Some computational methods for kinetic transport equations

Yingda Cheng

Michigan State University

Seminar, Umeå University,, June 2022

Supported by NSF DMS-2011838, AST-2008004.

Outline

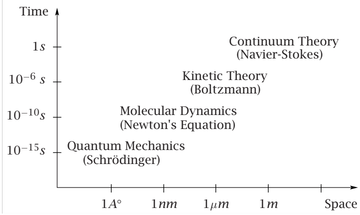
1 Introduction

2 Sparse grid DG method

3 Reduced basis method

4 Machine learning moment closure method

Kinetic equations



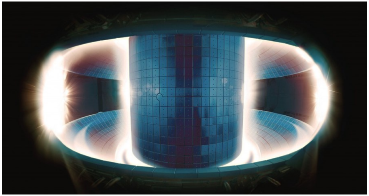
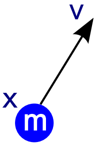


Figure: Left: Figure from E & Engquist, AMS Notice, 2003, Right: tokamak device

Kinetic models are widely used in many applications: plasma, astrophysics, rarefied gas dynamics, etc.

Boltzmann transport equation: an example

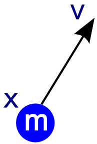


Gas Dynamics:
Position **x**, velocity **v**, time t, external force **F**

Equations of motion

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{v} \\ \frac{d\mathbf{v}}{dt} = \frac{\mathbf{F}}{m} \end{cases}$$

Boltzmann transport equation: an example



Gas Dynamics:

Position $\mathbf{x}$, velocity $\mathbf{v}$, time t, external force $\mathbf{F}$

Equations of motion

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{v} \\ \frac{d\mathbf{v}}{dt} = \frac{\mathbf{F}}{m} \end{cases}$$

The Boltzmann equation considers the **Probability Density Function (pdf)** $f(t, \mathbf{x}, \mathbf{v})$

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f = Q(f, f) \quad \textbf{Collision}$$

Q(f)=0 collisionless case: **Vlasov equation**

Kinetic simulations

- Two classes of methods: deterministic and probablistic.
- Probablistic methods: trace the particles, and have statistical noise.
- Deterministic methods: directly solve the PDEs, no noise.
- Computational challenges: $\underline{\text{high dimensions (3D+3V)}}$, conservation properties, multiple scales.
- 100 grid cell in each dimension, means 10^{12} points.

Overview

We developed several approaches for “reduced order modeling” of kinetic simulations. This means we want to manage the computational cost for such high dimensional system.

- Sparse grid discontinuous Galerkin (DG) method. Provide accurate and adaptive simulations for high dimensional PDEs.

  Wang *et al* JCP, 2016, Guo, Cheng, SISC, 2016, 2017, Tao *et al* JCP, SISC, 2019, Liu *et al*, JCP 2019, Tao *et al*, JCP, 2020, Huang *et al*, SISC 2020...

Overview

By building surrogate models,

- Reduced basis method (RBM). We build reduced order models computationally using the RBM approach.

  Peng, Chen, Cheng, Li, 2021.

- Machine learning based moment closure methods. Based on the moment methods, we use a data-driven approach to develop effective closure models.

  Huang, Cheng, Christlieb, Roberts, 2021.

Of course, this is a very active field, and there are many related important work in the literature!

Outline

1. Introduction

2. Sparse grid DG method

3. Reduced basis method

4. Machine learning moment closure method

Motivation

- Sparse grid. A technique for breaking the curse of dimensionality. Smolyak (63), Zenger (91), Griebel (91,98,05...)
- DG method. Reed and Hill (73), Cockburn and Shu (89, 90,...).

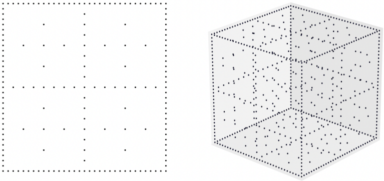


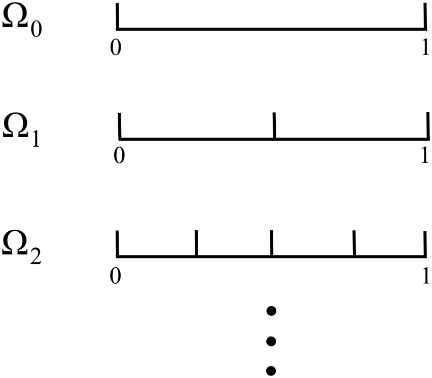
Fig. 5: Two-dimensional sparse grid (left) and three-dimensional sparse grid (right) of level $n = 5$.

Figure: From Garcke, SG in a nutshell

The ideas

Consider $\Omega = [0,1]$ and define n-th level grid

$$\Omega_n = \{I_n^j = (2^{-n}j, 2^{-n}(j+1)], j = 0, \ldots, 2^n - 1\}$$



Hierarchical decomposition of piecewise polynomial spaces in one dimension

Conventional approximation space on the n-th level grid Ω_n

$$V_n^k = \{v : v \in P^k(I_n^j), \forall j = 0, \dots, 2^n - 1\}$$

$$dim(V_n^k) = 2^n(k+1)$$

Nested structure

$$V_0^k \subset V_1^k \subset V_2^k \subset V_3^k \subset \cdots$$

W_n^k: orthogonal complement of V_{n-1}^k in V_n^k, for $n > 1$, represents the finer level details when the mesh is refined, satisfying

$$V_{n-1}^k \oplus W_n^k = V_n^k$$

$$W_n^k \perp V_{n-1}^k$$

Let $W_0^k := V_0^k$, then

$$V_N^k = \bigoplus_{0 \le n \le N} W_n^k$$

$$dim(W_n^k) = \left\lceil 2^{n-1} \right\rceil (k+1)$$

Hierarchical orthonormal bases: Alpert's multiwavelet

Bases in W_0^k: scaled orthonormal Legendre polynomials.

Bases in W_1^k:

$$h_i(x) = 2^{1/2} f_i(2x-1), \quad i = 1, \dots, k+1$$

The orthonormal, vanishing-moment functions $\{f_i(x)\}_k$ (Alpert 93), which are supported on $(-1,1)$ and depend on k, will be defined later.

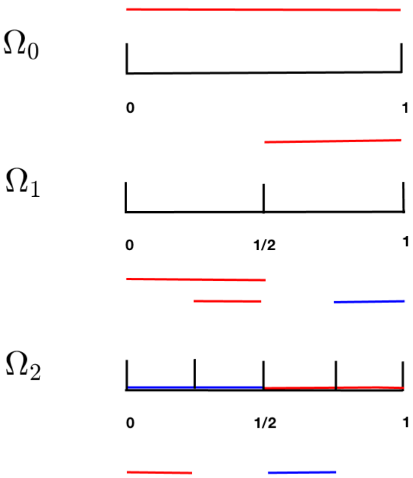
Bases in W_n^k, $n \geq 1$

$$v_{i,n}^j(x) = 2^{(n-1)/2}\, h_i(2^{n-1}x - j), \quad i = 1, \dots, k+1, \, j = 0, \dots, 2^{n-1}-1$$

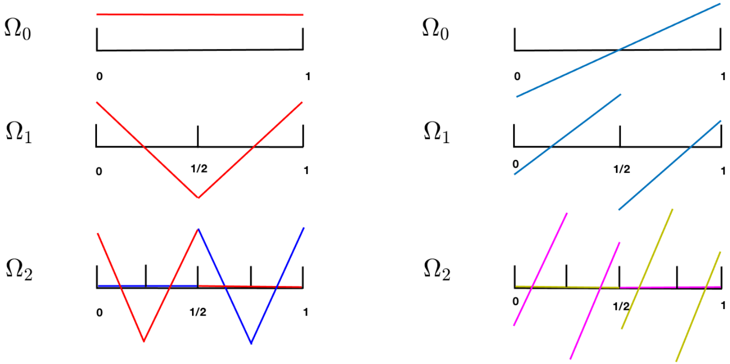
Orthonormality of multiwavelet bases across different hierarchical levels

$$\int_0^1 v_{i,n}^j(x) v_{i',n'}^{j'}(x)\, dx = \delta_{ii'} \delta_{nn'} \delta_{jj'}$$

Bases on different levels for $k = 0$



Bases on different levels for $k = 1$



Approximation space in multi-dimensions

Consider 2D case, $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and multi-index $\mathbf{l} = (l_1, l_2) \in \mathbb{N}_0^2$
The standard rectangular grid $\Omega_\mathbf{l}$ with mesh size

$$h_\mathbf{l} := (2^{-l_1}, 2^{-l_2})$$
$$h := \min\{2^{-l_1}, 2^{-l_2}\}$$

For each $I_\mathbf{l}^\mathbf{j} = \{(x_1, x_2) : x_i \in (2^{-l_i} j_i, 2^{-l_i}(j_i + 1)]\}$, the traditional tensor-product polynomial space is

$$\mathbf{V}_\mathbf{l}^k = \{\mathbf{v} : \mathbf{v}(\mathbf{x}) \in P^k(I_\mathbf{l}^\mathbf{j}),\ \mathbf{0} \le \mathbf{j} \le 2^\mathbf{l} - \mathbf{1}\}$$

P^k denotes polynomial of degree at most k in each dimension.

Approximation space in multi-dimensions

Consider 2D case, $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and multi-index $\mathbf{l} = (l_1, l_2) \in \mathbb{N}_0^2$
The standard rectangular grid $\Omega_{\mathbf{l}}$ with mesh size

$$h_{\mathbf{l}} := (2^{-l_1}, 2^{-l_2})$$
$$h := \min\{2^{-l_1}, 2^{-l_2}\}$$

For each $I_{\mathbf{l}}^{\mathbf{j}} = \{(x_1, x_2) : x_i \in (2^{-l_i} j_i, 2^{-l_i}(j_i + 1)]\}$, the traditional tensor-product polynomial space is

$$\mathbf{V}_{\mathbf{l}}^k = \{\mathbf{v} : \mathbf{v}(\mathbf{x}) \in P^k(I_{\mathbf{l}}^{\mathbf{j}}),\ \mathbf{0} \le \mathbf{j} \le 2^{\mathbf{l}} - \mathbf{1}\}$$

P^k denotes polynomial of degree at most k in each dimension. Uniform grid: $l_1 = l_2 = N$, $\mathbf{V}_{\mathbf{l}}^k = \mathbf{V}_N^k$, then

$$\mathbf{V}_N^k := V_{N,x_1}^k \times V_{N,x_2}^k = \bigoplus_{|\mathbf{l}|_\infty \le N} \mathbf{W}_{\mathbf{l}}^k$$

where

$$\mathbf{W}_{\mathbf{l}}^k := W_{l_1,x_1}^k \times W_{l_2,x_2}^k$$

The basis functions for $\mathbf{W}_{\mathbf{l}}^k$ can be defined by a tensor product

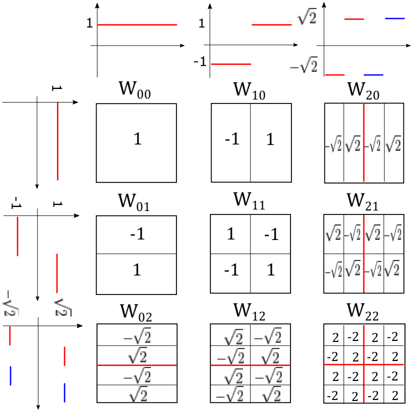
$$v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x}) := \prod_{t=1}^{2} v_{i_t,l_t}^{j_t}(x_t), \quad j_t = 0, \ldots, \max(0, 2^{l_t - 1} - 1), \quad i_t = 1, \ldots, k+1$$

Full grid approximation space

Full grid space:

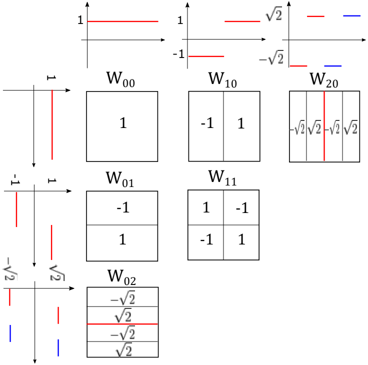
$$\mathbf{V}_N^k = \bigoplus_{|\mathbf{l}|_\infty \le N} \mathbf{W}_\mathbf{l}^k$$

$d = 2,\ N = 2,\ k = 0$



Sparse grid approximation space

We consider the sparse grid space: $\hat{\mathbf{V}}_N^k := \bigoplus_{|\mathbf{l}|_1 \le N} \mathbf{W}_\mathbf{l}^k$



A viewpoint without using multiwavelet space: $\hat{\mathbf{V}}_N^k = \bigoplus_{|\mathbf{l}|_1 \le N} \mathbf{V}_\mathbf{l}^k$.

$$dim(\hat{\mathbf{V}}_N^k) = O(2^N N^{d-1}(k+1)^d) \quad \text{or} \quad O(h^{-1}|\log_2 h|^{d-1})$$

DG method on sparse grids: linear transport problems

Consider the linear transport equation with variable coefficient

$$\begin{cases} u_t + \nabla \cdot (\boldsymbol{\alpha}(\mathbf{x}, t)\, u) = 0, \quad \mathbf{x} \in \Omega = [0,1]^d, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), \end{cases} \tag{1}$$

The semi-discrete DG formulation for (1) is defined as follows: find $u_h \in \hat{\mathbf{V}}_N^k$, such that

$$\int_\Omega (u_h)_t\, v_h\, d\mathbf{x} = \int_\Omega u_h \boldsymbol{\alpha} \cdot \nabla v_h\, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \widehat{\boldsymbol{\alpha} u_h} \cdot [v_h]\, ds, \tag{2}$$
$$\doteq A(u_h, v_h)$$

for $\forall\, v_h \in \hat{\mathbf{V}}_N^k$, where $\widehat{\boldsymbol{\alpha} u_h}$ defined on the element interface denotes a monotone numerical flux.

Stability (constant coefficient case)

Theorem (L^2 stability)

The DG scheme (2) *for* (1) *is* L^2 *stable when* $\boldsymbol{\alpha}$ *is a constant vector, i.e.*

$$\frac{d}{dt}\int_\Omega (u_h)^2\, d\mathbf{x} = -\sum_{e\in\Gamma}\int_e \frac{|\boldsymbol{\alpha}\cdot\mathbf{n}|}{2}|[u_h]|^2 ds \le 0. \tag{3}$$

Error estimate (constant coefficient case)

Similar to Schwab, Suli, Todor (08), we can establish error estimate in L^2 norm for the L^2 projection operator, combining with an estimate for DG method, we get

Theorem (L^2 error estimate)

Let u be the exact solution, and u_h be the numerical solution to the semi-discrete scheme (2) with numerical initial condition $u_h(0) = \mathbf{P}u_0$. For $k \geq 1$, $u_0 \in \mathcal{H}^{p+1}(\Omega)$, $1 \leq q \leq \min\{p, k\}$, $N \geq 1$, $d \geq 2$, we have for all $t \geq 0$,

$$\|u_h - u\|_{L^2(\Omega_N)} \leq$$
$$\left(2\sqrt{C_d\|\boldsymbol{\alpha}\|_2 t}\, C_\star(k, q, d, N) + (\bar{\bar{c}}_{k,0,q} + B_0(k, q, d)\kappa_0(k, q, N)^d)2^{-N/2}\right) 2^{-N(q+1/2)}|u_0|_{\mathcal{H}^{q+1}(\Omega)},$$

where C_d is a generic constant with dependence only on d, $C_\star(k, q, d, N) = \max_{s=0,1}\left(\bar{\bar{c}}_{k,s,q} + B_s(k, q, d)\kappa_s(k, q, N)^d\right)$. The constants $\bar{\bar{c}}_{k,s,q}$, $B_s(k, q, d)$, $\kappa_s(k, q, N)$ are defined in L^2 projection error estimates.

Convergence rate $O((\log h)^d h^{k+1/2})$.

Linear advection: sparse grid DG

We consider the following linear advection problem

$$\begin{cases} u_t + \sum_{m=1}^{d} u_{x_m} = 0, \quad \mathbf{x} \in [0,1]^d, \\ u(0,\mathbf{x}) = \sin\left(2\pi \sum_{m=1}^{d} x_m\right), \end{cases} \tag{4}$$

subject to periodic boundary conditions.

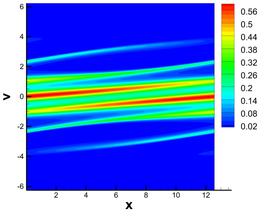
In the simulation, we compute the numerical solutions up to two periods in time, meaning that we let final time $T = 1$ for $d = 2$, $T = 2/3$ for $d = 3$, and $T = 0.5$ for $d = 4$.

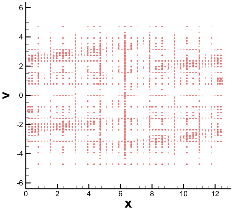
Table: L^2 errors and orders of accuracy at $T = 1$ when $d = 2$, $T = 2/3$ when $d = 3$, and $T = 0.5$ when $d = 4$. N is the number of mesh levels, h_N is the size of the smallest mesh in each direction, k is the polynomial order, d is the dimension. DOF denotes the degrees of freedom of the sparse approximation space $\hat{V}_N^k$. L^2 order is calculated with respect to h_N.

N	h_N	DOF	L^2 error	order	DOF	L^2 error	order	DOF	L^2 error	order
			$k=1,\ d=2$			$k=1,\ d=3$			$k=1,\ d=4$	
4	1/16	192	9.17E-02	–	832	3.72E-01	–	3072	4.99E-01	–
5	1/32	448	1.90E-02	2.27	2176	1.19E-01	1.64	8832	2.40E-01	1.06
6	1/64	1024	4.81E-03	1.98	5504	2.96E-02	2.01	24320	9.84E-02	1.28
7	1/128	2304	1.27E-03	1.92	13568	8.85E-03	1.74	64768	3.21E-02	1.62
			$k=2,\ d=2$			$k=2,\ d=3$			$k=2,\ d=4$	
4	1/16	432	2.13E-03	–	2808	1.10E-02	–	15552	2.80E-02	–
5	1/32	1008	4.39E-04	2.28	7344	1.79E-03	2.63	44712	5.82E-03	2.27
6	1/64	2304	4.45E-05	3.30	18576	3.97E-04	2.17	123120	1.37E-03	2.09
7	1/128	5184	7.68E-06	2.54	45792	5.14E-05	2.95	327888	2.58E-04	2.41
			$k=3,\ d=2$			$k=3,\ d=3$			$k=3,\ d=4$	
3	1/8	320	6.36E-04	–	2432	2.10E-03	–	16128	4.09E-03	–
4	1/16	768	8.93E-05	2.83	6656	2.37E-04	3.14	49152	6.06E-04	2.75
5	1/32	1792	4.07E-06	4.46	17408	2.49E-05	3.25	141312	6.85E-05	3.14
6	1/64	4096	3.47E-07	3.55	44032	1.83E-06	3.76	389120	7.19E-06	3.25
7	1/128	9216	1.97E-08	4.14	108544	2.03E-07	3.18	1036288	6.36E-07	3.50

Numerical examples: Kinetic equations

Vlasov-Poisson/Vlasov-Maxwell up to 4D. Example: Landau damping $t = 10$





Streaming Weibel instability

We consider 1D2V problem

$$f_t + \xi_2 f_{x_2} + (E_1 + \xi_2 B_3) f_{\xi_1} + (E_2 - \xi_1 B_3) f_{\xi_2} = 0 \ , \tag{5}$$

$$\frac{\partial B_3}{\partial t} = \frac{\partial E_1}{\partial x_2}, \quad \frac{\partial E_1}{\partial t} = \frac{\partial B_3}{\partial x_2} - j_1, \quad \frac{\partial E_2}{\partial t} = -j_2 \ , \tag{6}$$

The initial conditions are given by

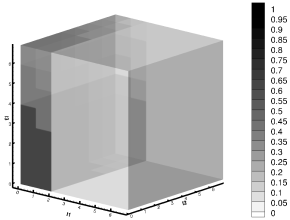
$$f(x_2, \xi_1, \xi_2, 0) = \frac{1}{\pi\beta} e^{-\xi_2^2/\beta} [\delta e^{-(\xi_1 - v_{0,1})^2/\beta} + (1-\delta) e^{-(\xi_1 + v_{0,2})^2/\beta}], \tag{7}$$

$$E_1(x_2, \xi_1, \xi_2, 0) = E_2(x_2, \xi_1, \xi_2, 0) = 0, \qquad B_3(x_2, \xi_1, \xi_2, 0) = b \sin(k_0 x_2) \ , \tag{8}$$

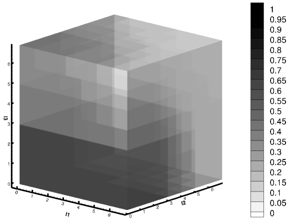
where $b = 0$ is an equilibrium state composed of counter-streaming beams propagating perpendicular to the direction of inhomogeneity, $\beta^{1/2}$ is the thermal velocity and δ is a parameter measuring the symmetry of the electron beams. $\beta = 0.01$, $b = 0.001$ Here, $\Omega_x = [0, L_y]$, where $L_y = 2\pi/k_0$, and we set $\Omega_\xi = [-1.2, 1.2]^2$. We consider the symmetric case

$$\underline{\text{choice 1}}: \ \delta \ = \ 0.5, v_{0,1} = v_{0,2} = 0.3, k_0 = 0.2 \tag{9}$$

Percent of active elements by ASG

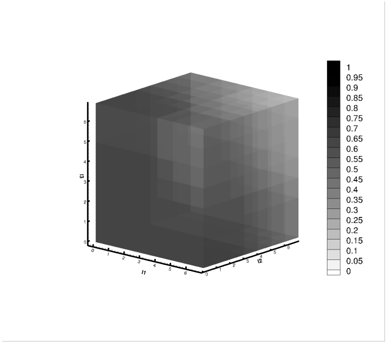


(c) $t = 0$. Active elements: 0.73%

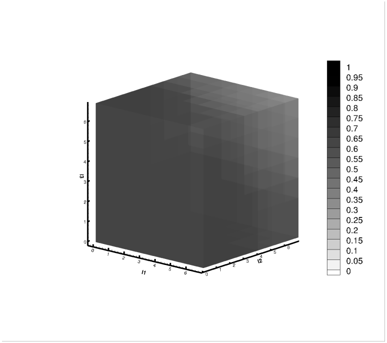


(d) $t = 55$. Active elements: 4.36%

Percent of active elements by ASG



(e) $t = 82$. Active elements: 26.55%



(f) $t = 100$. Active elements: 52.41%

Outline

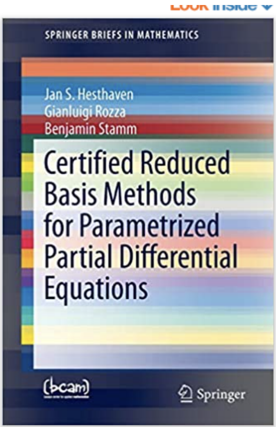
1 Introduction

2 Sparse grid DG method

3 Reduced basis method

4 Machine learning moment closure method

RBM



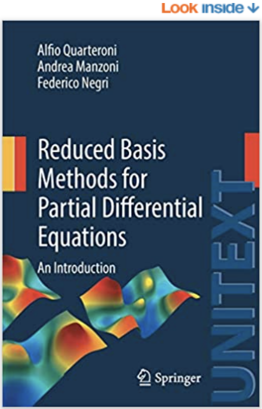
SPRINGER BRIEFS IN MATHEMATICS

Jan S. Hesthaven
Gianluigi Rozza
Benjamin Stamm

Certified Reduced Basis Methods for Parametrized Partial Differential Equations

bcam

Springer



Look inside

Alfio Quarteroni
Andrea Manzoni
Federico Negri

Reduced Basis Methods for Partial Differential Equations

An Introduction

Springer

UNITEXT

Reduced basis method (RBM): widely used in multi-query tasks, e.g. parametric PDEs. It employs a greedy algorithm that, via a rigorous error analysis, identifies the solution basis one-by-one.

Motivation

- Kinetic equation has inherently low rank structure.
  - Analytic model reduction. The classical fluid dynamics equations can be derived from Boltzmanns equations by ChapmanEnskog expansion. Grad's moment methods, and later many other work in this direction.
  - However, asumptions/ansatz are necessary for analytic type methods.
  - Recently, many computational based model reduction techniques are used, e.g. POD, tensor based methods.
- This work: develops a RBM that treats the velocity variable as the parameter, and can obtain intrinsic low rank structure when available. The outcome of the computation is a reduced order model to the kinetic simulations.

Model problem

A fundamental model in nuclear engineering, astrophysics, medical imaging: the radiative transfer equation (RTE)

$$\mathbf{\Omega} \cdot \nabla f = \sigma_s \langle f \rangle - \sigma_t f + G, \quad \forall\, \mathbf{x} \in \mathbf{X},\ \mathbf{\Omega} \in \mathbb{S}^{d-1} \tag{10}$$

where

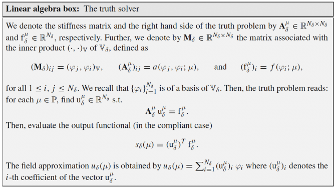
$$\langle f \rangle = \frac{1}{|\mathbb{S}^{d-1}|} \int_{\mathbb{S}^{d-1}} f d\mathbf{\Omega}. \tag{11}$$

Upon rescaling: $\frac{\sigma_s}{\varepsilon}$ and $\varepsilon \sigma_a$. When $\varepsilon \to 0$, $f(\mathbf{x}, \mathbf{\Omega}) \to \rho(\mathbf{x})$, and $\rho(\mathbf{x})$

solves the diffusion equation

$$\nabla \cdot (D \nabla \rho) = \sigma_a \rho. \tag{12}$$

Review: RBM, from Hesthaven, Rozza, Stamm



Linear algebra box: The truth solver

We denote the stiffness matrix and the right hand side of the truth problem by $\mathbf{A}_\delta^\mu \in \mathbb{R}^{N_\delta \times N_\delta}$ and $\mathrm{f}_\delta^\mu \in \mathbb{R}^{N_\delta}$, respectively. Further, we denote by $\mathbf{M}_\delta \in \mathbb{R}^{N_\delta \times N_\delta}$ the matrix associated with the inner product $(\cdot,\cdot)_{\mathbb{V}}$ of $\mathbb{V}_\delta$, defined as

$$(\mathbf{M}_\delta)_{ij} = (\varphi_j, \varphi_i)_{\mathbb{V}}, \qquad (\mathbf{A}_\delta^\mu)_{ij} = a(\varphi_j, \varphi_i; \mu), \qquad \text{and} \qquad (\mathrm{f}_\delta^\mu)_i = f(\varphi_i; \mu),$$

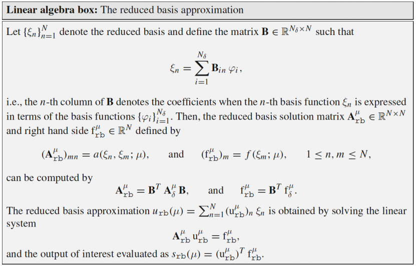
for all $1 \le i, j \le N_\delta$. We recall that $\{\varphi_i\}_{i=1}^{N_\delta}$ is of a basis of $\mathbb{V}_\delta$. Then, the truth problem reads: for each $\mu \in \mathbb{P}$, find $\mathrm{u}_\delta^\mu \in \mathbb{R}^{N_\delta}$ s.t.

$$\mathbf{A}_\delta^\mu \, \mathrm{u}_\delta^\mu = \mathrm{f}_\delta^\mu .$$

Then, evaluate the output functional (in the compliant case)

$$s_\delta(\mu) = (\mathrm{u}_\delta^\mu)^T \, \mathrm{f}_\delta^\mu .$$

The field approximation $u_\delta(\mu)$ is obtained by $u_\delta(\mu) = \sum_{i=1}^{N_\delta} (\mathrm{u}_\delta^\mu)_i \, \varphi_i$ where $(\mathrm{u}_\delta^\mu)_i$ denotes the i-th coefficient of the vector u_δ^μ.



Linear algebra box: The reduced basis approximation

Let $\{\xi_n\}_{n=1}^N$ denote the reduced basis and define the matrix $\mathbf{B} \in \mathbb{R}^{N_\delta \times N}$ such that

$$\xi_n = \sum_{i=1}^{N_\delta} \mathbf{B}_{in} \, \varphi_i ,$$

i.e., the n-th column of $\mathbf{B}$ denotes the coefficients when the n-th basis function ξ_n is expressed in terms of the basis functions $\{\varphi_i\}_{i=1}^{N_\delta}$. Then, the reduced basis solution matrix $\mathbf{A}_{\mathrm{rb}}^\mu \in \mathbb{R}^{N \times N}$ and right hand side $\mathrm{f}_{\mathrm{rb}}^\mu \in \mathbb{R}^N$ defined by

$$(\mathbf{A}_{\mathrm{rb}}^\mu)_{mn} = a(\xi_n, \xi_m; \mu), \qquad \text{and} \qquad (\mathrm{f}_{\mathrm{rb}}^\mu)_m = f(\xi_m; \mu), \qquad 1 \le n, m \le N,$$

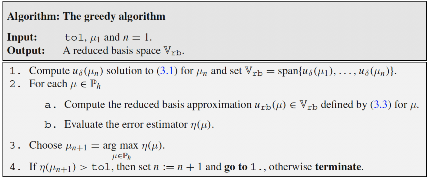
can be computed by

$$\mathbf{A}_{\mathrm{rb}}^\mu = \mathbf{B}^T \, \mathbf{A}_\delta^\mu \, \mathbf{B}, \qquad \text{and} \qquad \mathrm{f}_{\mathrm{rb}}^\mu = \mathbf{B}^T \, \mathrm{f}_\delta^\mu .$$

The reduced basis approximation $u_{\mathrm{rb}}(\mu) = \sum_{n=1}^N (\mathrm{u}_{\mathrm{rb}}^\mu)_n \, \xi_n$ is obtained by solving the linear system

$$\mathbf{A}_{\mathrm{rb}}^\mu \, \mathrm{u}_{\mathrm{rb}}^\mu = \mathrm{f}_{\mathrm{rb}}^\mu ,$$

and the output of interest evaluated as $s_{\mathrm{rb}}(\mu) = (\mathrm{u}_{\mathrm{rb}}^\mu)^T \, \mathrm{f}_{\mathrm{rb}}^\mu$.



Algorithm: The greedy algorithm

Input: `tol`, μ_1 and $n = 1$.
Output: A reduced basis space $\mathbb{V}_{\mathrm{rb}}$.

```
1. Compute $u_\delta(\mu_n)$ solution to (3.1) for $\mu_n$ and set $\mathbb{V}_{\mathrm{rb}} = \mathrm{span}\{u_\delta(\mu_1), \dots, u_\delta(\mu_n)\}$.
2. For each $\mu \in \mathbb{P}_h$
      a. Compute the reduced basis approximation $u_{\mathrm{rb}}(\mu) \in \mathbb{V}_{\mathrm{rb}}$ defined by (3.3) for $\mu$.
      b. Evaluate the error estimator $\eta(\mu)$.
3. Choose $\mu_{n+1} = \arg\max_{\mu \in \mathbb{P}_h} \eta(\mu)$.
4. If $\eta(\mu_{n+1}) > \mathtt{tol}$, then set $n := n + 1$ and go to 1., otherwise terminate.
```

Full order solver (FOM)

- Consider 1D, 2D simplified version

$$v\partial_x f = \frac{\sigma_s}{2}\int_{-1}^{1} f dv - \sigma_t f + G, \quad v \in [-1,1], \quad x \in \mathbf{X},$$
$$\cos(\theta)\partial_x f + \sin(\theta)\partial_y f = \sigma_s \langle f \rangle - \sigma_t f + G, \quad \theta \in [0, 2\pi], \ \mathbf{x} \in \mathbf{X}.$$

- Take upwind DG in space, S_N method in velocity, with synthetic accelerated source iteration.

Reduced order solver (ROM)

- Note that the velocity directions are coupled through the scattering term.
- We decouple the solutions for different angular samples by designing an iterative procedure where an approximation of the macroscopic density ρ is constructed from the RB snapshots and gradually refined as the RB space is built and RB solutions get more accurate.
- We use a least-squares density reconstruction strategy capable of integrating over an arbitrary set of selected angular samples.
- An L^1 residual-free error indicator is used for simplicity.
- The enrichment is done in a symmetric way.

Offline stage

Algorithm 2: RBM greedy algorithm for radiative transfer equation.

```
Input: iteration number tolerance $M_{\text{tol}}$, spectral ratio tolerance $r_{\text{tol}}$, training set
  $\mathcal{P} = \{\boldsymbol{\Omega}_1, \dots, \boldsymbol{\Omega}_{N_{\boldsymbol{\Omega}}}\}$, and an initial quadrature rule $\mathcal{P}_0 = \{\widehat{\boldsymbol{\Omega}}_k\}_{k=1}^{N_0}$.
Solve (globally coupled) (2.8) with a direct solver for $\boldsymbol{\Omega} \in \mathcal{P}_0$ to obtain $\mathbf{f}^0_{\widehat{\boldsymbol{\Omega}}_k}$ and $\boldsymbol{\rho}^0$.
Iteration:
Set $m = 0, r^m = 2r_{\text{tol}}$, $\mathbf{F}^0_{RB} = (\mathbf{f}^0_{\widehat{\boldsymbol{\Omega}}_1}, \dots, \mathbf{f}^0_{\widehat{\boldsymbol{\Omega}}_{N_0}})$ and $\mathcal{P}_{RB} = \{\widehat{\boldsymbol{\Omega}}_k, k = 1, \dots, N_0\}$.
while $m \le M_{\text{tol}}$ and $r^m > r_{\text{tol}}$ do
    Perform an SVD $\mathbf{F}^m_{RB} = \boldsymbol{\mathcal{U}}^m_{RB}\boldsymbol{\Lambda}^m_{RB}(\boldsymbol{\mathcal{V}}^m_{RB})^T$.
    Calculate the spectral ratio $r^m = \frac{\lambda^m_{\min}}{tr(\boldsymbol{\Lambda}^m_{RB})}$ where $\lambda^m_{\min}$ is the minimal singular value.
    for $i = 1 : N_{\boldsymbol{\Omega}}$ do
        if $\boldsymbol{\Omega}_i \notin \mathcal{P}_{RB}$, then
            Compute the RBM solution for $\boldsymbol{\Omega}_i$, $\mathbf{f}^m_{\boldsymbol{\Omega}_i} = \boldsymbol{\mathcal{U}}^m_{RB}\mathbf{c}^m_i$ with $\mathbf{c}^m_i$ solving

            $$(\boldsymbol{\mathcal{U}}^m_{RB})^T \mathbf{U}_{\boldsymbol{\Omega}_i} \boldsymbol{\mathcal{U}}^m_{RB}\mathbf{c}^m_i + (\boldsymbol{\mathcal{U}}^m_{RB})^T \boldsymbol{\Sigma}_t \boldsymbol{\mathcal{U}}^m_{RB}\mathbf{c}^m_i = (\boldsymbol{\mathcal{U}}^m_{RB})^T \boldsymbol{\Sigma}_s \boldsymbol{\rho}^m + (\boldsymbol{\mathcal{U}}^m_{RB})^T \mathbf{g}.$$

            Calculate the error indicator $\mathcal{E}^m_{\boldsymbol{\Omega}_i} = \sum_{k=1}^{N_m} |\widetilde{c}^m_k(\boldsymbol{\Omega}_i)|$ with $\widetilde{\mathbf{c}}^m_i = \boldsymbol{\mathcal{V}}^n_{RB}\,(\Lambda^n_{RB})^{-1}\,\mathbf{c}^m_i$.
        end if
    end for
    Set $i_{\text{new}} = \arg\max_i\{\mathcal{E}^m_{\boldsymbol{\Omega}_i}\}$, $m := m + 1$, and $\mathcal{P}_{RB} = \mathcal{P}_{RB} \bigcup \{\boldsymbol{\Omega}_{i_{\text{new}}}, \widetilde{\boldsymbol{\Omega}}_{i_{\text{new}}}\}$.
    Execute the SASI method to solve (2.8) and update $\mathbf{f}^m_{\boldsymbol{\Omega}_{j_k}}$ for $\boldsymbol{\Omega}_{j_k} \in \mathcal{P}_{RB}$ to assemble

    $$\mathbf{F}^m_{RB} = \left(\mathbf{f}^m_{\boldsymbol{\Omega}_{j_1}}, \dots, \mathbf{f}^m_{\boldsymbol{\Omega}_{j_{N_m}}}\right), \; N_m = 2m + N_0 \quad \text{and} \quad \mathcal{P}_{RB} = \{\boldsymbol{\Omega}_{j_k}\}_{k=1}^{N_m}.$$

    Reconstruct $\boldsymbol{\rho}^m$ based on $\mathbf{f}^m_{\boldsymbol{\Omega}}$ for $\boldsymbol{\Omega} \in \mathcal{P}_{RB}$ via (3.5) and (3.6) (or (3.7)).
end while
```

Online stage

During the online stage, we assume

$$\mathbf{f}_{\boldsymbol{\Omega}} \approx \mathcal{U}_{RB}\mathbf{c}_{RB}(\boldsymbol{\Omega}), \tag{13}$$

and compute $\mathbf{c}_{RB}(\boldsymbol{\Omega})$ by solving

$$\begin{aligned}(\boldsymbol{\mathcal{U}}_{RB})^T\mathbf{U}_{\boldsymbol{\Omega}_i}\boldsymbol{\mathcal{U}}_{RB}\mathbf{c}_{RB}(\boldsymbol{\Omega}) + (\boldsymbol{\mathcal{U}}_{RB})^T\boldsymbol{\Sigma}_t\boldsymbol{\mathcal{U}}_{RB}\mathbf{c}_{RB}(\boldsymbol{\Omega}) \\ = (\boldsymbol{\mathcal{U}}_{RB})^T\boldsymbol{\Sigma}_s\boldsymbol{\rho}_{RB} + (\boldsymbol{\mathcal{U}}_{RB})^T\mathbf{g}.\end{aligned}$$

This is the reduced order computational model we build using the RBM procedure.

Numerical result - 1D slab geometry

$N_0 = 2$. The training set consists of 24 Gauss-Legendre points.
Example 1 (scattering dominant):

$$\mathbf{X} = [0, 10],\ G = 0.01,\ \sigma_t = 100,\ \sigma_s = 100,\ f(0, v) = 0 \text{ with } v > 0,\ f(10, v) = 0 \text{ with } v \leq 0.$$

Example 2 (spatially varying scattering coefficient):

$$\begin{aligned} &\mathbf{X} = [0, 10],\ G = 0.01,\ \sigma_t = 100(1 + x),\ \sigma_s = 100(1 + x),\\ &f(0, v) = 0 \text{ with } v > 0,\ f(10, v) = 0 \text{ with } v \leq 0. \end{aligned}$$

Example 3 (two-material problem 1):

$$\begin{aligned} &\mathbf{X} = [0, 20],\ G = \begin{cases} 5,\ 0 < x < 10, \\ 0,\ 10 < x < 20, \end{cases} \quad \sigma_t = 100,\ \sigma_s = \begin{cases} 90,\ 0 < x < 10, \\ 100,\ 10 < x < 20, \end{cases}\\ &f(0, v) = 0 \text{ with } v > 0,\ f(20, v) = 0 \text{ with } v \leq 0. \end{aligned}$$

Example 4 (two-material problem 2):

$$\begin{aligned} &\mathbf{X} = [0, 11],\ G = 0,\ \sigma_t = \begin{cases} 100,\ 1 < x < 11, \\ 2,\ 0 < x < 1, \end{cases} \quad \sigma_s = \begin{cases} 100,\ 1 < x < 11, \\ 0,\ 0 < x < 1, \end{cases}\\ &f(0, v) = 5 \text{ with } v > 0,\ f(11, v) = 0 \text{ with } v \leq 0. \end{aligned}$$

Example 5 (transport dominant):

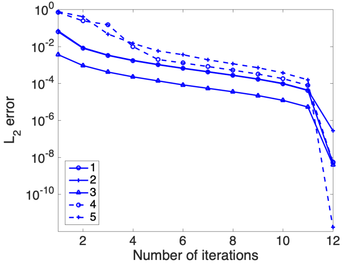
$$\mathbf{X} = [0, 10],\ G = 0.01,\ \sigma_t = 1.2,\ \sigma_s = 1,\ f(0, v) = 0 \text{ with } v > 0,\ f(10, v) = 0 \text{ with } v \leq 0.$$

Numerical result - 1D slab geometry

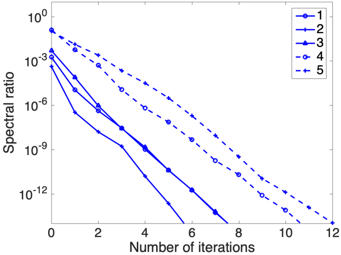
	RB dimension	$\mathcal{E}_f$	R_f	$\mathcal{E}_\rho$	R_ρ
Example 1	4	8.04e-3	9.26e-3%	8.00e-3	9.21e-3%
Example 2	4	8.51e-3	2.05e-3%	8.49e-3	2.04e-3%
Example 3	4	9.35e-4	5.13e-2%	8.25e-4	4.53e-2%
Example 4	8	9.99e-2	1.28e+1%	2.59e-3	2.54e-1%
Example 5	10	7.50e-2	5.39e-1%	4.37e-3	3.21e-2%

Table: Testing error for f and training error for ρ with $r_{\mathrm{tol}} = 10^{-4}$, 1D slab geometry.

Numerical result - 1D slab geometry

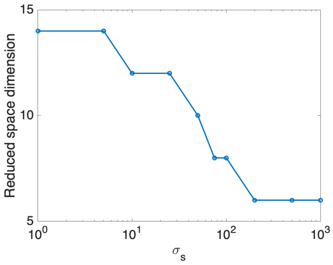


(g) Error history of f

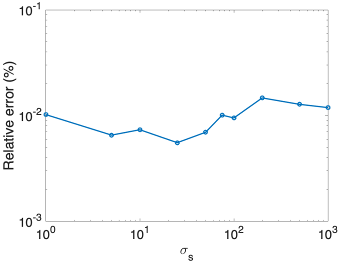


(h) History of spectral ratio

Numerical result - 1D slab geometry



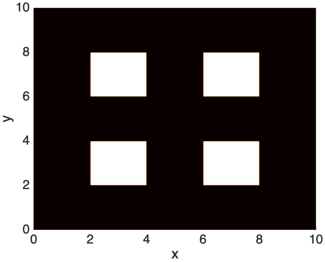
(i) Dimension of reduced space for $r_{\mathrm{tol}} = 10^{-8}$

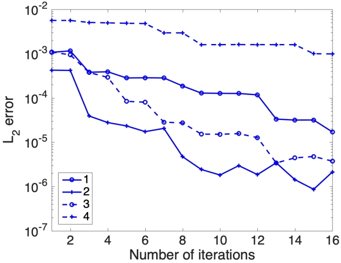


(j) Relative testing error of f

Numerical result

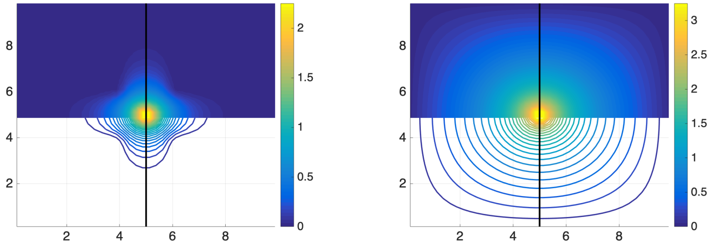
Example #1 is the classical checkerboard problem where $\sigma_s = 1$ and $\sigma_a = 1$ in the white region while $\sigma_s = 100$ and $\sigma_a = 0$ in the black region. Examples #2, 3, 4 have $\sigma_s = 100, 10,$ and 1, modeling the scattering, intermediate, and transport regimes, respectively, without the absorption effect.





Numerical result

Examples 1 and 2, obtained by 8 and 4 bases. Left: FOM, Right: ROM.



Outline

1. Introduction
2. Sparse grid DG method
3. Reduced basis method
4. Machine learning moment closure method

Machine learning moment closure

- ML has found use in many areas of scientific computing. In particular, ML has been used in the following works to assist moment closure for kinetic equations. Han, Ma, Ma and E (2019), Bois, Franck, Navoret and Vigon (2020), Ma, Zhu, Xu and Wang (2020), Wang, Xu, Zhu, Ma and Lei (2020), Maulik, Garland, Burby, Tang and Balaprakas (2020), Porteous, Laiu and Hauck (2021), Schotthöfer, Xiao, Frank and Hauck (2021)
- We begin by considering time dependent RTE $f = f(x, v, t)$, $x \in \mathbb{R}$ and $v \in [-1, 1]$

$$\partial_t f + v\partial_x f = \sigma_s \left(\frac{1}{2} \int_{-1}^{1} f dv - f \right) - \sigma_a f. \tag{14}$$

Model equation and moment method

Define the k-th order moment by

$$m_k(x,t) = \frac{1}{2}\int_{-1}^{1} f(x,v,t)P_k(v)dv, \quad k \geq 0.$$

with $P_k(v)$ the k-th Legendre polynomial.
Moment equations

$$\begin{aligned} \partial_t m_0 + \partial_x m_1 &= -\sigma_a m_0, \\ \partial_t m_1 + \frac{1}{3}\partial_x m_0 + \frac{2}{3}\partial_x m_2 &= -(\sigma_s + \sigma_a) m_1, \\ &\cdots \\ \partial_t m_N + \frac{N}{2N+1}\partial_x m_{N-1} + \frac{N+1}{2N+1}\partial_x m_{N+1} &= -(\sigma_s + \sigma_a) m_N. \end{aligned} \tag{15}$$

Traditional closures: P_N closure, M_N closure, etc

This work: we use the representability power of neural network to obtain the closure relations from the training data.

ML closure

Question: how to design the NN, so that the reduced model has good accuracy and stability properties?

- We developed gradient-based closure for RTE motivated by free streaming limit, directly learn $\partial_x m_{N+1}$ instead of m_{N+1}

$$\partial_x m_{N+1} = \sum_{k=0}^{N} \mathcal{N}_k(m_0, m_1, \ldots, m_N)\partial_x m_k \tag{16}$$

ML Closure

- We write the closure model into a system of first-order PDEs:

$$\partial_t \boldsymbol{m} + A(\boldsymbol{m})\partial_x \boldsymbol{m} = \boldsymbol{S}(\boldsymbol{m}) \tag{17}$$

  with $\boldsymbol{m} = (m_0, m_1, \cdots, m_N)^T$.

- Hyperbolicity
  - definition: The system (17) is hyperbolic if $A(\boldsymbol{m})$ is real diagonalizable
  - Hyperbolicity is crucial for long-time stability of the model!
  - difficulty: $A(\boldsymbol{m})$ depend on neural network, generally NOT real diagonalizable

Symmetrizer-based hyperbolic closure

- Gradient-based ML moment closure model:

$$\partial_t \boldsymbol{m} + A(\boldsymbol{m})\partial_x \boldsymbol{m} = \boldsymbol{S}(\boldsymbol{m}) \tag{18}$$

with $\boldsymbol{m} = (m_0, m_1, \cdots, m_N)$ and

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 & \dots & 0 \\ 0 & \frac{2}{5} & 0 & \frac{3}{5} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{N-1}{2N-1} & 0 & \frac{N}{2N-1} \\ a_0 & a_1 & \dots & a_{N-2} & a_{N-1} & a_N \end{pmatrix}$$

with a_j related to neural networks:

$$a_j = \begin{cases} \dfrac{N+1}{2N+1}\mathcal{N}_j, & j \neq N-1 \\ \dfrac{N}{2N+1} + \dfrac{N+1}{2N+1}\mathcal{N}_j, & j = N-1 \end{cases}$$

- **Key idea**: seek a symmetric positive definite matrix A_0 (also called a symmetrizer) such that $A_0 A$ is symmetric $\Rightarrow$ symmetrizable hyperbolic

Theorem (symmetrizable hyperbolic)

Consider matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$ *with* $N \geq 3$ *and* $a_i = 0$ *for* $i = 0, 1, \cdots, N-4$. *If the coefficients* a_i *for* $i = N-3, N-2, N-1, N$ *satisfy the following constraints:*

$$a_{N-3} > -\frac{(N-1)(N-2)}{N(2N-3)}, \quad a_{N-1} > \frac{g(a_{N-3}, a_{N-2}, a_N; N)}{(N-2)(a_{N-3}(2N-3)N + (N-1)(N-2))^2} \tag{19}$$

where $g = g(a_{N-3}, a_{N-2}, a_N; N)$ *is a function given by*

$$\begin{aligned} g = {} & a_{N-3}^3 (N-1) N^2 (3-2N)^2 + a_{N-2}(2N-1)(N-2)^3 (a_{N-2}N - a_N(N-1)) \\ & + a_{N-3}(N-2)^2 (a_N(4N^2 - 8N + 3)(a_{N-2}N - a_N(N-1)) + (N-1)^3) + 2a_{N-3}^2 (N-1)^2 N(2N-3)(N-2), \end{aligned}$$

then there exist a SPD matrix $A_0 = diag(D, B) \in \mathbb{R}^{(N+1)\times(N+1)}$ *such that* $A_0 A$ *is symmetric. Here,* $D = diag(1, 3, 5, \cdots, 2N-5) \in \mathbb{R}^{(N-2)\times(N-2)}$ *and* $B \in \mathbb{R}^{3\times 3}$ *is a SPD matrix.*

NN Training ($k = 3$)

- NN maps $(m_0, m_1, \cdots, m_N)$ onto $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4)$ with

  $$\mathcal{N}_N = \mathcal{M}_4, \quad \mathcal{N}_{N-2} = \mathcal{M}_2,$$

  $$\mathcal{N}_{N-3} = \sigma(\mathcal{M}_3) - \frac{(N-2)(N-1)(2N+1)}{N(2N-3)(N+1)}$$

  $$\mathcal{N}_{N-1} = \sigma(\mathcal{M}_1) - \frac{N}{N+1} + \frac{h(\mathcal{N}_{N-3}, \mathcal{N}_{N-2}, \mathcal{N}_N; N)}{(N-2)(\mathcal{N}_{N-3}(N+1)(2N-3)N + (N-2}$$

  Here $\sigma : \mathbb{R} \to \mathbb{R}$ is a positive function, i.e., $\sigma(x) > 0$ for any $x \in \mathbb{R}$.
- We use fully connected NN with the number of layers=6 and the number of nodes=256 with the hyperbolic tangent activation function.

Training data generation

We numerically solve the RTE to generate training data:

- unit interval $[0, 1]$ in the physical domain with periodic boundary conditions
- initial conditions, truncated Fourier series:

$$f_0(x, v) = a_0 + \sum_{k=1}^{k_{\max}} a_k \sin(2k\pi x + \phi_k), \tag{20}$$

$k_{\max} = 10$, $a_k \sim U(-\frac{1}{k}, \frac{1}{k})$ for $k \geq 1$, $\phi_k \sim U[0, 2\pi]$, $a_0 = c + \sum_{k=1}^{k_{\max}} \frac{1}{k}$ with $c \sim U[0, 1]$

- $\sigma_s \sim U[0.1, 100]$ and $\sigma_a \sim U[0, 10]$, constants over the domain
- take 100 different initial data
- space time DG method[1]: $N_x = 512$, $\Delta t = 8\Delta x$, $T = 1$

Numerical results

- Two-material problem. (Different scattering coef in different parts of the domain) $N = 6$ at $t = 0.5$ and $t = 1$. Gray part: optically thin regime; other part: intermediate regime.

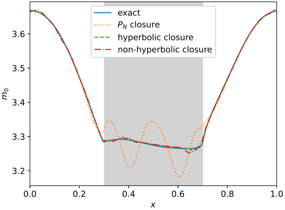


Figure: m_0 at $t = 0.5$

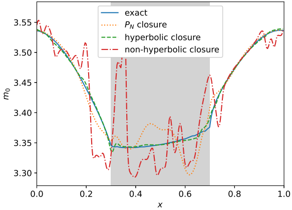


Figure: m_0 at $t = 1$

Comparison:

- P_N closure: stable, but not accurate in optically thin regime
- Non-hyperbolic ML closure: accurate for short time, but blow up for long time
- Hyperbolic ML closure: stable and accurate for long time

Errors in different regimes

- Relative L^2 error vs. scattering coefficient σ_s
  - hyperbolic ML closure is stable and more accurate than P_N closure

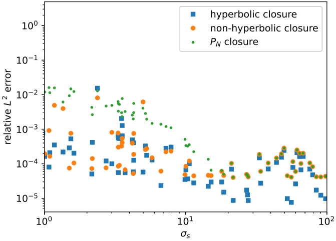


Figure: error of m_0 at $t = 0.5$

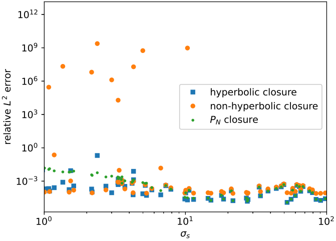


Figure: error of m_0 at $t = 1$

Conclusions and future work

We developed three approaches to speed up kinetic simulations.

- Sparse grid DG method. Adaptive and accurate. Cost reduction depends on the solution profile. In future, we will incorporate well known techniques (mm decomposition) to treat the collisional case. Software package `https://github.com/JuntaoHuang/adaptive-multiresolution-DG`
- Reduced basis method. Effective reduction in diffusive/fluid regime. This approach computes a reduced order model for the kinetic equation. In future, we will generalize this to other kinetic models, e.g. time dependent, BGK.
- ML moment closure methods. Using a data-driven approach, ML offers an interesting approach for providing reduced order models for wide regimes. We are working on extensions to 2D models.

The END!

Thank You!