Summary of Dynamical Low-Rank Approximation for Kinetics

Jack Coughlin

July 7, 2022

Basic example of the Dynamical Low-Rank Method

Dynamical low-rank approximation method is a method that can greatly reduce the number of degrees of freedom required to represent a solution to a kinetic equation. It tackles the curse of dimensionality by projecting a scalar problem posed on $2d$ dimensions into a pair of r-component problems posed on d dimensions, each.

We'll demonstrate using the collisionless Vlasov equation:

$$\partial_t f + v \cdot \nabla_x f + E \cdot \nabla_v f = 0 \tag{1}$$

Suppose the solution is of the form

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = \sum_{ij}^{r} X_i(\boldsymbol{x}, t) S_{ij}(t) V_j(\boldsymbol{v}, t). \tag{2}$$

The integer r is the rank of the solution. Sums over i and j always run from 1 to r. Our solver will keep track of the basis functions X_i, V_j and the matrix which combines them, S_{ij}. The angle bracket notations used below are defined thus:

$$\langle \cdot \rangle_x = \int_{\Omega_x} \cdot \, \mathrm{d}\boldsymbol{x}, \qquad \langle \cdot \rangle_v = \int_{\Omega_v} \cdot \, \mathrm{d}\boldsymbol{v}, \qquad \langle \cdot \rangle_{xv} = \int_{\Omega_x} \int_{\Omega_v} \cdot \, \mathrm{d}\boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \tag{3}$$

Given: low rank factors X_i^n, V_j^n, S_{ij}^n, and the electric field E^n at time n.

We compute the new unknowns $X_i^{n+1}, V_j^{n+1}, S_{ij}^{n+1}$, and E^{n+1} at time t^{n+1} as follows.

- **E step**:
  - Compute the charge density as follows:

$$\rho(\boldsymbol{x}) = \int_{\Omega_v} f \, \mathrm{d}\boldsymbol{v} = \sum_{ij} X_i^n(\boldsymbol{x}) S_{ij}^n \int_{\Omega_v} V_j^n(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v} \tag{4}$$

  - Solve Poisson's equation for E
- **K step**:
  - Define $K_j^n = \sum_i X_i^n S_{ij}^n$
  - Solve for K^{n+1} using a Forward Euler step:

$$K_j^{n+1} = K_j^n - \Delta t \sum_l (\nabla_x K_l^n) \cdot \left\langle \boldsymbol{v} V_j^n V_l^n \right\rangle_v - \Delta t \sum_l K_l^n E \cdot \left\langle V_j^n \nabla_v V_l^n \right\rangle_v \tag{5}$$

    Cost: $\mathcal{O}(r^2 N_v + r^2 N_x)$.

– Extract a new basis X^{n+1} and intermediate matrix S' using a QR decomposition:

$$\sum_i X_i^{n+1}(\boldsymbol{x}) S'_{ij} = K_j^{n+1}(\boldsymbol{x}). \tag{6}$$

Cost: $\mathcal{O}(rN_x)$.

- **S step**:
  - Solve for the next intermediate state S'' using a Forward Euler step:

$$S''_{ij} = S'_{ij} + \Delta t \sum_{kl} S'_{kl} \left\langle X_i^{n+1} \nabla_x X_k^{n+1} \right\rangle_x \cdot \left\langle \boldsymbol{v} V_j^n V_l^n \right\rangle_v + \Delta t \sum_{kl} S'_{kl} \left\langle X_i^{n+1} E X_k^{n+1} \right\rangle_x \cdot \left\langle V_j^n \nabla_v V_l^n \right\rangle_v \tag{7}$$

This is a system of r^2 ODEs. Note that this step runs "backwards", which can be a problem for diffusive operators.

Cost: $\mathcal{O}(r^2 N_x + r^2 N_v)$ to compute the matrices and $\mathcal{O}(r^4)$ to advance the ODE.

- **L step**:
  - Define $L_i^n = \sum_j S''_{ij} V_j^n$.
  - Solve for L_i^{n+1} with a Forward Euler step:

$$L_i^{n+1} = L_i^n - \Delta t \sum_k L_k^n \left\langle X_i \nabla_x X_k \right\rangle_x - \Delta t \sum_k (\nabla_v L_k^n) \left\langle X_i^{n+1} E X_k^{n+1} \right\rangle_x \tag{8}$$

Cost: $\mathcal{O}(r^2 N_x)$ to compute the matrices and $\mathcal{O}(r^2 N_v)$ to advance the PDE.

  - Extract a new basis V_j^{n+1} and the new matrix S_{ij}^{n+1} using a QR decomposition:

$$\sum_j S_{ij}^{n+1} V_j^{n+1}(\boldsymbol{v}) = L_i^{n+1}(\boldsymbol{v}) \tag{9}$$