Conservative Adaptive Rank Integrators for Nonlinear Kinetic Models

Jingmei Qiu

Department of Mathematical Sciences
University of Delaware, USA.

Joint work with W. Guo (TTU),
J. Nakao (Swarthmore), L. Einkemmer (Innsbruck)

IPAM, UCLA, Feb. 7th, 2024.

Outline

- Scope: Leverage efficiency speedup from tensor network to traditional mesh-based solvers in the field of computational fluid dynamics and multi-scale modeling and simulations of PDEs.
- Kinetic equations: six-dimensional time dependent PDEs.
- Adaptive rank representation of high D PDE solutions
  - 1D1V Vlasov model: explicit integration
  - LoMaC: **Lo**cally **Ma**croscopic **C**onservative projection
  - 0D2V Fokker-Planck model: implicit integration
  - High dimensional problems: Hierarchical Tucker tensor
- Outlook for future directions.

Boltzmann equation

Plasma dynamics may be described by the Boltzmann-Maxwell system

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \boldsymbol{B}) \cdot \nabla_{\mathbf{v}} f_s = C(f). \tag{1}$$

with the electromagnetic fields $\mathbf{E}$ and $\boldsymbol{B}$ satisfying the Maxwell equations,

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \mathbf{E}, \quad \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \boldsymbol{B} - \mathbf{J},$$

$$\nabla \cdot \mathbf{E} = \rho, \quad \nabla \cdot \boldsymbol{B} = 0,$$

with the charge density $\rho = \sum_s q_s \int f_s d\mathbf{v}$ the current density $\mathbf{J} = \sum_s q_s \int f_s \mathbf{v} d\mathbf{v}$.

- $f_s(t, v, x)$ is the distribution function of particle of species s.
- 6D + time nonlinear dynamical systems.

Curse of dimensionality of traditional mesh-based methods

Introduced by Bellman: to achieve a prescribed accuracy ϵ, the complexity of an algorithm scales like $O(\epsilon^{-d/\alpha})$ for a d-dimensional problem,

- Sparse grid methods.
- Reduced order modeling.
- Low-rank tensor approximation of kinetic solutions.
  - Dynamical low rank approximation (DLR): derive a set of differential equations for the low-rank factors by projecting the update onto the tangent space
  - Step and Truncate(SAT): low rank truncation of a full rank high order solver.

DLR : spatial discret. $\rightarrow$ low rank projection $\rightarrow$ temporal discret.

SAT : spatial discret. $\rightarrow$ temporal discret. $\rightarrow$ adaptive rank projection.

Explicit low rank integration of 1D1V Vlasov solutions

Vlasov-Poisson 1D1V system

In the zero magnetic limit, the Vlasov-Maxwell system is reduced to the Vlasov-Poisson system for electrons,

$$f_t + v \cdot \nabla_x f + E(x,t) \cdot \nabla_v f = 0, \quad (2)$$

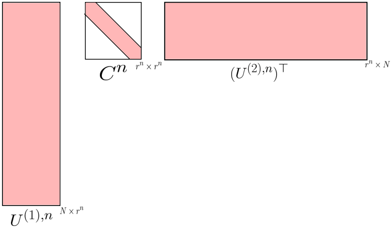
$$E(x,t) = -\nabla_x \phi(x,t), \quad -\triangle_x \phi(x,t) = \rho(x,t) = \int f dv - 1. \quad (3)$$

A low rank idea for 1D1V Vlasov-Poisson system

- At the continuous level: Schmidt decomposition.

$$f(x, v, t^{(n)}) = \sum_{j=1}^{r^{(n)}} \left(C_j^{(n)} \quad U_j^{(1),(n)}(x) U_j^{(2),(n)}(v) \right), \qquad (4)$$

- At the discrete level: SVD of A with $A_{ij} = f(x_i, v_j)$.



Frames in x and v-directions as orthogonal global basis for function approximations. Storage: $\mathcal{O}(Nr)$. Low rank if $r^n \ll N$.

At the continuous level

1D1V VP system

$$f_t + v \cdot f_x + E \cdot f_v = 0.$$

Keep the phase space continuous, a low rank representation of function f at t^n

$$f^{(n)}(x,v) = \sum_j c_j^{(n)} U_j^{(1),(n)}(x) U_j^{(2),(n)}(v)$$

A forward Euler discretization of the VP system gives

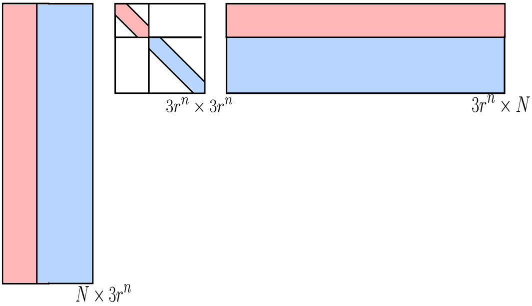
$$f^{(n+1)}(x,v) = \sum_{j=1}^{r^{(n)}} C_j^{(n)} \left[U_j^{(1),(n)}(x) U_j^{(2),(n)}(v) - \Delta t \right.$$
$$\left. \left(\frac{\partial}{\partial x} U_j^{(1),(n)}(x) \otimes (v U_j^{(2),(n)}(v)) + (E^n(x) U_j^{(1),(n)}) \otimes \frac{\partial}{\partial v} U_j^{(2),(n)}(v) \right) \right],$$

Dynamically and adaptively update basis and solutions

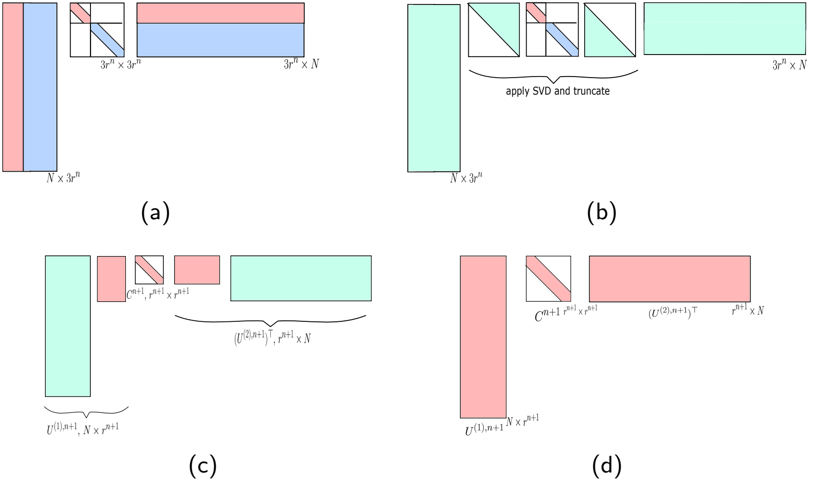
1 Add basis. Take the forward Euler method for example,

$$f^{(n+1),*} = \sum_{j=1}^{r^{(n)}} C_j^{(n)} \left[\left(U_j^{(1),(n)} \otimes U_j^{(2),(n)} \right) \right.$$
$$\left. -\Delta t \left(D_x U_j^{(1),(n)} \otimes v \star U_j^{(2),(n)} + E^n \star U_j^{(1),(n)} \otimes D_v U_j^{(2),(n)} \right) \right],$$

Here D_x and D_v represent a differentiation matrix, e.g. from spectral method or high order finite difference methods with upwind principle.



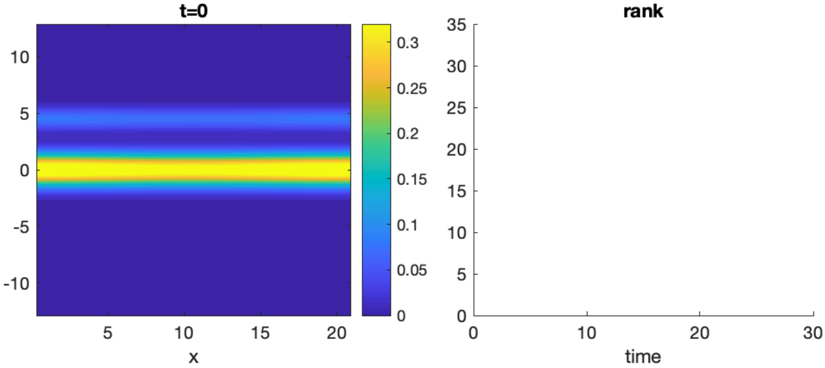
2 SVD Truncation.



(a)→(b): perform Gram-Schmidt to obtain orthonormal basis;
(b)→(c): perform SVD on and truncate small singular values;
(c)→(d) update low rank form of solution at t^{n+1}.

Bump-on-tail instability

Truncation threshold $\varepsilon = 10^{-4}$. $N_x \times N_v = 64 \times 128$.



Extension to high D: hierarchical Tucker decomposition

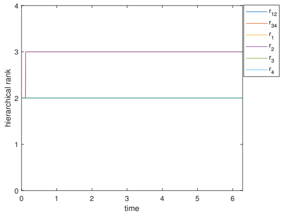
$$u_t + \sum_{m=1}^{d} u_{x_i} = 0, \quad \mathbf{x} \in [-\pi, \pi]^d$$

with periodic boundary conditions, a smooth initial condition

$$u(\mathbf{x}, t = 0) = \exp\left(-2(x_1^2 + x_2^2)\right) \sin(x_3 + x_4), \tag{5}$$

$d = 4$ and $T = 2\pi$. For the low rank algorithm with $N = 128$ and $\varepsilon = 10^{-6}$, the CPU time is 4.2s, 7.8s, 14.2, 30.1s, 65.3s, respectively.

N	L^2 error	order
16	2.56E-02	
32	5.76E-03	2.15
64	1.41E-03	2.04
128	3.52E-04	2.00
256	8.09E-05	2.12



Low rank approach for 1D1V Vlasov system

- Built upon traditional methods: high order spatial and temporal accuracy.
- Explicit time stepping: evolve and truncate with low rank and rank adaptive
- Extension to high D: via hierarchical Tucker representation of high D functions
- Overcoming the CoD: CPU time only doubled with mesh refinement, but not 2^{d+1}.

Remaining issues:

- Loss of conservation with SVD truncation: conservative in the "add basis step"; but conservation is destroyed in the truncation step.
- Implicit time stepping

Locally **Ma**croscopic **C**onservative (LoMaC) Projection

Macroscopic conservation

The VP system satisfies the macroscopic moment equations

$$\begin{aligned}\rho_t + \nabla_{\mathbf{x}} \cdot \mathbf{J} &= 0\\ \partial_t \mathbf{J} + \nabla_{\mathbf{x}} \cdot \sigma &= -\rho \mathbf{E}\\ \partial_t e + \nabla_{\mathbf{x}} \cdot \mathbf{Q} &= \mathbf{E} \cdot (\mathbf{E}_t - \mathbf{J})\,.\end{aligned}$$

- particle density: $\rho(\mathbf{x}, t) = \int_{\Omega_{\boldsymbol{v}}} f(\mathbf{x}, \boldsymbol{v}, t) d\boldsymbol{v}$
- current density: $\mathbf{J}(\mathbf{x}, t) = \int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v} f(\mathbf{x}, \boldsymbol{v}, t) d\boldsymbol{v}$
- kinetic energy density: $\kappa(\mathbf{x}, t) = \frac{1}{2} \int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v}^2 f(\mathbf{x}, \boldsymbol{v}, t) d\boldsymbol{v}$
- energy density: $e(\mathbf{x}, t) = \kappa(\mathbf{x}, t) + \frac{1}{2}\mathbf{E}^2$
- fluxes:
  - $\sigma(\mathbf{x}, t) = \int_{\Omega_{\boldsymbol{v}}} (\boldsymbol{v} \otimes \boldsymbol{v}) f(\mathbf{x}, \boldsymbol{v}, t) d\boldsymbol{v}$,
  - $\mathbf{Q}(\mathbf{x}, t) = \frac{1}{2} \int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v}\boldsymbol{v}^2 f(\mathbf{x}, \boldsymbol{v}, t) d\boldsymbol{v}$.

- If a discretization of the VP system can reduce to a consistent and conservative discretization for the macroscopic system, then it is locally conservative. Local conservation leads to global conservation.
- The discretization used in the adding bases step is locally mass and momentum conservative. Conservation in energy requires implicit energy conserving time integrators *.
- The mass, momentum and energy conservation is lost in the SVD truncation step.
- We propose a mass, momentum and energy conservative SVD truncation.

*Cheng, Christlieb, Zhong 2014

General idea: orthogonal decomposition of $\mathbf{f}^{(n+1)}$ in the add basis step

$$\mathbf{f}^{(n+1)} = \mathbf{f}_1 + \mathbf{f}_2, \quad \mathbf{f}^{(n+1)} \in \mathbf{R}^{N_x \times N_v}.$$

- $\mathbf{f}_1$ represents the orthogonal projection of $\mathbf{f}^{(n+1)}$ onto the subspace $W = span(\mathbf{1}_v, \boldsymbol{v}, \boldsymbol{v}^2)$.

$$\mathbf{f}_1 = P_W(\mathbf{f}^{(n+1)})$$

- The orthogonal remainder $\mathbf{f}_2$ contains zero mass, momentum and kinetic energy,

$$\mathbf{f}_2 = (I - P_W)(\mathbf{f}^{(n+1)})$$

The compressed low rank solution at $t^{(n+1)}$ is

$$\mathbf{f}^{(n+1)} = \mathbf{f}_1 + \mathcal{T}(\mathbf{f}_2),$$

where $\mathcal{T}$ is the low rank truncation of $\mathbf{f}_2$ with zero mass, momentum and kinetic energy.

Weighted inner product space

- Inner product space $l_w^2(\Omega_v)$:
$$< \mathbf{f}, \mathbf{g} >_w \doteq \sum \mathbf{f}_j \mathbf{g}_j \mathbf{w}_j w_{q,j},$$
with weight function $\mathbf{w}$. It can be viewed as a discrete analog of $< f, g >= \int f(v) g(v) w(v) dv$.
- One may choose $\mathbf{w}_j = \exp(-v_j^2/2)$ to ensure
$$W \subset l_w^2(\Omega_v).$$
Such weight function will also ensure $\mathbf{f}_1$ have proper decay as $v \to \infty$.
- Subspace: $W = span(\mathbf{1}_v, \boldsymbol{v}, \boldsymbol{v}^2)$ for conservation of mass, momentum and energy.

Construction of $\mathbf{f}_1$

- Introduce the rescaled $\tilde{\mathbf{f}} = \mathbf{f} \star \frac{1}{\mathbf{w}}$,
- perform the orthogonal projection of $P_W(\tilde{\mathbf{f}})$, s.t.

$$< P_W(\tilde{\mathbf{f}}), \mathbf{g} >_w = < \tilde{\mathbf{f}}, \mathbf{g} >_w, \quad \forall \mathbf{g} \in W.$$

- Then let

$$\mathbf{f}_1 = \mathbf{w} \star P_W(\tilde{\mathbf{f}})$$

Proposition.

$$\mathbf{f}_1 = \mathbf{w} \star (c_1 \otimes \mathbf{1}_v + c_2 \otimes \mathbf{v} + c_3 \otimes (\boldsymbol{v}^2 - c)), \tag{6}$$

with $c = \frac{\langle \mathbf{1}_v, \boldsymbol{v}^2 \rangle_w}{\langle \mathbf{1}_v, \mathbf{1}_v \rangle_w}$, $c_1 = \frac{\rho}{\|\mathbf{1}_v\|_w^2}$, $c_2 = \frac{j}{\|\boldsymbol{v}\|_w^2}$, and $c_3 = \frac{2\kappa - \rho c}{\|\boldsymbol{v}^2 - c\|_w^2}$. ρ, j and $\kappa \in \mathbf{R}^{N_x}$ are macroscopic charge, current and kinetic densities. f_1 preserves the mass, momentum and kinetic energy of f.

Weighted truncation of $\mathbf{f}_2$

Let $\mathbf{f}_2 = \mathbf{f}^{(n+1)} - \mathbf{f}_1$.

- $\mathbf{f}_2$ is orthogonal to W in $l_w^2(\Omega_v)$.
- A weighted SVD truncation for $\mathcal{T}_w(\mathbf{f}_2)$

$$\mathbf{f}_2 \overset{rescaling}{\rightarrow} \tilde{\mathbf{f}}_2 = \mathbf{f}_2 \star \frac{1}{\sqrt{\mathbf{w}}} \overset{truncation}{\rightarrow} \mathcal{T}(\tilde{\mathbf{f}}_2) \overset{rescaling}{\rightarrow} \sqrt{\mathbf{w}} \star \mathcal{T}(\tilde{\mathbf{f}}_2)$$

- Update the solution $\mathbf{f}^{(n+1)} = \mathbf{f}_1 + \mathcal{T}_w(\mathbf{f}_2)$.

Proposition. $\mathcal{T}_w(\mathbf{f}_2)$ has zero density, zero current density, and zero kinetic energy. Hence, $\rho^{(n+1)}$, $J^{(n+1)}$, $\kappa^{(n+1)}$ are preserved for $f^{(n+1)}$ after the truncation. Furthermore, the method is locally conservative with mass and momentum, as the full rank high order method.

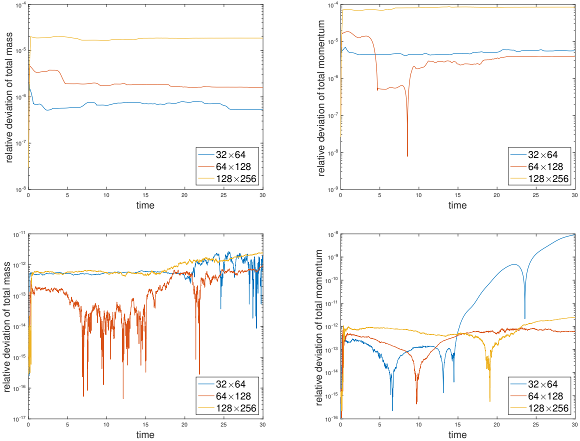
Algorithm 1 The conservative truncation for the 1D1V VP system.

- Input: the pre-compressed low-rank solution at time $t^{(n+1)}$: $\mathbf{f}^{*,(n+1)} = \sum_{i=1}^{R} C_i^* \ \boldsymbol{U}_i^{*,(1)} \otimes \boldsymbol{U}_i^{*,(2)}$.
- Output: the compressed low-rank solution $\mathbf{f}^{n+1}$ with conservation on charge, current and kinetic energy density.

1. Compute $\boldsymbol{\rho}^{*,(n+1)}$, $\mathbf{J}^{*,(n+1)}$, $\boldsymbol{\kappa}^{*,(n+1)} \in \mathbf{R}^{N_x}$ of $\mathbf{f}^{*,(n+1)}$ in a low-rank fashion.
2. Compute $\mathbf{f}_1 = \mathbf{w} \star (c_1 \otimes \mathbf{1}_v + c_2 \otimes \mathbf{v} + c_3 \otimes (\boldsymbol{v}^2 - c))$ with c_1, c_2, c_3 and c from $\boldsymbol{\rho}^{*,(n+1)}$, $\mathbf{J}^{*,(n+1)}$, $\boldsymbol{\kappa}^{*,(n+1)}$.
3. Perform the weighted SVD truncation of $\mathbf{f}_2 = \mathbf{f}^{*,(n+1)} - \mathbf{f}_1$: $\mathcal{T}_w(\mathbf{f}_2)$.
4. Update the compressed low-rank solution by

$$\mathbf{f}^{(n+1)} = \mathbf{f}_1 + \mathcal{T}_w(\mathbf{f}_2).$$

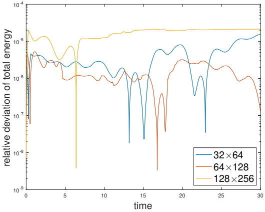
Bump-on-tail instability



Non-conservative (upper panels) and conservative (lower panels) method.
$\varepsilon = 10^{-4}$.

However, energy is not conserved

- Conservation of energy is of paramount importance to avoid unphysical plasma self-heating or cooling.
- Existing full-rank methods with energy conservation require implicit symplectic time integrators.
- Our work: an explicit LoMaC method by working with macroscopic equations alongside with the kinetic Vlasov model.



$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0$$

Kinetic flux →

← Macroscopic densities

$$\begin{aligned} \rho_t + \nabla_{\mathbf{x}} \cdot \mathbf{J} &= 0 \\ \partial_t \mathbf{J} + \nabla_{\mathbf{x}} \cdot \sigma &= \rho \mathbf{E} \\ \partial_t e + \nabla_{\mathbf{x}} \cdot \mathbf{Q} &= 0. \end{aligned}$$

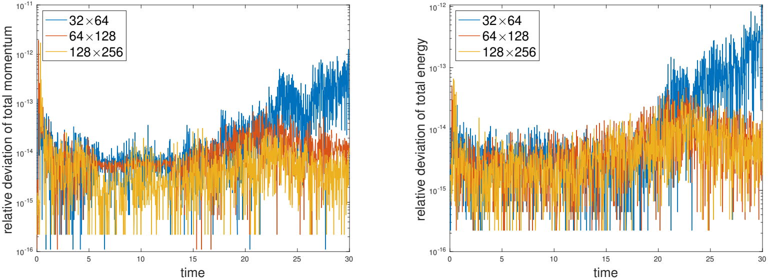
1. Employ a conservative method for the macroscopic model with kinetic flux vector splitting (KFVS) computed from $\mathbf{f}$:

$$\nabla_x \cdot \begin{pmatrix} \mathbf{J} \\ \sigma \\ \mathbf{Q} \end{pmatrix} = \nabla_x \cdot \left[\int f v \begin{pmatrix} 1 \\ v \\ \frac{1}{2} v^2 \end{pmatrix} dv \right]$$

2. Replace the macroscopic charge, current and kinetic energy densities in $\mathbf{f}_1$ by the ones from the macroscopic model.

$$\mathbf{f}_1 = \mathbf{w} \star (c_1 \otimes \mathbf{1}_v + c_2 \otimes \mathbf{v} + c_3 \otimes (\boldsymbol{v}^2 - c)),$$

with c, c_1, c_2, and c_3 depending on macroscopic charge, current and kinetic energy density $(\rho, \mathbf{J}, e)$.



Bump-on-tail instabilities. The time evolution of relative deviation of total momentum and total energy from the energy conserving method. Mesh $N_x \times N_v = 128 \times 256$. $\varepsilon = 10^{-4}$.

Reduced Augmentation Implicit Low rank (RAIL) schemes for Fokker-Planck models

The Fokker-Planck model and matrix differential equations

- The Fokker-Planck model

$$\frac{\partial f_\alpha}{\partial t} = \sum_{\beta}^{N_s} \gamma_{\alpha\beta}\{D_{\alpha\beta}\frac{\partial^2 f_\alpha}{\partial v^2} + \frac{\partial}{\partial v}[(v-u_\beta)f_\alpha]\}$$

- **F** on velocity discretizations on tensor product of grids: matrix differential equation

$$\frac{\partial \mathbf{F}}{\partial t} = \mathrm{Ex}(\mathbf{F}) + \mathrm{Im}(\mathbf{F}) \tag{7}$$

  - $\mathrm{Ex}(\mathbf{F})$: explicit treatment as in step-and-truncate (SAT)
  - $\mathrm{Im}(\mathbf{F})$: e.g. $D_1\mathbf{F} + \mathbf{F}D_2^T$

- RAIL as implicit low rank integrators

$$\frac{\partial \mathbf{F}}{\partial t} = D_1\mathbf{F} + \mathbf{F}D_2^T \tag{8}$$

evolving the matrix **F** in the low rank format $\mathbf{F} = \mathbf{V}^x\mathbf{S}(\mathbf{V}^y)^T$.

DIRK time discretization

Seek $\mathbf{F}^{\cdot} = \mathbf{V}^{x,\cdot}\mathbf{S}^{\cdot}(\mathbf{V}^{y,\cdot})^T$

- At the k-th stage, with $t^{(n,k)} = t^{(n)} + c_k \Delta t$, $k = 1, 2, ..., s$.

$$\mathbf{F}^{(n,k)} = \mathbf{F}^{(n)} + \Delta t \sum_{\ell=1}^{k} a_{k\ell} \mathbf{Y}_\ell, \tag{9}$$

with $\mathbf{Y}_k := D_1 \mathbf{V}^x \mathbf{S} (\mathbf{V}^y)^T + \mathbf{V}^x \mathbf{S} (D_2 \mathbf{V}^y)^T$.

- Final RK update of the solution:

$$\mathbf{F}^{(n+1)} = \mathbf{F}^{(n)} + \Delta t \sum_{k=1}^{s} b_k \mathbf{Y}_k, \tag{10}$$

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \dots & 0 \\ c_2 & a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

Backward Euler: K-L steps

$$\mathbf{V}^{x,(n+1)}\mathbf{S}^{(n+1)}(\mathbf{V}^{y,(n+1)})^T - \Delta t\Big(D_1\mathbf{V}^{x,(n+1)}\mathbf{S}^{(n+1)}(\mathbf{V}^{y,(n+1)})^T + \mathbf{V}^{x,(n+1)}\mathbf{S}^{(n+1)}(D_2\mathbf{V}^{y,(n+1)})^T\Big) = \mathbf{V}^{x,(n)}\mathbf{S}^{n}(\mathbf{V}^{y,(n}$$

- Unknowns: $\mathbf{V}^{x,(n+1)}$, $\mathbf{S}^{(n+1)}$, $(\mathbf{V}^{y,(n+1)})^T$
- Our strategy of divide and conquer: predict $\mathbf{V}^{\cdot,(n+1)}$ via K-L steps, and update $\mathbf{S}^{(n+1)}$ via S-step.
    - K step of size $N \times r$: frozen and project around $\mathbf{V}_\star^{y,(n+1)} := \mathbf{V}^{y,(n)}$ (with $\mathcal{O}(\Delta t)$ error) gives the Sylvester equation

$$\mathbf{K}^{(n+1)} - \Delta t D_1 \mathbf{K}^{(n+1)} - \Delta t \mathbf{K}^{(n+1)}(D_2\mathbf{V}^{y,(n)})^T\mathbf{V}^{y,(n)} = \mathbf{V}^{x,(n)}\mathbf{S}^{(n)}. \tag{11}$$

    with $\mathbf{K}^{(n+1)} = \mathbf{V}^{x,(n+1)}\mathbf{S}^{(n+1)}(\mathbf{V}^{y,(n+1)})^T\mathbf{V}_\star^{y,(n+1)} \in \mathbb{R}^{N\times r}$.
    - Perform QR factorization to $\mathbf{K}^{(n+1)}$ to obtain orthonormal basis $\mathbf{V}_\ddagger^{x,(n+1)}$:

$$\mathbf{K}^{(n+1)} = \mathbf{V}_\ddagger^{x,(n+1)}\mathbf{R}$$

    - Similarly for other dimensions:

$$\mathbf{V}^{\cdot,(n)} \in \mathbb{R}^{N\times r^{(n)}} \xrightarrow[K \text{ and } L \text{ equations}]{\text{Use } \mathbf{V}_\star^{\cdot,(n+1)} \text{ for projection}} \mathbf{V}_\ddagger^{\cdot,(n+1)} \in \mathbb{R}^{N\times r_\ddagger^{(n+1)}}$$

Backward Euler: S step

- Reduced augmentation: strike a balance between augmentation(enriching the subspaces as much as possible) and reduced (truncation on redundancy)

$$\mathbf{V}_{\text{aug}}^{x,(n+1)} := \left[\ \mathbf{V}_{\ddagger}^{x,(n+1)} \mid \mathbf{V}^{x,(n)} \ \right] \in \mathbb{R}^{N\times(r_{\ddagger}^{(n+1)}+r^{(n)})}. \tag{12}$$

$$\mathbf{V}_{\text{aug}}^{x,(n+1)} = \underbrace{\mathbf{Q}_{\text{aug}}^{x}\mathbf{R}_{\text{aug}}^{x}}_{\text{reduced QR}} = \mathbf{Q}_{\text{aug}}^{x}\underbrace{\mathbf{U}_{\text{aug}}^{x}\mathbf{\Sigma}_{\text{aug}}^{x}(\mathbf{V}_{\text{aug}}^{x})^{T}}_{\text{SVD}}.$$

$$\mathbf{V}^{x,(n+1)} := \mathbf{Q}_{\text{aug}}^{x}\mathbf{U}_{\text{aug}}^{x}(\,:\,,1:R),$$
$$\mathbf{V}^{y,(n+1)} := \mathbf{Q}_{\text{aug}}^{y}\mathbf{U}_{\text{aug}}^{y}(\,:\,,1:R).$$

- S-step:

$$\left(\mathbf{I} - \Delta t(\mathbf{V}^{x,(n+1)})^{T}D_1\mathbf{V}^{x,(n+1)}\right)\mathbf{S}^{(n+1)} - \mathbf{S}^{(n+1)}\left(\Delta t(D_2\mathbf{V}^{y,(n+1)})^{T}\mathbf{V}^{y,(n+1)}\right)$$
$$= (\mathbf{V}^{x,(n+1)})^{T}\mathbf{V}^{x,(n)}\mathbf{S}^{(n)}(\mathbf{V}^{y,(n)})^{T}\mathbf{V}^{y,(n+1)}.$$

- Finally, the updated solution $\mathbf{V}^{x,(n+1)}\mathbf{S}^{(n+1)}(\mathbf{V}^{y,(n+1)})^{T}$ is truncated using an SVD-type procedure.

Extensions to general DIRK and IMEX schemes

- BE:
  - formal analysis to first order local truncation error (LTE)
  - unconditional stability
- General DIRK: per RK stage
  - Prediction of basis: from BE and previous RK stages via reduced augmentation procedure
  - K-L steps in updating basis
  - S step in updating coefficients
- General IMEX methods for a mixture of implicit-explicit treatment

RAIL IMEX integrators

Convection-diffusion:

$$\begin{cases} u_t + \nabla \cdot (\mathbf{a}(\mathbf{x},t)u) = \nabla \cdot (\mathbf{D} \cdot \nabla u) + \phi(\mathbf{x},t), & \mathbf{x} \in \Omega, \quad t > 0, \\ u(\mathbf{x}, t=0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \end{cases}$$

Low rank assumption:

$$\mathbf{U}(t) = \mathbf{V}^x(t)\mathbf{S}(t)(\mathbf{V}^y(t))^T$$

Spatial discretization:

$$\frac{d}{dt}\mathbf{U} = \mathrm{Ex}(\mathbf{U}) + \mathrm{Im}(\mathbf{U}) + \mathbf{\Phi}.$$

IMEX RK methods:

Table 2: Implicit Scheme

0	0	0	0	0	0
c_1	0	a_{11}	0	$\dots$	0
c_2	0	a_{21}	a_{22}	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	0	a_{s1}	a_{s2}	$\dots$	a_{ss}
	0	b_1	b_2	$\dots$	b_s

Table 3: Explicit Scheme

0	0	0	0	$\dots$	0
c_1	$\hat{a}_{21}$	0	0	$\dots$	0
c_2	$\hat{a}_{31}$	$\hat{a}_{32}$	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	$\hat{a}_{\sigma 1}$	$\hat{a}_{\sigma 2}$	$\hat{a}_{\sigma 3}$	$\dots$	0
	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\dots$	$\hat{b}_\sigma$

Rotational problem with diffusion: IMEX RAIL integrators

$$u_t - yu_x + xu_y = d(u_{xx} + u_{yy}) + \phi, \qquad x, y \in (-2\pi, 2\pi)$$

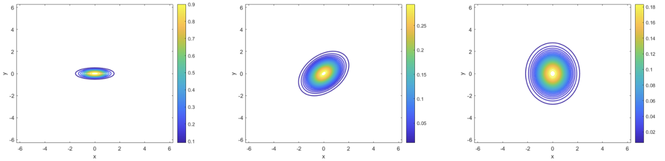
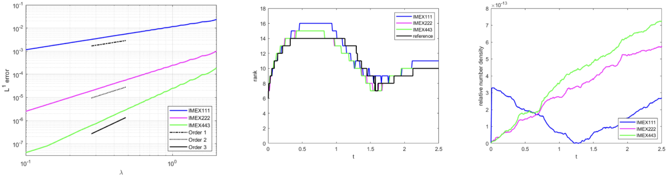


Figure 5: Various snapshots of the numerical solution to equation (3.3) with initial condition $\exp(-(x^2+9y^2))$. Mesh size $N = 200$, tolerance $\epsilon = 1.0E-08$, time-stepping size $\Delta t = 0.15\Delta x$, initial rank $r^0 = 20$, using IMEX(4,4,3). Times: 0, $\pi/4$, $\pi/2$.

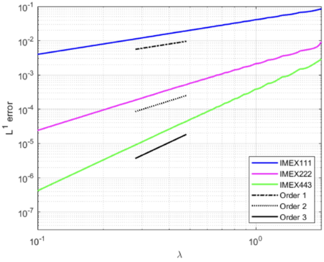


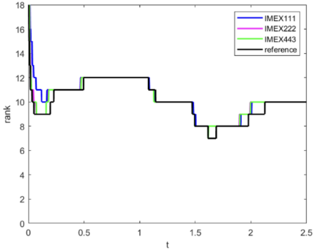
Swirling deformation problem with diffusion

$$u_t - \left(\cos^2(x/2)\sin(y)f(t)u\right)_x + \left(\sin(x)\cos^2(y/2)f(t)u\right)_y = u_{xx} + u_{yy}, \qquad x, y \in (-\pi, \pi) \qquad ($$

where we set $f(t) = \cos(\pi t/T_f)\pi$. The initial condition is the smooth (with C^5 smoothness) cosine bell

$$u(x,y,t=0) = \begin{cases} r_0^b \cos^6\left(\frac{r^b(x,y)\pi}{2r_0^b}\right), & \text{if } r^b(x,y) < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \qquad ($$





0D2V Lenard-Bernstein-Fokker-Planck equation

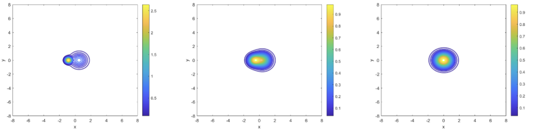
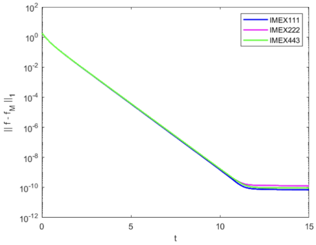


Figure 9: Various snapshots of the numerical solution to equation (3.6) with initial condition $f_{M1}(v_x, v_y) + f_{M2}(v_x, v_y)$. Mesh size $N = 300$, tolerance $\epsilon = 1.0E-06$, time-stepping size $\Delta t = 0.15\Delta x$, initial rank $r^0 = 30$, using IMEX(4,4,3). Times: 0, 0.25, 1.



Tensor approach for high dimensional PDEs

Low-rank tensor approximation of functions

- Huge amount of literature. Textbook by [Hachbusch 2012].
- The canonical polyadic (CP) tensor decomposition represents a multivariate function as a sum of rank-one separable functions [Hitchcock 1927], [Carroll, Chang 1970], [Kolda, Bader 2009], ...).
- Tensor networks for sparse data tensor decomposition: the hierarchical Tucker (HT) format ([Hackbusch, Kühn 2009], [Grasedyck 2010], ...) and the tensor train (TT) format ([Oseledets 2011], ...). Storage complexity: linear scaling with the dimension.

Tensor

Algebraically, a tensor is a multilinear array (this is a very simplified setting).

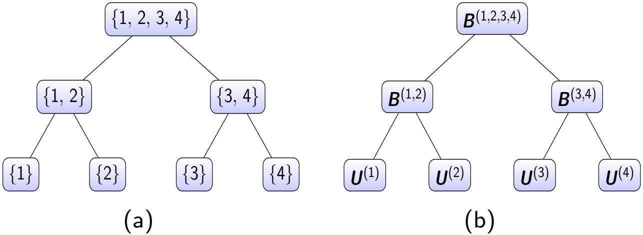
$$\mathbf{a}[\mathbf{i}] = \mathbf{a}[i_1, i_2, \ldots, i_d] \in \mathbb{R},$$

$i_j \in I_j = \{1, 2, \ldots, n_j\}$, $\mathbf{I} = \otimes_{j=1}^{n} I_j$. $\mathbf{a} \in \mathbb{R}^{\mathbf{I}}$ is called a d-th order tensor. Example

- $d = 1$, vector
- $d = 2$, matrix
- grid function based on the direct discretization of a multivariate function $f(\mathbf{x})$ on product grids, $\mathbf{a}[i_1, i_2, \ldots, i_d] = f(i_1 h, i_2 h, \cdots, i_d h)$. h: the mesh size.
  - Storage cost: $\prod_{j=1}^{d} n_j$, suffers the curse of dimensionality.
  - Low rank decomposition to represent or approximate tensors of high order in a data sparse format.

Hierarchical Tucker (HT) format

To represent a four-order tensor in HT format, we define a dimension tree.



- The storage complexity of the HT format scales like $O(ndr + dr^3)$, where $r = \max_\alpha\{r_\alpha \in \mathbf{r}_{HT}\}$, avoiding the exponential scaling on d.
  - leaf node $\{j\}$, the basis $u^{(j)}$ of U_j is explicitly stored
  - at a non-leaf node α, the *transfer* tensor $\boldsymbol{B}^{(\alpha)}$ is stored
- a tensor already in the HT format can be truncated to smaller HT rank with a quasi-optimal error bound, and the cost only scales like $O(dnr^2 + dr^4)$, i.e., linear scaling with d.

2D2V Vlasov-Poisson solution

Hierarchical Tucker tensor representation of order-four tensor $f(x_1, x_2, v_1, v_2)$.

$$f^{(n)} = \sum_{i_{12}=1}^{r_{12}} \sum_{i_{34}=1}^{r_{34}} \boldsymbol{B}^{(1,2,3,4),(n)}_{i_{12},i_{34},1} \boldsymbol{U}^{(1,2),(n)}_{i_{12}} \otimes \boldsymbol{U}^{(3,4),(n)}_{i_{34}}, \qquad (13)$$

with

$$\boldsymbol{U}^{(1,2),(n)}_{i_{12}} = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \boldsymbol{B}^{(1,2),(n)}_{i_1,i_2,i_{12}} \boldsymbol{U}^{(1),(n)}_{i_1} \otimes \boldsymbol{U}^{(2),(n)}_{i_2}, \quad i_{12} = 1, \ldots, r_{12}, \qquad (14)$$

and

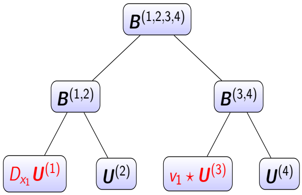
$$\boldsymbol{U}^{(3,4),(n)}_{i_{34}} = \sum_{i_3=1}^{r_3} \sum_{i_4=1}^{r_4} \boldsymbol{B}^{(3,4),(n)}_{i_3,i_4,i_{34}} \boldsymbol{U}^{(3),(n)}_{i_3} \otimes \boldsymbol{U}^{(4),(n)}_{i_4}, \quad i_{34} = 1, \ldots, r_{34}. \qquad (15)$$

2D2V Vlasov-Poisson equation

$$f_t + v_1 f_{x_1} + v_2 f_{x_2} + E_1 f_{v_1} + E_2 f_{v_2} = 0, \tag{16}$$

As with the 1D1V case, in the adding basis step, we need to discretize the derivatives term-by-term. For example,

$$v_1 f_{x_1} \approx (v_1 \otimes D_{x_1}) f^n.$$

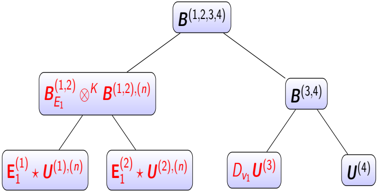


Treatment for $v_2 f_{x_2}$ is similar.

2D2V Vlasov-Poisson equation, cond.

For $E_1 f_{v_1}$, assume $E_1^{(n)}$ is in the HT format.

$$E_1 f_{v_1} \approx (E_1 \otimes D_{v_1}) f^{(n)}$$



$\boldsymbol{E} = (E_1, E_2)$ is solved by a low-rank conjugate gradient method from Poisson's equation †.

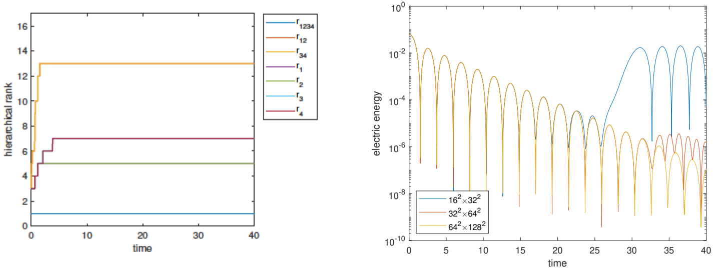
† Grasedyck and Löbbert 2018

2D2V Vlasov-Poisson equation: low rank truncation

- Matlab toolbox `htucker` [‡].
- Similar to SVD truncation of matrices, the HOSVD truncation for hierarchical Tucker tensor consist of (1) orthogonalize the frames and transfer tensors, (2) compute the reduced Gramians and (3) the associated eigen-decompostion with truncation.
- Cost: $\mathcal{O}\left(dNr^2 + (d-2)r^4\right)$.
- Mass, momentum and energy conservative truncation are being developed (rather involved).

[‡] Kressner, Tobler 2012

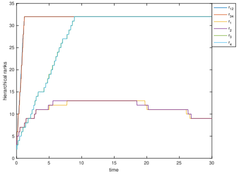
Nonlinear Vlasov-Poisson: weak Landau damping



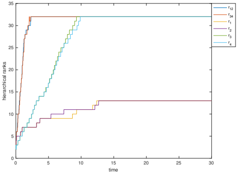
(c) (d)

Weak Landau damping 2D2V. Truncation threshold in SVD is $\epsilon = 1.e - 6$. $N_x \times N_v = 16^2 \times 32^2$, $32^2 \times 64^2$ and $64^2 \times 128^2$. CPU time is 76s, 117s, and 265s, respectively. The time evolution of the rank for $64^2 \times 128^2$ (left), the electric energy (right)

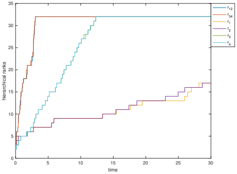
Nonlinear Vlasov-Poisson: strong Landau damping



(a)



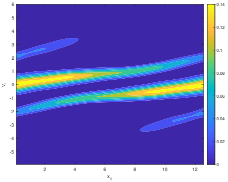
(b)



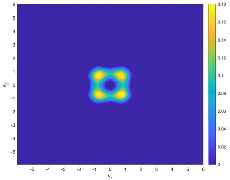
(c)

Strong Landau damping 2D2V. Hierarchical ranks. $\varepsilon = 10^{-3}$. $r_{max} = 32$. (a) $N_x^2 \times N_v^2 = 32^2 \times 64^2$. (b) $N_x^2 \times N_v^2 = 64^2 \times 128^2$. (c) $N_x^2 \times N_v^2 = 128^2 \times 256^2$. The CPU time is $438.9s$, $1278.7s$, and $2606.4s$.

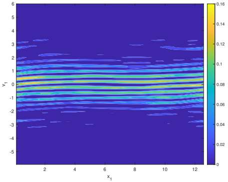
Nonlinear Vlasov-Poisson: strong Landau damping



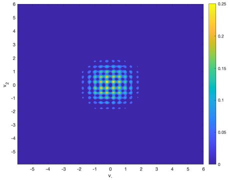
(a)



(b)



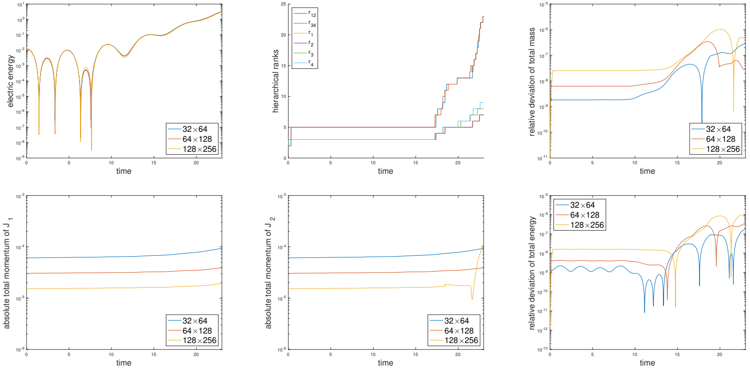
(c)



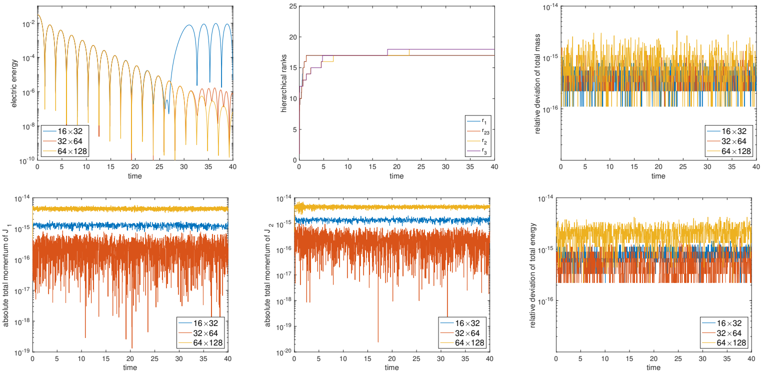
(d)

Strong Landau damping 2D2V. $\varepsilon = 10^{-3}$. $r_{max} = 32$. $N_x^2 \times N_v^2 = 128^2 \times 256^2$.
(a) 2D cut at $(x_2, v_2) = (2\pi, 0)$ $t = 5$. (b) 2D cut at $(x_1, x_2) = (2\pi, 2\pi)$ $t = 5$.
(c) 2D cut at $(x_2, v_2) = (2\pi, 0)$ $t = 30$. (d) $(x_1, x_2) = (2\pi, 2\pi)$ $t = 30$..

Non-conservative low rank method for 2D2V two stream instability. $\varepsilon = 10^{-5}$.



LoMaC low rank method for 2D2V weak Landau damping. $\varepsilon = 10^{-5}$.



Summary

- A low rank explicit tensor approach for the high dimensional Vlasov equation
- RAIL algorithm as an implicit low rank approach for stiff collisional terms
- A LoMaC projection for consistency with nonlinear macroscopic conservation laws

Ongoing work

- A Krylov subspaces approach for implicit tensor integration.
- Multi-scale low rank approach for various kinetic models.

References:

- Low rank Vlasov equation: Guo & Q., JCP 2022
- LoMaC projection: Guo Q., SISC, arXiv: 2207.00518
- Reduced augmentation of implicit low rank method (RAIL), Nakao, Q. & Einkemmer, arXiv: 2311.15143

Questions? Thank you!