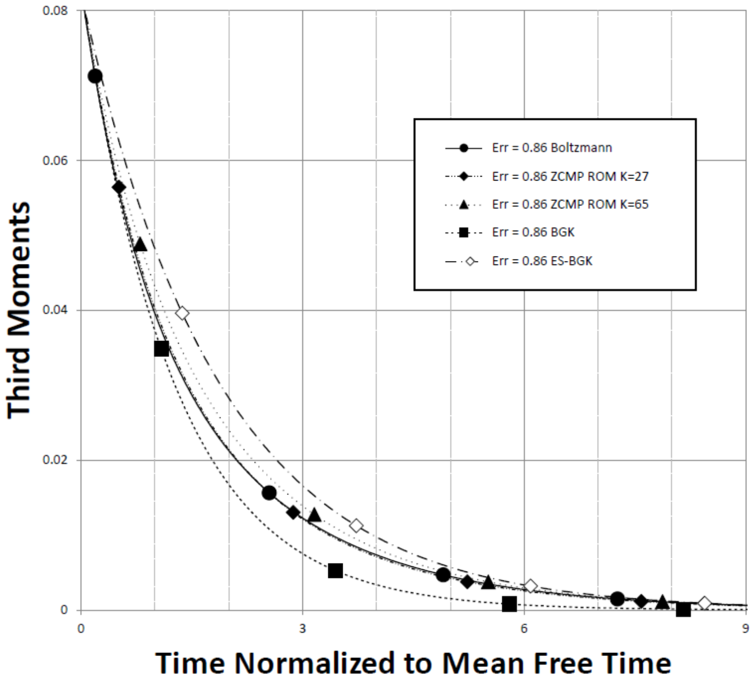
An Issue with the ROM Basis



Basis function #2

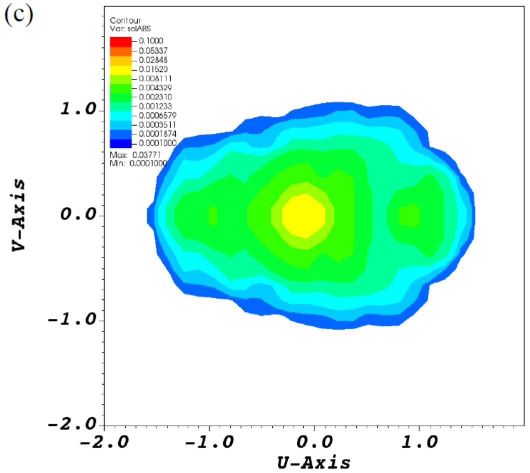


Basis function #60

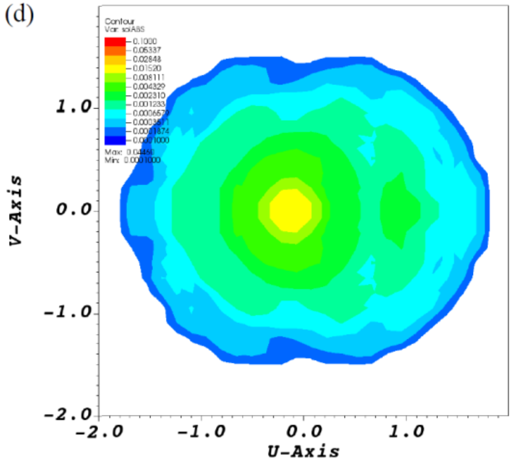


Counterintuitive Observation: Moments computed using larger ROMs are less accurate

- Support of basis vectors grows and becomes more oscillatory
- These oscillations affect residual and higher moments in solutions
- As a result, solutions for large ROM basis show less accurate predictions of higher moments while resolving stronger features in solutions
- Need a better POD for kinetic solutions or a better model to control the residual.



Error in ZCMP ROM k=27



Error in ZCMP ROM k=65

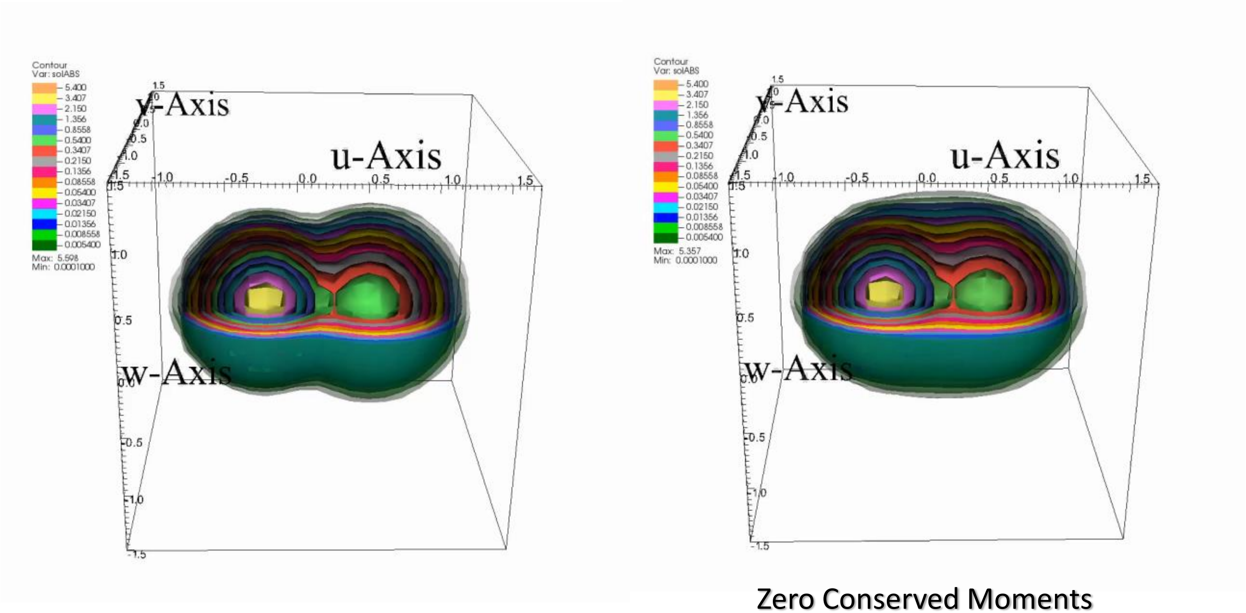
Overall Performance of ROMs

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

Use of ROMs improves computation time for BCI:

Approach	Cost	CPU Time	Speed-up	%
Nodal-DG	$O(M^6)$	151 sec		
ROM, K=53	$O(K^3)$	0.35 sec	431	0.2%

CPU time to evaluate collision operator. Acceleration is relative to $O(n^6)$ nodal-DG discretization on 41^3 velocity mesh (151 sec).



Nodal-DG Velocity Discretization

Zero Conserved Moments Perturbation ROM

Spatially Homogeneous Relaxation of two Homogeneous Gauss Densities

- ROMs can approximate Boltzmann collision operator efficiently.
- ROM approach can be applied to 2D and 3D flows.
- Extension to 1D is underway.
- Off-line training phase is expensive. Method is suitable for problems where many similar computations need to be performed, e.g., grid parameter search.
- Work is underway to learn ROM on-line.

Support: AFOSR F4FGA08305J005, FA955020QCOR100, NSF DMS-1620497, DMS-2111612, SFFP, XSEDE

Approximation of Collision Operator using Convolutional Neural Network

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

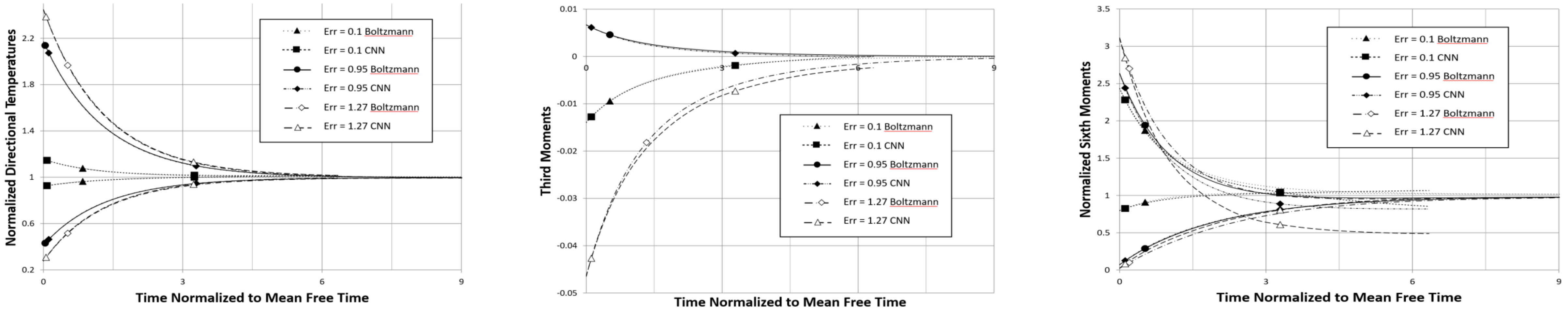
Results: Tom Nguyen, CSUN

Convolutional Neural Network (CNN) was trained to predict values of the collision operator for the class of solutions discussed above. The CNN predictions were used to approximate collision operator in Euler time stepping scheme to solve

$$\partial_t f = Q_{CNN}(f)$$

Structure of the CNN:

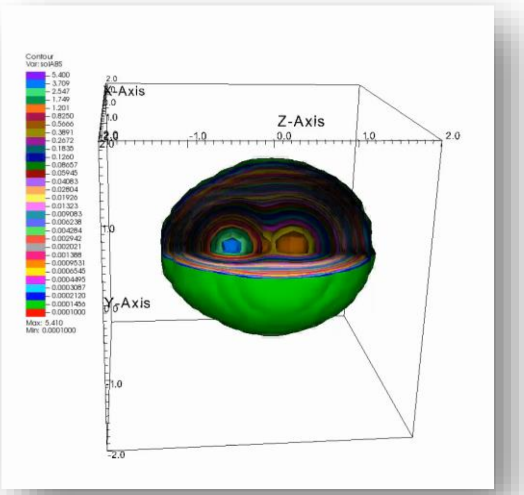
- Input: discrete f at 41^3 points
- 1^{st} hidden layer: 4 filters; 5x5x5 kernel
- 2^{nd} hidden layer: 8 filters; 3x3x3 kernel
- 3^{rd} hidden layer: max pooling; 2x2x2 kernel
- 4^{th} hidden layer: 16 filters; 3x3x3 kernel
- 5^{th} hidden layer: max pooling; 2x2x2 kernel
- 6^{th} hidden layer: 32 filters; 3x3x3 kernel
- Output: fully connected, discrete collision Q_{CNN} on 41^3 points
- parametric leaky ReLU, MAE loss, adamax optimizer.
- One evaluation 0.18 sec



Relaxation of moments in solutions obtained using CNN approximation of collision operator

Further Thoughts on Use of Machine Learning

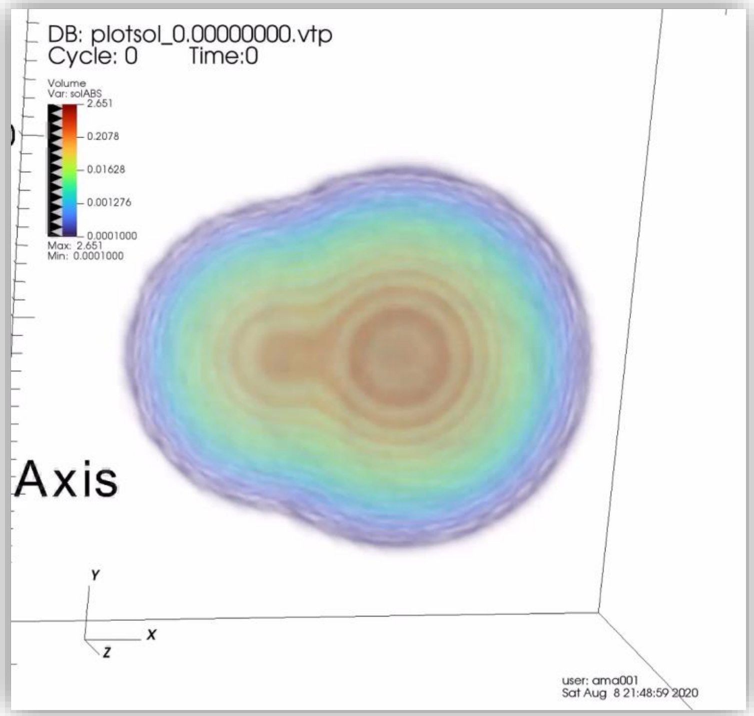
- Can approximate the collision operator
- Special architectures are needed to enforce stability and error for solutions “not-in-class” solutions.
- The expensive off-line training stage is limiting applications



Spurious oscillations in CNN solutions

- Growing body of literature on using neural networks, and machine learning in general, in computational mathematics
- Current efforts seem to be directed toward using on-line training
- In **T. Xiao & M. Frank, JCP 2021** neural network correction to BGK term is learned using information from fast spectral solver. Solver applied to 0D, 1D1V, 2D2V problems. Architecture enforces conservation laws.
- In **S. Miller, N. Roberts, S. Bond, E. Cyr, JCP 2022**, neural network correction to BGK is learned based on DSMC solver data. Architecture enforces conservation laws.
- Overall, use of neural network still seems promising.

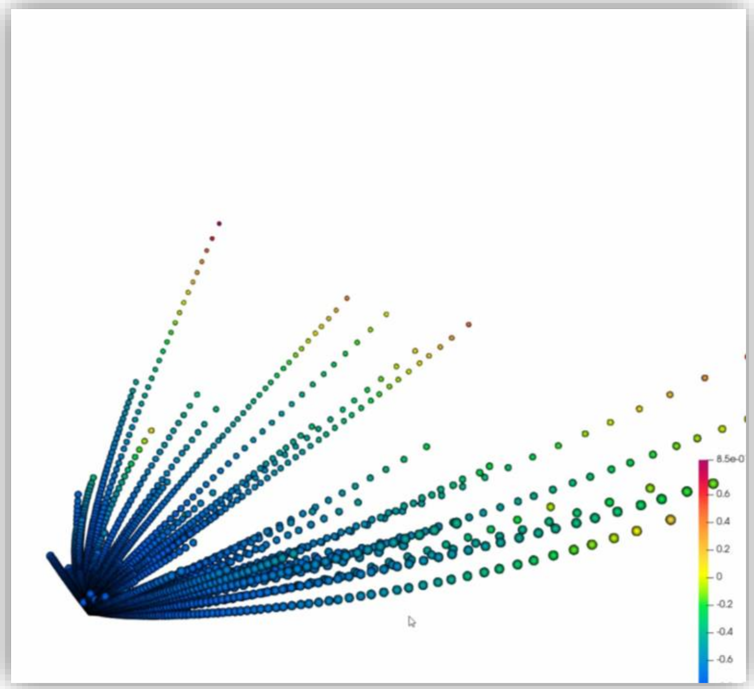
Exploring Low Dimensional Structure of Solutions



Velocity Distribution function

Observations

- Class of solutions depends on three parameters and time
- Trajectories look very simple in SVD basis
- Kinetic Entropy $H(f) = \int_{\mathbb{R}^3} f(\vec{v}) \ln f(\vec{v})\, dv$ decreases monotonically in time and could be a candidate for the potential function.



Trajectories of solutions in the basis of first three singular vectors of D_{ij} .

Potential Directions

- Find NN approximation to s.p.d. matrix $\Theta(f)$ such that
  $$\langle \partial_t f(t,\vec{v}), \phi(v) \rangle = -\langle \nabla H, \Theta(f)\phi(\vec{v}) \rangle$$
- Recover trajectories by integrating along $\Theta(f)\nabla H$
- Formulate a minimization problem using appropriate action, see e.g. Erbar 2016

Thank You Very Much for Your Attention!

Any Questions?