The Vlasov-Fokker-Planck Equation with High Dimensional Parametric Forcing Term

Shi Jin* Yuhua Zhu† Enrique Zuazua‡§¶

Abstract

We consider the Vlasov-Fokker-Planck equation with random electric field where the random field is parametrized by countably many infinite random variables due to uncertainty. At the theoretical level, with suitable assumption on the anisotropy of the randomness, adopting the technique employed in elliptic PDEs [5], we prove the best N approximation in the random space breaks the dimension curse and the convergence rate is faster than the Monte Carlo method. For the numerical method, based on the adaptive sparse polynomial interpolation (ASPI) method introduced in [2], we develop a residual based adaptive sparse polynomial interpolation (RASPI) method which is more efficient for multi-scale linear kinetic equation, when using numerical schemes that are time dependent and implicit. Numerical experiments show that the numerical error of the RASPI decays faster than the Monte-Carlo method and is also dimension independent.

Key words. Dimension curse, Kinetic equation, Fokker-Planck operator, Hypocoercivity, Adaptive sparse polynomial interpolation, Residual based greedy algorithm

1 Introduction

We consider the Vlasov-Fokker-Planck (VFP) equation with a random electric field due to uncertainty. Typically uncertainty is modeled by a stochastic field, which by the Karhunen-Lòeve approximation is parametrized by countably infinite random variables [29, 30]. One of the difficulties in the development of numerical methods for such problems is the possible curse of dimension. Sampling methods, such as the Monte-Carlo methods, are often used, which are dimension independent. However, these methods suffer from the low convergence rate, as the numerical errors are of $O(N^{-\frac{1}{2}})$ with N sample points. In this paper we seek a more efficient numerical method based on best N approximation and greedy algorithms, originally developed for elliptic equations, for uncertain VFP equation in which the electric field depends on high

*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSEC and SHL-MAC, Shanghai Jiao Tong University, Shanghai 200240, China. (shijin-m@sjtu.edu.cn)

†Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53706, USA. (yzhu232@wisc.edu)

‡Chair in Applied Analysis, Alexander von Humboldt-Professorship, Department of Mathematics Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany.

§Chair of Computational Mathematics, Fundación Deusto, Av. de las Universidades 24, 48007 Bilbao, Basque Country, Spain.

¶Departamento de Matemáticas, Universidad Autónoma de Madrid, 28049 Madrid, Spain.

dimensional random variables in order to achieve a numerical convergence rate faster than the Monte-Carlo methods.

There are two separated parts in this paper. In Section 3, we reviewe the best N approximation, and prove the convergence rate of it when applied to the VPF equation, under suitable assumption on random field. We point out in Section 4, the best N approximation is a non-linear approximation hard to implement in practice. Therefore, for the numerical method, we develop a residual based adaptive sparse polynomial interpolation (RASPI) method, which is shown in Section 5 by different numerical experiments to be efficient in practice and indeed converges faster than the Monte Carlo method.

Our theoretical results in Section 3 are based on the results in a series of paper [3, 6, 7, 5]. For uniformly distributed random variables, we seek approximate solutions in a finite dimensional space spanned by the Legendre polynomial basis, that is $u(\mathbf{z}) \approx u_\Lambda(\mathbf{z}) = \sum_{\nu\in\Lambda} \omega_\nu L_\nu(\mathbf{z})$ for $\#(\Lambda) = N$, where $\#(\Lambda)$ is the number of elements in Λ. The best N approximation is to truncate the basis according to the N largest coefficient ω_ν, so that the mean square error, which is represented by $\sum_{\nu\notin\Lambda} |\omega_\nu|^2$, can be as small as possible. It is summarized in [5] that holomorphy and anisotropy of the solution in the random space implies that the best N approximation can break the curse of dimension. Furthermore it converges faster than the usual Monte Carlo Method. It has been successfully applied to elliptic equations, including parametric PDEs, control problems, inverse problems, etc [11, 12, 23, 36, 35, 37, 8, 28]. However, since this result requires analyticity of the solution in the random space, it hasn't been widely used in other PDEs, such as kinetic and related equations. Thanks to recent studies on the regularity of the solution to most of the kinetic equations using hypocoercivity of the kinetic operators, which give rise to high order regularity in the random space for kinetic equations with uncertainties, if the (random) initial data and (random) coefficients have such regularities, [18, 27, 22, 39, 42, 19], we first extend such reegularity study for the VFP equation with *infinite* dimensional random variables, which gives the *first* error estimate for uncertain kinetic equation that is *independent* of the dimension of the random variables. While for moderately high dimensionality, sparse grids were used for uncertain kinetic equations [14], here we are interested in much high dimensions in the the random space.

Based on the theoretical results, we develop a numerical method in Section 4, which is then applied in Section 5 to several examples to verify that it indeed successfully breaks the curse of dimension. The numerical method we develop is a residual based adaptive sparse polynomial interpolation (RASPI) method, which combines the idea from the adaptive sparse polynomial interpolation (ASPI) method and the residual based greedy search. The ASPI method, introduced in [2] in line of [33, 32], is a non-intrusive method that computes a polynomial approximation by interpolation of the solution map at N well chosen points. In particular it could be applied in the case when the exact model is not known, and only the numerical solver is given. However, in order to find the "well chosen" points, one needs to calculate the solution at number of sample points much bigger than N, which can be very costly when the PDE is time dependent and the numerical scheme is in implicit form. Actually for most multi-scale kinetic equations, one indeed needs to use implicit schemes due to the presence of small parameter or numerical stiffness [17, 20, 10, 21]. This means one needs to invert the approximate kinetic operator, for each mesh point, which is a large matrix in each time step. As will be shown later, the inversion of a large matrix can be avoided by computing the residual of the PDE instead. The idea of using

residual of a PDE has been used in greedy algorithm [34] for parametric PDEs and recently also applied to parametric control problems [26, 13], but mainly in low dimensions. Although such a method may end up using even less basis compared to polynomial approximations, their offline stage is potentially very costly, especially in high dimensions. The RASPI method combines the advantages of both methods, so that one can save much computational cost by calculating the residual of the PDE instead of the numerical solution of the PDE. At the same time, the offline stage is still efficient in high dimensions.

We would like to point out that although all numerical experiments in Section 5 verify the fast decay rate independent of the dimension, a rigorous proof that the ASPI and the RASPI can achieve the convergence rate we obtained in Section 3 by the best N approximation is still an open question.

Here is the structure of the paper. In Section 2, we introduce the VFP equation with random electric field. In Section 3, we prove the best N approximation converges to the solution with an error of $O\left(N^{-s}\right)$, for $s > \frac{1}{2}$, based on the result in [5]. We then in Section 4 give an improved numerical method RASPI based on the ASPI introduced in [2], and provide explicitly the computational cost it saved compared to the ASPI. Numerical experiments are conducted in Section 5 to show the convergence rates for various electric fields. The paper is concluded in Section 6.

Gallery of Notations 1.1. Define $\Omega = [0, l] \times \mathbb{R}$ to be the domain for x, v. The following norms are defined in Ω:

- $\|h\|^2 = \int_\Omega h^2 \, dxdv.$
- $\|h\|_\omega^2 = \|h\|^2 + \|\partial_v h\|^2 + \|vh\|^2.$
- $\|h\|_V^2 = \|h\|^2 + \|\partial_x h\|^2.$
- $\|h\|_{V,\omega}^2 = \|h\|_\omega^2 + \|\partial_x h\|_\omega^2.$

For the metric space V, accordingly, one has the following Poincare inequality,

$$\|\partial_x h\|^2 \geq C_s \|h\|_V^2, \quad C_s \leq \frac{1}{2}, \quad \text{for } \forall h \in V \quad \text{and} \int h dx = 0. \tag{1.1}$$

Define U as the parameter space for $\mathbf{z}$, and assume $\mathbf{z}$ is a random vector with probability density function $\rho(\mathbf{z})$, so it has a corresponding weighted L^2 norm in the parameter space,

$$\|h\|_{L^2(U,d\rho)}^2 = \int_U h^2 \, d\rho = \int_U h^2 \rho(\mathbf{z}) \, d\mathbf{z}. \tag{1.2}$$

The following norms are defined in $\Omega \times U$:

- $\|h\|_{L^\infty(U,V)} = \sup_{\mathbf{z}\in U} \|h(\mathbf{z})\|_V;$
- $\|h\|_{L^\infty(U,W_x^{1,\infty})} = \sup_{\mathbf{z}\in U} \left(\|h(\mathbf{z})\|_{L_x^\infty} + \|\partial_x h(\mathbf{z})\|_{L_x^\infty} \right);$
- $\|h\|_{L^2(U,V,d\rho)}^2 = \int \|h(\mathbf{z})\|_V^2 \, d\rho.$

Define $\mathcal{F}$ to be the set of all sequences $\nu = (\nu_j)_{j\geq 1}$ of nonnegative integers such that only finite many ν_j are non-zeros. We call $\Lambda \subset \mathcal{F}$ an index set. The following notations are defined for index ν:

- $J(\nu) = \sup\{j : \nu_j > 0\}, \quad J(\Lambda) = \sup\{J(\nu) : \nu \in \Lambda\}$;
- $|\nu| = \sum_{j\geq 1} \nu_j$;
- $\nu! = \prod_{j\geq 1} \nu_j!$;
- $\partial_{\mathbf{z}}^{\nu} h = \partial_{z_1}^{\nu_1} \cdots \partial_{\mathbf{z}_n}^{\nu_n} h$, for $n = J(\nu)$;
- For infinite dimensional vector $d = (d_1, d_2, \cdots)$, $d^{\nu} = \prod_{j\geq 1} d_j^{\nu_j}$.

2 The parametric Vlasov-Fokker-Planck equation

Consider the following parametric Vlasov-Fokker-Planck equation,

$$\epsilon \partial_t f + v\partial_x f - \partial_x \phi \partial_v f = \frac{1}{\epsilon} \mathcal{Q} f, \quad x, v \in \Omega = (0, l) \times \mathbb{R} \tag{2.1}$$

for $\epsilon \leq 1$ (without loss of generality), with periodic condition on $x \in [0, l]$, and initial data $f(0, x, v, z) = f_0(x, v, z)$. Here f represents the probability density distribution of particles at position x with velocity v, ϵ represents the rescaled mean free path. $\mathcal{F}$ is the Fokker Planck operator that reads

$$\mathcal{Q} f = \partial_v \left(M \partial_v \left(\frac{f}{M} \right) \right), \tag{2.2}$$

with the global Maxwellian M,

$$M(v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{|v|^2}{2}}. \tag{2.3}$$

$\phi(t, x, \mathbf{z})$ is a given parametric potential that reads,

$$\phi(t, x, \mathbf{z}) = \bar{\phi}(t, x) + \sum_{j\geq 1} z_j \phi_j(t, x), \tag{2.4}$$

and $E(t, x, \mathbf{z}) = -\partial_x \phi$ is the parametric electric field that reads,

$$E(t, x, \mathbf{z}) = \bar{E}(t, x) + \sum_{j\geq 1} z_j E_j(t, x) = \partial_x \bar{\phi}(t, x) + \sum_{j\geq 1} z_j \partial_x \phi_j(t, x). \tag{2.5}$$

Here $\mathbf{z} \in U = [-1, 1]^{\infty}$ is an infinite dimensional parameter. We assume assume $\mathbf{z}$ to be i.i.d random variable following the uniform distribution on U, although other distributions can also be used. $\bar{\phi}(t, x)$ and $\bar{E}(t, x)$ are the expectations of ϕ, E respectively. We furthermore define the corresponding weighted norm $L^2(U, d\rho)$,

$$d\rho = \bigotimes_{j\geq 1} \frac{dz_j}{2}. \tag{2.6}$$

In addition, assume $\phi(t,x,\mathbf{z})$ and $E(t,x,\mathbf{z})$ converge to $\phi^\infty(x,\mathbf{z})$ and $E^\infty(x,\mathbf{z})$ respectively uniformly in time. That is, for $\forall\epsilon>0$, there exists T, such that

$$\sup_{\mathbf{z}\in U}\|\phi(t,x,\mathbf{z})-\phi^\infty(x,\mathbf{z})\|_{L^\infty_x}\le\epsilon,\quad \sup_{\mathbf{z}\in U}\|E(t,x,\mathbf{z})-E^\infty(x,\mathbf{z})\|_{L^\infty_x}\le\epsilon,\quad \text{for } \forall t\ge T. \tag{2.7}$$

It is easy to check that

$$F(x,v,\mathbf{z})=e^{-\phi^\infty}M(v) \tag{2.8}$$

is the stationary solution of (2.1), also called the global equilibrium. Let

$$h(t,x,v,\mathbf{z})=\frac{f(t,x,v,\mathbf{z})-F(x,v,\mathbf{z})}{\sqrt{M(v)}} \tag{2.9}$$

be the *perturbative* distribution function around F, and furthermore define the perturbative density and perturbative flux as follows,

$$\sigma=\int h\sqrt{M}\,dv,\quad u=\int hv\sqrt{M}\,dv. \tag{2.10}$$

Furthermore, in this paper, we will only focus on the randomness that comes from the electric random field. Therefore, we assume the following condition on the initial data.

Assumption 2.1. *Assume there is no initial random perturbation around the steady state F(x,v,z), and the initial perturbative mass is zero. That is, the initial data satisfies the following two equations:*

$$f(0,x,v,z)=F(x,v,z)+h(0,x,v)\sqrt{M}(v); \tag{2.11}$$

$$\int h(0,x,v)\sqrt{M}(v)dxdv=0. \tag{2.12}$$

It is easy to check that h satisfies,

$$\epsilon\partial_t h+v\partial_x h-\frac{1}{\epsilon}\mathcal{L}h=-E\left(\partial_v-\frac{v}{2}\right)h+v\sqrt{M}\left(E-E^\infty\right)e^{-\phi^\infty}, \tag{2.13}$$

where $\mathcal{L}$ is the linearized Fokker-Planck collision operator,

$$\mathcal{L}h=\frac{1}{\sqrt{M}}\partial_v\left(M\partial_v\left(\frac{h}{\sqrt{M}}\right)\right), \tag{2.14}$$

which satisfies the local coercivity property [9],

$$-\langle\mathcal{L}h,h\rangle\ge\lambda\|(1-\Pi)h\|^2_\omega, \tag{2.15}$$

where $\lambda<1$ is a constant, Π is the projection operator onto the null space of $\mathcal{L}$,

$$\Pi h=\left(\int_{\mathbb{R}}h\,dv\right)\sqrt{M}. \tag{2.16}$$

We call equation (2.13) the *microscopic equation*.

3 Decay rate of the best N approximation

In this section, we first review the best N approximation, and then study the convergence rate of this method applied to the Vlasov-Fokker-Planck equation with random electric field.

3.1 The best N approximation and our result

Since the solution $f(t,x,v,\mathbf{z}) = F + \sqrt{M}h(t,x,v,\mathbf{z})$, where F and M are given in (2.8) and (2.3) respectively, as long as one gets the approximate solution for h, then one can easily obtain the approximation solution for f. Hence we seek approximate solution h_Λ in a finite dimensional space,

$$\mathbb{P}_\Lambda = \{h_\Lambda : h_\Lambda = \sum_{\nu\in\Lambda} h_\nu(t,x,v)L_\nu(\mathbf{z})\}, \tag{3.1}$$

where Λ is an index set with infinite dimensional vectors ν. Here $L_\nu(\mathbf{z})$ is the orthonormal Legendre polynomial which forms a basis in $L^2(U,d\rho)$ such that,

$$L_\nu = \prod_{j\geq 1} L_{\nu_j}(z_j), \quad \int_{-1}^{1} L_k(z_j)L_l(z_j)\frac{dz_j}{2} = \delta_{kl}. \tag{3.2}$$

If h solves (2.13), then naturally one has the projection of the solution h onto $\mathbb{P}_\Lambda$,

$$P_\Lambda h = \sum_{\nu\in\Lambda}\left(\int_U hL_\nu d\rho\right)L_\nu := \sum_{\nu\in\Lambda} h_\nu L_\nu = \underset{h_\Lambda\in\mathbb{P}_\Lambda}{\operatorname{argmin}} \|h - h_\Lambda\|_{L^2(U,V,d\rho)}. \tag{3.3}$$

The best N approximation is a form of nonlinear approximation that searches for $\nu\in\Lambda$ according to the largest N coefficients $\|h_\nu\|_V$. It is proved in [5] that the decay rate of such approximation depends on the holomorphy and anisotropy of the solution in the random space, as stated in the following theorem.

Theorem 3.1 (Corollary 3.11 of [5]). *Consider a parametric equation of the form*

$$\mathcal{P}(f,a) = 0, \tag{3.4}$$

with random field $a = \bar{a}(x) + \sum_{j\geq 1} z_j\psi_j(x) \in X$, $\mathbf{z}\in U$, where X is certain space of x. Assume the solution map $a\to f(a)$ admits a holomorphic extension to an open set $\mathcal{O}\in X$ which contains $a(U) = \{a(\mathbf{z}) : \mathbf{z}\in U\}$, with uniform bound

$$\sup_{a\in\mathcal{O}} \|f(a)\|_V \leq C. \tag{3.5}$$

If in addition $\left(\|\psi_j\|_X\right)_{j\geq 1} \in \ell^p(\mathbb{N})$ for some $p<1$, then for the set of indices Λ_n that corresponds to the n largest $f_\nu = \left\|\int fL_\nu d\rho\right\|_V$, one has,

$$\left\|f - \sum_{\nu\in\Lambda_n} f_\nu L_\nu\right\|_{L^2(U,V,d\rho)} \leq \frac{C_p}{(n+1)^s}, \quad s = \frac{1}{p} - \frac{1}{2} \tag{3.6}$$

where $C_p := \left\|(\|f_\nu\|_V)_{\nu\in\mathcal{F}}\right\|_{\ell^p}$.

In order to apply the above theorem, one needs to prove the holomorphy of the solution map. In the case of kinetic equation, since we are only dealing with a real function, the holomorphy of a solution map is equivalent to: There exists constant B and C_j, such that

$$\left\|\partial_{z_j}^k h\right\|_V^2 \leq BC_j^k k!, \quad \text{for any nonnegative integer } k. \tag{3.7}$$

This result will be proved in Theorem 3.4. In the following context, ν always represents an infinite dimensional index, e_j is an infinite dimensional vector with only the j-th component being 1 and all others zeros. ∂^ν represents $\partial_{\mathbf{z}}^\nu$ for the convenience of writing. Theorems 3.4 and 3.5 are both based on the following assumptions on $E(t,x,\mathbf{z})$ and $\{E_j(t,x)\}_{j\geq 1}$.

Assumption 3.2. *Assume $E(t,x,\mathbf{z}) = \bar{E}(t,x) + \sum_{j\geq 1} z_j E_j(t,x)$ satisfies the following assumptions,*

$$\sup_{t\geq 0} \|E_j(t)\|_{L^\infty\left(U,W_x^{1,\infty}\right)} \leq C_j, \quad (C_j)_{j\geq 1} \in l^p(\mathbb{N}), \text{ for some } p \leq 1 \tag{3.8}$$

$$\|E(t)\|_{L^\infty(U,W_x^{1,\infty})} \leq \left\|\bar{E}(t)\right\|_{W_x^{1,\infty}} + \sum_{j\geq 1} \|E_j(t)\|_{W_x^{1,\infty}} \leq C_E, \text{ for all } t, \quad C_E \leq \frac{\lambda C_s}{8}. \tag{3.9}$$

There exists a continuous function $D(t)$, such that

$$\left\|\partial^\nu \left((E - E^\infty)\, e^{-\phi^\infty}\right)\right\|^2_{L^\infty(U,V)} \leq D(t)(K^\nu \nu!)^2, \text{ for all } t, \quad \int_0^\infty D(s)dt = \bar{D}, \tag{3.10}$$

where $(K_j)_{j\geq 1} \in l^p(\mathbb{N})$, for some $p \leq 1$.

Remark 3.3. *If one uses the Karhunen-Loève expansion to parametrize the random field, then the smoothness properties of the covariance function for the random field determine the ℓ_p-summability of the random variables. For random field $a(\omega,x) \in L^2(\Omega, dP; L^\infty(D))$ in a polyhedral domain $D \subset \mathbb{R}^d$ with mean field $E_a(x) = \int_\Omega a(\omega,x)dP(\omega)$ and covariance $V_a(x,x') = \int_\Omega (a(\omega,x) - E_a(x))\,(a(\omega,x') - E_a(x'))\,dP(\omega)$, if the stationary covariance $g_a(z)$ is analytic outside of $z = 0$, and C^k at zero, where $V_a(x,x') = g_a(|x-x'|)$, then it is $\ell^{d/k}$-summable. See [38, 40] for details.*

In the above assumptions, equation (3.8) guarantees the anisotropy of E. (3.9) - (3.10) are required for the analyticity of the solution in the random space. Basically, it requires E to be bounded and converges to E^∞ fast enough so that the improper integral $D(t)$ exists.

We first state our Theorem about the analyticity of h.

Theorem 3.4. *Under Assumptions 2.1 and 3.2 the ν-th derivative of the perturbative solution to (2.13) in the random space can be bounded as follows,*

$$\|\partial^\nu h(t)\|_{L^\infty(U,V)} \leq Q(t)\,(|\nu|!)\,d^\nu, \tag{3.11}$$

where d is an infinite dimensional vector with the j-th component

$$d_j = \max\left\{\sqrt{\frac{20C_j}{\lambda C_s}}, K_j\right\}, \tag{3.12}$$

where C_j, K_j are defined in (3.8), (3.10) respectively; $Q(t)$ is a function exponentially decaying in t,

$$Q(t) = \min\left\{\frac{1}{\epsilon}e^{-\frac{\xi}{\epsilon^2}t}, e^{-\xi t}\right\} 2\left(\|h(0)\|_V + \sqrt{\frac{\lambda\bar{D}}{C_E}}\right), \tag{3.13}$$

for $\xi = \frac{\lambda C_s}{10}$, $C_s, \lambda, \bar{D}$ are constants defined in (1.1), (2.15), (3.10) respectively.

By taking $\nu = ke_j$, one can derive the inequality (3.7), which implies the analyticity of h in the random space. Therefore, based on the above theorem and assumptions, we can conclude that the best N approximation converges independent of dimensionality of the parameter $\mathbf{z}$ and faster than the Monte Carlo method, as stated in the following theorem:

Theorem 3.5. *If $E(t,x,\mathbf{z})$ satisfies Condition 3.2, then the approximate solution obtained by the best N approximation converges to the exact solution with the error,*

$$\|f - f_\Lambda\|_{L^2(V,U,d\rho)} \leq \frac{C_p}{(n+1)^s}, \quad s = \frac{1}{p} - \frac{1}{2}, \tag{3.14}$$

where $p \leq 1$ depends on (3.9), $C_p = \|(\|f_\nu\|_V)_{\nu\in\mathcal{F}}\|_{\ell^p} < \infty$.

Remark 3.6. *How large is C_p?*

- *For the case when $\left\|d/\sqrt{3}\right\|_{\ell^1} < 1$:*
  [6] gives a way to calculate the upper bound for C_p when $\left\|d/\sqrt{3}\right\|_{\ell^1} < 1$. First by Rodrigre's Formula (See Section 6 of [6]),

$$\|h_\nu\|_V = \left\| \int h L_\nu d\rho \right\|_V = \frac{\left(\sqrt{3}\right)^{-\nu}}{\nu!} \|\partial^\nu h\|_{L^\infty(U,V)}.$$

  Then by the estimate in (3.11), one has

$$\|h_\nu\|_V \leq B(t) \frac{|\nu|!}{\nu!} \left(\frac{d}{\sqrt{3}}\right)^\nu.$$

  According to Theorem 7.2 of [6], let $\alpha = \frac{d}{\sqrt{3}}$ be an infinite dimensional vector, then if $\|\alpha\|_{\ell^1} \leq 1$, and $\alpha \in \ell^p$, one has

$$\|(\|h_\nu\|_V)_{\nu\in\mathcal{F}}\|_{\ell^p} \leq \frac{2}{\eta} \exp\left(\frac{2(1-p)(J(\eta) + \|\alpha\|_{\ell^p}^p)}{p^2\eta}\right) \tag{3.15}$$

  where $\eta = \frac{1-\|\alpha\|_{\ell^1}}{2}$, $J(\eta)$ is the smallest positive integer such that $\sum_{j\geq J} |\alpha_j|^p \leq \frac{\eta}{2}$.

- *For the case $\left\|d/\sqrt{3}\right\|_{\ell^1} \geq 1$:*
  There is no explicit expression for the upper bound of C_p (See Remark 3.22 of [5]).

3.2 Proof of Theorem 3.4

Define a Lyapunov functional

$$G_i^\nu = \theta_i \left(\frac{\epsilon}{2} \|\partial^\nu h\|_V^2\right) + \frac{1}{2\epsilon} \left(\epsilon \langle \partial^\nu u, \partial_x \partial^\nu \sigma\rangle + \frac{1}{2} \|\partial^\nu \sigma\|^2\right), \quad i = 1, 2. \tag{3.16}$$

with $\theta_1 = \frac{8}{7\lambda}$, $\theta_2 = \frac{8}{7\lambda\epsilon^2}$. Similar Lyapunov functional has been introduced in [15] for the deterministic nonlinear Vlasov-Poisson-Fokker-Planck (VPFP) system with $\epsilon = 1$. For the case where the uncertainty and scaling parameter ϵ are involved, [22] gives a modified Lyapunov functional, which is more suitable for different scaling of ϵ. Actually, G_i^ν is equivalent to $\|\partial^\nu h\|_V^2$. Since by Young's inequality, one has

$$-\frac{\epsilon^2}{2} \|\partial^\nu \partial_x u\|^2 - \frac{1}{2} \|\partial^\nu \sigma\|^2 \leq \epsilon \langle \partial^\nu \partial_x u, \partial^\nu \sigma\rangle \leq \frac{\epsilon^2}{2} \|\partial^\nu \partial_x u\|^2 + \frac{1}{2} \|\partial^\nu \sigma\|^2,$$

so,

$$-\frac{\epsilon}{4} \|\partial^\nu \partial_x u\|^2 \leq \frac{1}{2\epsilon} \left(-\epsilon \langle \partial^\nu \partial_x u, \partial^\nu \sigma\rangle + \frac{1}{2} \|\partial^\nu \sigma\|^2\right) \leq \frac{\epsilon}{4} \|\partial^\nu \partial_x u\|^2 + \frac{1}{2\epsilon} \|\partial^\nu \sigma\|^2.$$

Because that $\|u\|^2, \|\sigma\|^2 \le \|h\|^2$, the above inequality becomes

$$-\frac{\epsilon}{4}\|\partial^\nu h\|_V^2 \le \frac{1}{2\epsilon}\left(-\epsilon\langle\partial^\nu\partial_x u, \partial^\nu\sigma\rangle + \frac{1}{2}\|\partial^\nu\sigma\|^2\right) \le \frac{1}{2\epsilon}\|\partial^\nu h\|_V^2. \tag{3.17}$$

Plug the above inequalities to the definition of G_i^ν, one ends up with

$$\text{for } \theta_1 = \frac{8}{7\lambda}, \quad \frac{\epsilon}{2\lambda}\|\partial^\nu h\|_V^2 \le G_1^\nu \le \frac{3}{2\lambda\epsilon}\|\partial^\nu h\|_V^2 \tag{3.18}$$

$$\text{for } \theta_2 = \frac{8}{7\lambda\epsilon^2}, \quad \frac{1}{2\lambda\epsilon}\|\partial^\nu h\|_V^2 \le G_2^\nu \le \frac{3}{2\lambda\epsilon}\|\partial^\nu h\|_V^2. \tag{3.19}$$

Lemma 3.7. *Under Condition 3.2, for $\forall \mathbf{z} \in U$, the following estimates hold,*

$$\text{for } |\nu| = 0 : \partial_t G_i^0 + \frac{\eta}{\epsilon}\|h\|_V^2 \le \frac{1}{\epsilon C_E}\left\|\partial^\nu\left((E - E^\infty)e^{-\phi^\infty}\right)\right\|_V^2; \tag{3.20}$$

$$\text{for } |\nu| > 1 : \partial_t G_i^\nu + \frac{\eta}{\epsilon}\|h\|_V^2 \le \frac{2}{\lambda\epsilon}\sum_{\nu_j\neq 0}\nu_j^2 C_j\left\|\partial^{\nu-e_j}h\right\|_V^2 + \frac{1}{\epsilon C_E}\left\|\partial^\nu\left((E - E^\infty)e^{-\phi^\infty}\right)\right\|_V^2, \tag{3.21}$$

where $i = 1, 2$, $\eta = \frac{C_s}{10}$, and

$$\frac{\epsilon}{2\lambda}\|\partial^\nu h\|_V^2 \le G_1^\nu \le \frac{3}{2\lambda\epsilon}\|\partial^\nu h\|_V^2, \quad \frac{1}{2\lambda\epsilon}\|\partial^\nu h\|_V^2 \le G_2^\nu \le \frac{3}{2\lambda\epsilon}\|\partial^\nu h\|_V^2. \tag{3.22}$$

Proof. See Appendix A. ∎

Lemma 3.8. *For fixed $\mathbf{z}$, the following estimates hold,*

$$G_i^\nu(t) \le (|\nu|!)^2 (2d)^{2\nu}\left(G_i^0(0) + \frac{\bar{D}}{\epsilon C_E}\right) - \frac{\eta}{\epsilon}\int_0^t \|\partial^\nu h(s)\|_V^2\, ds, \tag{3.23}$$

where $d = (d_1, d_2, \cdots)$ with $d_j = \max\left\{\sqrt{\frac{2C_j}{\lambda\eta}}, K_j\right\}$.

Proof. First for $|\nu| = 0$, by (3.20) in Lemma 3.7, one has

$$G_i^0(t) \le G_i^0(0) + \frac{1}{\epsilon C_E}\int_0^t D(s)ds - \frac{\eta}{\epsilon}\int_0^t \|h(s)\|^2\, ds, \tag{3.24}$$

which satisfies (3.23). Then by induction, assume the following holds,

$$G_i^{\nu-e_j}(t) \le ((|\nu|-1)!)^2 (2d)^{2(\nu-e_j)}\left(G_i^0(0) + \frac{\bar{D}}{\epsilon C_E}\right) - \frac{\eta}{\epsilon}\int_0^t \left\|\partial^{\nu-e_j}h(s)\right\|_V^2 ds. \tag{3.25}$$

By integrating (3.21) over t, one has,

$$G_i^\nu(t) \le G_i^\nu(0) - \frac{\eta}{\epsilon}\int_0^t \|\partial^\nu h(s)\|_V^2\, ds + \frac{2}{\lambda\epsilon}\sum_{\nu_j\neq 0}\nu_j^2 C_j\int_0^t \left\|\partial^{\nu-e_j}h(s)\right\|_V^2 ds + \frac{\bar{D}}{\epsilon C_E}(K^\nu\nu!)^2. \tag{3.26}$$

Since the initial perturbation h is independent of the parameter $\mathbf{z}$, so $G_i^\nu(0) = 0$ for $\forall|\nu| > 0$. Multiplying $\frac{2\nu_j^2 C_j}{\lambda\eta}$ to (3.25), and summing it over $\nu_j \neq 0$, then combining it with (3.26), one gets

$$\begin{aligned} G_i^\nu(t) + \sum_{\nu_j\neq 0}\frac{2\nu_j^2 C_j}{\lambda\eta}G_i^{\nu-e_j}(t) \le{}& \sum_{\nu_j\neq 0}\frac{2\nu_j^2 C_j}{\lambda\eta}((|\nu|-1)!)^2(2d)^{2(\nu-e_j)}\left(G_i^0(0) + \frac{\bar{D}}{\epsilon C_E}\right) \\ &+ \frac{\bar{D}}{\epsilon C_E}(K^\nu\nu!)^2 - \frac{\eta}{\epsilon}\int_0^t \|\partial^\nu h(s)\|_V^2\, ds. \end{aligned} \tag{3.27}$$

Since $G^{\nu-e_j}(t)$ is always positive, we can omit the second term on the LHS. In addition, note that $\frac{2C_j}{\lambda\eta}(2d)^{2(\nu-e_j)} \leq (2d)^{2\nu}/2$, so

$$\begin{aligned}
&\frac{\bar{D}}{\epsilon C_E}(K^\nu \nu!)^2 + \sum_{\nu_j\neq 0} \frac{2\nu_j^2 C_j}{\lambda\eta}\left((|\nu|-1)!\right)^2 (2d)^{2(\nu-e_j)}\left(G_i^0(0)+\frac{\bar{D}}{\epsilon C_E}\right)\\
\leq&\frac{\bar{D}}{\epsilon C_E}(d^\nu \nu!)^2 + (2d)^{2\nu}\frac{1}{2}\left(G_i^0(0)+\frac{\bar{D}}{\epsilon C_E}\right)\left((|\nu|-1)!\right)^2 \sum_{\nu_j\neq 0}\nu_j^2\\
\leq&\frac{\bar{D}}{\epsilon C_E}(d^\nu \nu!)^2 + \left((2d)^\nu|\nu|!\right)^2\frac{1}{2}\left(G_i^0(0)+\frac{\bar{D}}{\epsilon C_E}\right)\\
\leq&\left((2d)^\nu|\nu|!\right)^2\left(\frac{1}{2}\left(G_i^0(0)+\frac{\bar{D}}{\epsilon C_E}\right)+\frac{1}{2}\frac{\bar{D}}{\epsilon C_E}\right) \leq \left((2d)^\nu|\nu|!\right)^2\left(G_i^0(0)+\frac{\bar{D}}{\epsilon C_E}\right) \qquad (3.28)
\end{aligned}$$

where the second inequality is because of $\sum_{\nu_j\neq 0}\nu_j^2 \leq |\nu|^2$, and the third inequality holds for any $|\nu|>0$. Plugging (3.28) into (3.27), and omitting the second term on the LHS give (3.23) complete the induction and consequently the proof for Lemma 3.8. ■

From Lemma 3.8, and the equivalent relationship between G_1^ν and $\|\partial^\nu h\|_V^2$ in (3.22), (3.23), one has

$$\begin{aligned}
\frac{\epsilon}{2\lambda}\|\partial^\nu h\|_V^2 &\leq (|\nu|!(2d)^\nu)^2\left(\frac{3}{2\lambda\epsilon}\|h(0)\|_V^2+\frac{\bar{D}}{\epsilon C_E}\right)-\frac{\eta}{\epsilon}\int_0^t\|\partial^\nu h(s)\|_V^2\,ds,\\
\|\partial^\nu h\|_V^2 &\leq (|\nu|!b^\nu)^2\left(\frac{3}{\epsilon^2}\|h(0)\|_V^2+\frac{2\lambda\bar{D}}{\epsilon^2 C_E}\right)-\frac{2\lambda\eta}{\epsilon^2}\int_0^t\|\partial^\nu h(s)\|_V^2\,ds. \qquad (3.29)
\end{aligned}$$

By Grownwall's inequality

$$\|\partial^\nu h(t)\|_V \leq \frac{2}{\epsilon}|\nu|!b^\nu\left(\|h(0)\|_V+\sqrt{\frac{\lambda\bar{D}}{C_E}}\right)e^{-\frac{\xi t}{\epsilon^2}}, \qquad (3.30)$$

for $\xi=\frac{\lambda C_s}{10}$. Similarly, for G_2^ν, by (3.22) and (3.23), one obtains

$$\|\partial^\nu h(t)\|_V^2 \leq (|\nu|!b^\nu)^2\left(3\|h(0)\|_V^2+\frac{2\lambda\bar{D}}{C_E}\right)-2\lambda\eta\int_0^t\|\partial^\nu h(s)\|_V^2\,ds.$$

Grownwall's inequality then implies,

$$\|\partial^\nu h(t)\|_V \leq 2|\nu|!b^\nu\left(\|h(0)\|_V+\sqrt{\frac{\lambda\bar{D}}{C_E}}\right)e^{-\xi t}, \qquad (3.31)$$

which gives the conclusion in Theorem 3.4.

4 The Numerical Method

The convergence rate obtained in Theorem 3.5 is based on the best N approximation, which means one needs to calculate all coefficients of the Legendre series h_ν in order to find the N largest $\|h_\nu\|_V$. In practice, one needs a more efficient numerical method to find the best basis. Based on the greedy search method introduced in [2], in line of [33, 32], we formulate a new *residual based adaptive sparse polynomial interpolation* (RASPI) method. We will first introduce

the framework of the *adaptive sparse polynomial interpolation* (ASPI) method in Section 4.1. Then in Section 4.2 the new residual based method will be introduced, and the reason why this method is computationally efficient, particularly for time dependent kinetic equation when ϵ is small, is also explained in Section 4.2. Finally a comparison of the computational cost between the ASPI and the RASPI methods for general kinetic equations is given in Section 4.3.

In this section, we assume $\|E_j(t,x)\|_V$ decreases as j increases for all $t \geq 0$.

4.1 The adaptive sparse polynomial interpolation (ASPI)

The ASPI is a numerical method that approximates the solution map by a sparse polynomial interpolation at well chosen points. Let us first define the representation of infinite dimensional random variable and polynomial interpolation bases. For a sequence $\Gamma = (\beta_k)_{k\geq 0}$ of distinct points in $[-1,1]$, and index $\nu = (\nu_j)_{j\geq 1}$, define points

$$\mathbf{z}_\nu = (\beta_{\nu_j})_{j\geq 1} \tag{4.1}$$

and hierarchical Lagrange basis

$$H_\nu(\mathbf{z}) = \prod_{j\geq 1} l_{\nu_j}(z_j), \quad l_0 = 1, \quad l_k(\beta) = \prod_{m=0}^{k-1} \frac{\beta - \beta_m}{\beta_k - \beta_m}. \tag{4.2}$$

Note that

$$H_\nu(\mathbf{z}_\nu) = 1, \quad \text{for all } \nu \in \mathcal{F}, \qquad\qquad H_\nu(\mathbf{z}_{\tilde{\nu}}) = 0, \quad \text{for all } \tilde{\nu} < \nu. \tag{4.3}$$

Here $\tilde{\nu} \leq \nu$ if and only if all components of $\tilde{\nu}$ are smaller than or equal to ν; $\tilde{\nu} < \nu$ represents that $\nu \leq \nu$ and $\tilde{\nu} \neq \nu$. We call the index set $\{\Lambda_k\}_{k\geq 1}$ *monotone* if $\Lambda_k \subset \Lambda_{k+1}$ for all k. We further call index set $\Lambda \subset \mathcal{F}$ *downward closed*,

$$\text{if } \nu \in \Lambda, \quad \tilde{\nu} \leq \nu, \quad \text{then } \tilde{\nu} \in \Lambda. \tag{4.4}$$

Secondly, when is the infinite dimensional polynomial interpolation well defined? Actually for a downward closed set $\Lambda \subset \mathcal{F}$, given the grid $\mathbf{z}_\Lambda$ and the corresponding solution f_Λ on the grids,

$$\mathbf{z}_\Lambda := \{\mathbf{z}_\nu, \nu \in \Lambda\}, \quad f_\Lambda := (f_\nu)_{\nu\in\Lambda} := (f(t,x,v,\mathbf{z}_\nu))_{\nu\in\Lambda}\,, \tag{4.5}$$

there exists a unique polynomial

$$I_\Lambda(t,x,v,\mathbf{z}) = \sum_{\nu\in\Lambda} \alpha_\nu(t,x,v) H_\nu(\mathbf{z}), \tag{4.6}$$

such that I_Λ has the same value as f_Λ at $\mathbf{z}_\Lambda$. Namely, I_Λ is the polynomial interpolation of f_Λ at interpolating points $\mathbf{z}_\Lambda$. From the above framework, the multi-dimensional polynomial interpolation is uniquely determined by the sequence Γ and index set Λ. There are three questions to be answered at this point.

- How to choose the sequence $\Gamma = (\beta_k)_{k\geq 0}$;
- How to calculate I_Λ if given $\mathbf{z}_\Lambda$ and f_Λ;

- How to find the Λ_n with $\#(\Lambda_n) = n$, such that I_{Λ_n} is the closest to $f(t, x, v, \mathbf{z})$,

where $\#(\Lambda_n)$ represents number of elements in Λ_n.

Choosing different sequences will result in different stability and accuracy of the interpolation mapping, which is characterized by the Lebesgue constant. The Leja sequence is usually considered a good choice, which starts with an arbitrary $\beta_0 \in [-1, 1]$, and then defined by,

$$\beta_k := \operatorname{argmax} \left\{ \prod_{l=0}^{k-1} |\beta - \beta_l| : \beta \in [-1, 1] \right\}. \tag{4.7}$$

[2] proved that if the Lebesgue constant of a univariate polynomial interpolation on sequence $\{\beta_l\}_{l=0}^k$ is $O\left((k+1)^\theta\right)$, then the Lebesgue constant λ_Λ of polynomial interpolation on $\mathbf{z}_\Lambda$ is $O(\#(\Lambda)^{\theta+1})$ for any monotone set Λ. [4] proved the Lebesgue constant on the Leja sequences is less than $3(k+1)^2 \log(k+1)$, which implies the Lebesgue constant of the multidimensional polynomial interpolation on $\mathbf{z}_\Lambda$ is less than $O\left(\#(\Lambda)^4\right)$.

After determining the sequence Γ, given arbitrary $\Lambda_n = \{\nu_1, \cdots, \nu_n\}$, and the corresponding $\mathbf{z}_{\Lambda_n}, f_{\Lambda_n}$, since the interpolation polynomial satisfies

$$\begin{bmatrix} H_{\nu_1}(\mathbf{z}_{\nu_1}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_1}) \\ \vdots \\ H_{\nu_1}(\mathbf{z}_{\nu_n}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_n}) \end{bmatrix} \begin{bmatrix} \alpha_{\nu_1} \\ \vdots \\ \alpha_{\nu_n} \end{bmatrix} = \begin{bmatrix} f_{\nu_1} \\ \vdots \\ f_{\nu_n} \end{bmatrix}, \tag{4.8}$$

one can invert the first matrix to get the coefficient $(\alpha_\nu)_{\nu \in \Lambda_n}$. In general, one needs to do the inversion all over again if the index Λ_n changes.

However, if $\{\Lambda_n\}_{n=1}^N$ is monotone and downward closed, there is a progressive construction of the interpolation operator, which allows to avoid inverting a matrix. If $\Lambda_n = \Lambda_{n-1} \cup \{\nu_n\}$, then

$$I_{\Lambda_n} = I_{\Lambda_{n-1}} + \alpha_{\nu_n} H_{\nu_n}, \quad \alpha_{\nu_n} = f_{\nu_n} - I_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}), \quad \text{with } I_{\Lambda_0} = 0. \tag{4.9}$$

Actually, one can prove this by induction. For $n = 1$, $I_{\Lambda_1} = f_{\nu_1}$ is indeed the interpolation on $\mathbf{z}_{\nu_1}$. Assume $I_{\Lambda_{n-1}}$ constructed in the above way is the interpolation on $\mathbf{z}_{\Lambda_{n-1}}$, then since $\nu_1 < \cdots < \nu_{n-1} < \nu_n$, so by (4.3), $H_{\nu_n}(\mathbf{z}_{\nu_k}) = 0$, for $k \le n-1$; $H_{\nu_n}(\mathbf{z}_{\nu_n}) = 1$. Therefore

$$k \le n-1: \quad I_{\Lambda_n}(\mathbf{z}_{\nu_k}) = I_{\Lambda_{n-1}}(\mathbf{z}_{\nu_k}) = f_{\nu_k}, \tag{4.10}$$

$$k = n: \quad I_{\Lambda_n}(\mathbf{z}_{\nu_n}) = I_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}) + (f_{\nu_n} - I_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n})) = f_{\nu_n}, \tag{4.11}$$

which implies that I_{Λ_n} is the interpolation operator on $\mathbf{z}_{\Lambda_n}$.

In order to use this progressive construction to find the interpolation operator, we require the index set $\{\Lambda_n\}_{n=1}^N$ to be monotone and downward closed, that is,

$$\Lambda_{n+1} = \Lambda_n \cup \{\nu_{n+1}\}, \quad \nu_n \in N(\Lambda_n), \quad N(\Lambda_n) = \{\nu \notin \Lambda_n, \Lambda_n \cup \{\nu\} \text{ is downward closed}\}$$

where we call $N(\Lambda_n)$ the neighborhood of index set Λ_n.

Now we come to the last question. Assume we already determined Λ_n, in order to find the best Λ_{n+1}, how should one select the optimal ν_{k+1} from the neighborhood of Λ_n? First we notice for infinite dimensional $\mathbf{z}$, $\#\{N(\Lambda_k)\}$ is also infinite. Even for finite dimension $\mathbf{z} \in \mathbb{R}^d$,

$\#\{N(\Lambda_k)\} \sim O(k^d)$, which is too big to search numerically. So we introduce *anchored neighbors* $\tilde{N}(\Lambda)$,

$$\tilde{N}(\Lambda) = \{\nu \in N(\Lambda) : \nu_j = 0 \text{ if } j < j(\Lambda) + 1\}, \quad j(\Lambda) = \max\{j : \nu_j > 0 \text{ for some } \nu \in \Lambda\} \quad (4.12)$$

The reason why searching the anchored neighbor makes sense is because we assume at the beginning of this section $\|E_j(t,x)\|_{W_x^{1,\infty}}$ decreases as j increases, then from Theorem 3.4, one notices the upper bound of $\left\|\partial_{z_j} h(t,\mathbf{z})\right\|_V$ decreases as j increases, which formally indicates that z_j becomes less sensitive when j increases. So if for all $\nu \in \Lambda_n$, the components larger than and equal to $(j(\Lambda_n)+1)$ of $\mathbf{z}_\nu$ are the same, then when searching for the next interpolation point, one should first consider adding a point along $(j(\Lambda_k)+1)$-st component before all the other components larger than $(j(\Lambda_k)+1)$.

Note that because of the monotonicity of Λ_n, one can actually construct $\tilde{N}(\Lambda_n)$ based on $\tilde{N}(\Lambda_{n-1})$ in the following way. Define

$$\hat{N}(\Lambda_n) = \{e_{j(\Lambda_n)+1}, \nu_n + e_j, j \le j(\nu_n)\}, \quad j(\nu) = \max\{j : \nu_j > 0\} \quad (4.13)$$

$$N^*(\Lambda_n) = \hat{N}(\Lambda_n) \backslash \left(\tilde{N}(\Lambda_{n-1}) \cap \hat{N}(\Lambda_n)\right), \quad (4.14)$$

then

$$\tilde{N}(\Lambda_n) = \tilde{N}(\Lambda_{n-1}) \cup N^*(\Lambda_n). \quad (4.15)$$

Here is an example that shows the anchored neighbors in three dimension. Since we assume

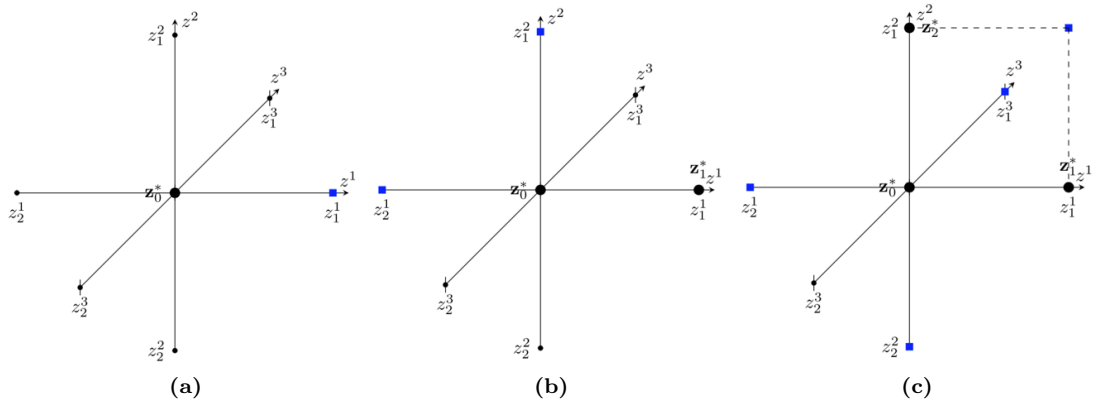


Figure 1: The blue dots represent the anchored neighbors of $\Lambda_0 = \{\mathbf{z}_0^*\}, \Lambda_1 = \{\mathbf{z}_0^*, \mathbf{z}_1^*\}, \Lambda_2 = \{\mathbf{z}_0^*, \mathbf{z}_1^*, \mathbf{z}_2^*\}$ for $\mathbf{z} = (z^1, z^2, z^3) \in [-1,1]^3$.

the direction z^1 is more important than z^2, z^3, so in Figure 1(a), we explore more points in the direction z^1 first, so $\tilde{N}(\{(0,0,0)\}) = \{(1,0,0)\}$. Then in Figure 1(b), since we already have 2 points on the z^1-axis, instead of exploring more points on the z^1 direction, we start to explore the z^2 direction, so $\tilde{N}(\{(0,0,0),(1,0,0)\}) = \{(2,0,0),(0,1,0)\}$. Assume after doing greedy search on $\tilde{N}(\Lambda_1)$, one gets $\nu_2 = (0,1,0)$. Then in Figure 1(c), Since one has two points on z^1, z^2 respectively, so one starts to explore more points on the third direction z^3 at this step. so $\tilde{N}(\{(0,0,0),(1,0,0),(0,1,0)\}) = \{(2,0,0),(0,2,0),(1,1,0),(0,0,1)\}$

Note that the size of $N^*(\Lambda_n)$ depends on $j(\nu_n)$, and $j(\nu_n) \leq n$, so

$$\#(\tilde{N}(\Lambda_n)) \leq \#(\tilde{N}(\Lambda_{n-1})) + n, \tag{4.16}$$

which gives the size of $\tilde{N}(\Lambda_n)$ is at most

$$\#(\tilde{N}(\Lambda_n)) \leq \frac{1}{2}n(n-1) \sim O(n^2). \tag{4.17}$$

After one constructed the anchored neighbors of Λ_n, one searches for the $\nu \in \tilde{N}(\Lambda_n)$ that maximizes the interpolation error at the new grid point. In summary, we have the following algorithm.

Algorithm 4.1. *(ASPI) [2]*

- *Step* 0. *Construct the Leja sequence* $\Gamma = \{\beta_j\}_{j\geq 0}$ *starting from* 0 *as in (4.7) and the basis* $\{l_k(\beta)\}_{k\geq 1}$ *as in* (4.2).
- *Step* 1. *Define* $\Lambda_1 = \{0\}$ *as the null multi-index and the corresponding polynomial interpolation* $I_{\Lambda_1}(\mathbf{z}) = f(\mathbf{z}_{\nu_1})$.
- *Step n. Assume we already have* $\tilde{N}(\Lambda_{n-1})$, Λ_n *and* I_{Λ_n}.
  - *Construct* $\hat{N}(\Lambda_n)$ *by (4.13), then* $N^*(\Lambda_n)$ *can be constructed through (4.14),* $\tilde{N}(\Lambda_n)$ *through (4.15).*
  - $\nu_{n+1} = \underset{\nu\in\tilde{N}(\Lambda_n)}{\operatorname{argmax}} \left\|I_{\Lambda_{n+1}} - I_{\Lambda_n}\right\|_{L^2(U,V,d\rho)} = \underset{\nu\in\tilde{N}(\Lambda_n)}{\operatorname{argmax}} \|\alpha_\nu\|_V \|H_\nu\|_{L^2(U,d\rho)}$, *where* $\alpha_\nu(t,x,v)$, $H_\nu(\mathbf{z})$ *are defined in (4.9), (4.2) respectively.*

After step 0, the polynomial interpolation will be uniquely depending on the index set Λ. In step n, it gives the way to find the best Λ_n such that $I_{\Lambda_n}(t,x,v,\mathbf{z})$ is closest to $f(t,x,v,\mathbf{z})$ in the $L^2(U,V,d\rho)$ space. In the greedy search step, the reason why $\left\|I_{\Lambda_{n+1}} - I_{\Lambda_n}\right\|_{L^2(U,V,d\rho)} = \|\alpha_\nu\|_V \|H_\nu\|_{L^2(U,d\rho)}$ is because of the progressive construction of the polynomial interpolation, which is interpreted in (4.9). $\operatorname{argmax}_{\nu\in\tilde{N}(\Lambda_n)}$ is obtained by directly searching for the maximal value.

Furthermore, note that when calculating α_ν, one actually needs to calculate the function value $f(T,x,v,\mathbf{z}_\nu)$ at point $\mathbf{z}_\nu$. Although the final approximate solution $f(T,x,v,\mathbf{z})$ is a polynomial interpolating on N points $f(\mathbf{z}_{\nu_1}),\cdots,f(\mathbf{z}_{\nu_N})$, in order to get the best $\mathbf{z}_{\nu_{n+1}}$ $(1 \leq n \leq N)$, one needs to do the greedy search on $\tilde{N}(\Lambda_n)$, which includes calculating $f(\mathbf{z}_\nu)$ for all $\nu \in \tilde{N}(\Lambda_n)$. Since most PDEs have no analytic solution, the computational cost of obtaining the solution at sample point $\mathbf{z}_\nu$ highly depends on the numerical algorithm. We will see in the next section, the ASPI method is computationally inefficient for time dependent kinetic equation with small ϵ.

4.2 The residual based adaptive sparse polynomial interpolation (RASPI)

As stated at the end of the previous section, we will explain in more details in this section why the ASPI is not as efficient as the RASPI for general linear kinetic equation. The general form of a kinetic equation without uncertainty reads,

$$\partial_t f + v\partial_x f = \frac{1}{\epsilon}Q(f), \tag{4.18}$$

where $f(t,x,v)$ is the probability density distribution of particles, $Q(f)$ describes the collision between particles. The parameter ϵ represents the dimensionless mean free path or the Knudsen number, which connects the microscopic kinetic model to the macroscopic hydrodynamic model when $\epsilon \to 0$. Kinetic equations give a uniform description of both mesoscopic and macroscopic physical quantities for all range of ϵ. A numerical scheme that preserves the asymptotic transitions from kinetic equations to their macroscopic limits in the numerically discrete space is called *Asymptotic Preserving* (AP) scheme [16, 17]. For numerical stability independent of ϵ, the numerical scheme that is AP usually is implicit for the discretization of $Q(f)$. Let $f^m \in \mathbb{R}^M$ be the discretized vector for $f(m\delta_t, x, v)$, where δ_t is the time step, then the general form of the scheme for a linear $Q(f) = Bf$ with B independent of f is,

$$\frac{f^{m+1} - f^m}{\delta_t} + Af^k - \frac{1}{\epsilon}B^{m+1}f^{m+1} = 0, \tag{4.19}$$

where $A, B^{m+1} \in \mathbb{R}^{M\times M}$ are constant matrices.

For the kinetic equation with uncertainty in the collision operator,

$$\partial_t f + v\partial_x f = \frac{1}{\epsilon}Q(f, \mathbf{z}), \tag{4.20}$$

for $\forall \mathbf{z} \in U$, the general form of the scheme is

$$\frac{f^{m+1}(\mathbf{z}) - f^m(\mathbf{z})}{\delta_t} + Af^m(\mathbf{z}) - \frac{1}{\epsilon}B^{m+1}(\mathbf{z})f^{m+1}(\mathbf{z}) = 0. \tag{4.21}$$

For example, the VFP equation (2.1) we considered in this paper, if one moves the forcing term $E\partial_v f$ to the RHS of the equation as is typically done in the high field regime [20], then the collision operator becomes,

$$Q(f,z) = \partial_v\left((v+E)f + \partial_v f\right).$$

That's why in the most general case, the numerical operator B depends on both $\mathbf{z}$ and t. Equivalently, (4.21) can also be written as,

$$f^{m+1}(\mathbf{z}) = \left(\frac{1}{\delta_t} - \frac{1}{\epsilon}B^{m+1}(\mathbf{z})\right)^{-1}\left(\frac{f^m(\mathbf{z})}{\delta_t} - Af^k(\mathbf{z})\right). \tag{4.22}$$

This means that in order to calculate $f(T,x,v,\mathbf{z}_\nu)$ at $T = N_t\delta_t$, one needs to invert an $\mathbb{R}^{M\times M}$ matrix for N_t times, where $M = N_x \times N_v$ with N_x, N_v being the number of grid points in x and v respectively. So for each $\mathbf{z}_\nu$, the cost is $O(N_t M^3)$. Algorithm 4.1 requires calculating $f(T,x,v,\mathbf{z}_\nu)$ for all $\mathbf{z}_\nu \in \tilde{N}(\Lambda_n)$, where the size of $\tilde{N}(\Lambda_n)$ could be $O(n^2/2)$. So the ASPI method (Algorithm 4.1) for multi-scale kinetic equations could be computationally expensive, see Section 4.3 for the total cost.

Next we will introduce an algorithm where calculating $f(T,x,v,\mathbf{z}_\nu)$ for all $\mathbf{z}_\nu \in \tilde{N}(\Lambda_n)$ can be avoided.

At step n, we already have $\Lambda_n = \{\nu_1, \cdots \nu_n\}$ and the numerical solution $f^m_{\nu_k} = f(m\delta_t, \mathbf{z}_{\nu_k})$ and $f^{m-1}_{\nu_k} = f((m-1)\delta_t, \mathbf{z}_{\nu_k})$ for $1 \le k \le n$. Let operator $\mathcal{S}$ be the numerical kinetic operator,

$$\mathcal{S}(f^m(\mathbf{z}), f^{m-1}(\mathbf{z})) = \frac{f^m(\mathbf{z}) - f^{m-1}(\mathbf{z})}{\delta_t} + A^{m-1}f^{m-1}(\mathbf{z}) - \frac{1}{\epsilon}B^m(\mathbf{z})f^m(\mathbf{z}). \tag{4.23}$$

For $f^m_{\nu_k}, f^{m-1}_{\nu_k}$ obtained from the numerical scheme, it must satisfy $\mathcal{S}(f^m_{\nu_k}, f^{m-1}_{\nu_k}) = 0$. For the interpolation approximations $I^m_{\Lambda_n}(\mathbf{z}), I^{m-1}_{\Lambda_n}(\mathbf{z})$ interpolating on $f^m_{\nu_k}, f^{m-1}_{\nu_k}$ respectively, $1 \le k \le n$,

$\mathcal{S}(I^m_{\Lambda_n}(\mathbf{z}), I^{m-1}_{\Lambda_n}(\mathbf{z}))$ represents the residual of the scheme for the polynomial interpolation $I^m_{\Lambda_n}$ at $\mathbf{z}$. So one can search for the biggest residual with respect to $\mathcal{S}(I^m_{\Lambda_n}(\mathbf{z}), I^{m-1}_{\Lambda_n}(\mathbf{z}))$ on $\nu \in \tilde{N}(\Lambda_n)$ to get ν_{n+1}. We will later see that the greedy search in this way costs less than the ASPI method. Since the interpolation $I^m_{\Lambda_n}(\mathbf{z})$ on data $f^m_{\nu_k}$ for $1 \le k \le n$ can be represented by a linear combination of $f^m_{\nu_k}$, which can be written as

$$I^m_{\Lambda_n}(\mathbf{z}) = \sum_{k=1}^{n} \gamma^k_{\Lambda_n}(\mathbf{z}) f^m_{\nu_k}, \tag{4.24}$$

where

$$\left[\gamma^1_{\Lambda_n}(\mathbf{z}) \cdots \gamma^n_{\Lambda_n}(\mathbf{z})\right] = \left[H_{\nu_1}(\mathbf{z}) \cdots H_{\nu_n}(\mathbf{z})\right] \begin{bmatrix} H_{\nu_1}(\mathbf{z}_{\nu_1}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_1}) \\ \vdots \\ H_{\nu_1}(\mathbf{z}_{\nu_n}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_n}) \end{bmatrix}^{-1}, \tag{4.25}$$

hence, plugging (4.24) into operator $\mathcal{S}$ gives

$$\begin{aligned} \mathcal{S}(I^m_{\Lambda_n}(\mathbf{z}), I^{m-1}_{\Lambda_n}(\mathbf{z})) =& \sum_{k=1}^{n} \gamma^k_{\Lambda_n} \left(\frac{f^m_{\nu_k} - f^{m-1}_{\nu_k}}{\delta_t} + A^{m-1} f^{m-1}_{\nu_k} \right) - \frac{1}{\epsilon} B^m(\mathbf{z}) \sum_{j=1}^{n} \gamma^k_{\Lambda_n} f^m_{\nu_k} \\ =& \sum_{k=1}^{n} \gamma^k_{\Lambda_n} \left(\frac{1}{\epsilon} B^m_{\nu_k} f^m_{\nu_k} \right) - \frac{1}{\epsilon} B^m(\mathbf{z}) \sum_{k=1}^{n} \gamma^k_{\Lambda_n} f^m(\mathbf{z}_{\nu_k}) \\ =& \frac{1}{\epsilon} \sum_{k=1}^{n} \gamma^k_{\Lambda_n} \left(B^m_{\nu_k} f^m_{\nu_k} - B^m(\mathbf{z}) f^m_{\nu_k} \right), \end{aligned} \tag{4.26}$$

where the second equality is because of $S(f^m_{\nu_k}, f^{m-1}_{\nu_k}) = 0$.

In addition, when calculating $\gamma_{\Lambda_n}(\mathbf{z}) = [\gamma_1(\mathbf{z}), \cdots, \gamma_n(\mathbf{z})]$ in (4.25), we don't need to invert the whole matrix on the RHS at every step. Because of the monotonicity of $\Lambda_n = \Lambda_{n-1} \cup \{\nu_n\}$, and the Schur complement of the inversion from the previous step, we can avoid computing the inversion. Specifically, define

$$H_{\Lambda_n}(\mathbf{z}_{\Lambda_n}) := \begin{bmatrix} H_{\nu_1}(\mathbf{z}_{\nu_1}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_1}) \\ \vdots \\ H_{\nu_1}(\mathbf{z}_{\nu_n}) \cdots H_{\nu_n}(\mathbf{z}_{\nu_n}) \end{bmatrix}, \quad H_{\Lambda_{n-1}}(\mathbf{z}_\nu) = [H_{\nu_1}(\mathbf{z}_\nu), \cdots, H_{\nu_{n-1}}(\mathbf{z}_\nu)], \tag{4.27}$$

then by (4.3), $H_{\Lambda_n}(\mathbf{z}_{\Lambda_n})$ can also be written in the form of block matrix,

$$H_{\Lambda_n}(\mathbf{z}_{\Lambda_n}) = \begin{bmatrix} H_{\Lambda_{n-1}}(\mathbf{z}_{\Lambda_{n-1}}) & \mathbf{0}_{(n-1)\times 1} \\ H_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}) & 1 \end{bmatrix}. \tag{4.28}$$

It is easy to check that,

$$H^{-1}_{\Lambda_n}(\mathbf{z}_{\Lambda_n}) = \begin{bmatrix} H^{-1}_{\Lambda_{n-1}}(\mathbf{z}_{\Lambda_{n-1}}) & 0 \\ -\gamma_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}) & 1 \end{bmatrix}. \tag{4.29}$$

Let

$$S^{\Lambda_n}_\nu = \sum_{k=1}^{n} \gamma^k_{\Lambda_n}(\mathbf{z}_\nu) \left(B_{\nu_k} f_{\nu_k} - B_\nu f_{\nu_k} \right) \in \mathbb{R}^M \tag{4.30}$$

be the residual of interpolation of $\mathbf{z}_\nu$ at time T, where $f_{\nu_k} = f(T, \mathbf{z}_{\nu_k})$, $B_\nu = B(T, \mathbf{z}_\nu)$, so based on this residual, we construct the following new algorithm.

Algorithm 4.2. *(RASPI)*

- *Step* 0*, Step* 1 *are the same as Algorithm 4.1.*
- *Step n. Assume we have* Λ_n*,* $\tilde{N}(\Lambda_{n-1})$ *and* $I_{\Lambda_n}(\mathbf{z})$*,*
  - *Construct* $\hat{N}(\Lambda_n)$ *by (4.13), then* $N^*(\Lambda_n)$ *can be constructed through (4.14),* $\tilde{N}(\Lambda_n)$ *through (4.15).*
  - $\nu_{n+1} = \text{argmax}_{\nu\in\tilde{N}(\Lambda_n)} \left\|S_\nu^{\Lambda_n}\right\|_V$*, where* $S_\nu^{\Lambda_n}$ *is defined in (4.30).*

Compared with Algorithm 4.1, the above algorithm is more efficient since for each $\mathbf{z} \in \tilde{N}(\Lambda_n)$, one only needs to multiply an $\mathbb{R}^{M\times M}$ matrix to a $M-$dimensional vector once. The computational cost for each $\mathbf{z}_\nu$ is $O(M^2)$, which is much less than $O(N_t M^3)$. We will compare the total computational cost of the two algorithms in details in the next section.

4.3 Computational cost

In this section, we will compare the computational costs between the ASPI (Algorithm 4.1) and the RASPI (Algorithm 4.2). In order to get an approximate solution with an error less than δ, for a first order discretization in the phase space, one needs to use $N_x = O(\delta^{-1})$, $N_v = O(\delta^{-1})$. So the explicit expression of the approximate solution $\hat{f}(T, \mathbf{z}) = I_{\Lambda_N}(\mathbf{z})$ at time T should be an M-dimensional vector where $M = N_x \cdot N_v = O(\delta^{-2})$. In the random space, according to Theorem 3.5, the best N approximation gives the error $\left\|f - \hat{f}\right\|_{L^2(V,U,d\rho)} \leq N^{-s}$, thus one requires $N = O\left(\delta^{-1/s}\right)$ to get an $O(\delta)$ error.

For the ASPI method, at the n-th step of Algorithm 4.1, one needs to do the following calculation:

⓪ From the previous steps, one has,

- $\alpha_{\nu_k} \in \mathbb{R}^M$, for all $\nu_k \in \Lambda_n$;
- $\alpha_\nu \in \mathbb{R}^M$, for all $\nu \in \tilde{N}(\Lambda_{n-1})$.
- $c_\nu = \|\alpha_\nu\| \|H_\nu\|_{L^2(V,d\rho)}$, for all $\nu \in \tilde{N}(\Lambda_{n-1})$.

① Obtain $f(T, \mathbf{z}_\nu) \in \mathbb{R}^M$ by numerical scheme (4.21), for all $\nu \in N^*(\Lambda_n)$.

② Obtain $\alpha_\nu = f(T, \mathbf{z}_\nu) - I_{\Lambda_n}(\mathbf{z}_\nu)$ for all $\nu \in N^*(\Lambda_n)$.

- To get the value of $I_{\Lambda_n}(\mathbf{z}_\nu)$, one needs to do the summation $I_{\Lambda_n}(\mathbf{z}_\nu) = \sum_{k=1}^n \alpha_{\nu_k} H_{\nu_k}(\mathbf{z}_\nu)$.

③ Obtain $c_\nu = \|\alpha_\nu\| \|H_\nu\|_{L^2(V,d\rho)}$ for all $\nu \in N^*(\Lambda_n)$ and find $\nu_{n+1} = \text{argmin}_{\nu\in\tilde{N}(\Lambda_n)} c_\nu$.

In step ①, one needs to calculate the numerical solution to the PDE at time T for all $\mathbf{z}_\nu \in N^*(\Lambda_n)$, where the size of $N^*(\Lambda_n)$ is $O(n)$. For general implicit scheme as (4.21), the computational cost to obtain $f(T, \mathbf{z}) \in \mathbb{R}^M$ is $O((M^3 + M^2)N_t)$, where M^3 comes from the inversion of matrix, M^2 comes from the multiplication of matrices, and these have to be done in each step. Therefore, the computational cost in step ① is

$$O(n \times (M^3 + M^2)N_t). \tag{4.31}$$

There are also cases where the inversion can be completed within a cost of $O(M^2)$, or the inversion only needs to be done once if B is time independent, then the computational cost of these are calculated in Remark 4.3.

In step ②, for each $\nu \in N^*(\Lambda_n)$, the computational cost to get $I_{\Lambda_n}(\mathbf{z}_\nu)$ is $O\left(nM + \sum_{k=1}^{n} \text{Cost}\{H_{\nu_k}(\mathbf{z}_\nu)\}\right)$, where the cost of $H_{\nu_k}(\mathbf{z}_\nu)$ is $O(k)$ for each ν. Hence one requires

$$O\left(n \times \left(nM + n^2\right)\right) \tag{4.32}$$

of computational operations to complete step ②.

At last, calculating $\|\alpha_\nu\| \, \|H_\nu\|$ for each $\nu \in N^*(\Lambda_n)$ requires $O(M + n\sqrt{M})$ operations. Then searching for the smallest one requires $O\left(\#\left(\tilde{N}(\Lambda_n)\right)\right)$ operations. Hence the total cost is

$$O\left(n \times \left(M + n\sqrt{M}\right) + n^2\right). \tag{4.33}$$

To sum up, the total cost at the n-th step of the ASPI method is

$$O\left(nM^3 N_t + n^2 M + n^3\right). \tag{4.34}$$

Plug in $M = O(\delta^{-2})$, and assume time $T \sim O(1)$ and $\delta_t \sim \delta_x$, so $N_t \sim O(\delta^{-1})$, hence the total computational cost at the n-th step of Algorithm 4.1 is

$$O\left(\delta^{-7} n + \delta^{-2} n^2 + n^3\right). \tag{4.35}$$

While for the RASPI method, one needs to do the following calculation at the n-th step,

⓪ From the previous steps, one has,

- $H^{-1}_{\Lambda_{n-1}}\left(\mathbf{z}_{\Lambda_{n-1}}\right)$;
- $\boldsymbol{\gamma}_{\Lambda_{n-1}}(\mathbf{z}_\nu)$, for $\nu \in \tilde{N}(\Lambda_{n-1})$ and $\nu = \nu_n$;
- $f_{\nu_k}, B_{\nu_k} f_{\nu_k}$, $B_\nu f_{\nu_k}$ for all $\nu_k \in \Lambda_n$, $\nu \in \tilde{N}(\Lambda_{n-1})$.

① Obtain $\boldsymbol{\gamma}_n(\mathbf{z}_\nu)$, for all $\nu \in \tilde{N}(\Lambda_n)$:

- Get $H^{-1}_{\Lambda_n}$ by (4.29).
- For $\nu \in \tilde{N}(\Lambda_{n-1})$, $\boldsymbol{\gamma}_{\Lambda_n}(\mathbf{z}_\nu) = [\boldsymbol{\gamma}_{\Lambda_{n-1}}(\mathbf{z}_\nu) - H_{\nu_n}(\mathbf{z}_\nu)\boldsymbol{\gamma}_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}), H_{\nu_n}(\mathbf{z}_\nu)]$.
- For $\nu \in N^*(\Lambda_n)$, $\boldsymbol{\gamma}_n(\mathbf{z}_\nu) = [H_{\Lambda_{n-1}}(\mathbf{z}_\nu) H^{-1}_{\Lambda_{n-1}} - H_{\nu_n}(\mathbf{z}_\nu)\boldsymbol{\gamma}_{\Lambda_{n-1}}(\mathbf{z}_{\nu_n}), H_{\nu_n}(\mathbf{z}_\nu)]$.

② Obtain $S^{\Lambda_n}_\nu$ by (4.30).

③ Find $\nu_{n+1} = \text{argmin}_{\nu \in \tilde{N}(\Lambda_n)} \left\|S^{\Lambda_n}_\nu\right\|^2$.

Firstly in step ①, since one already has $H^{-1}_{\Lambda_{n-1}}\left(\mathbf{z}_{\Lambda_{n-1}}\right)$ and $\boldsymbol{\gamma}_{\Lambda_{n-1}}(\mathbf{z}_\nu)$ from the previous step, one only needs to plug them in to get $H^{-1}_{\Lambda_n}$. For each $\nu \in \tilde{N}(\Lambda_{n-1})$, one needs $O(n)$ operations to get $\boldsymbol{\gamma}_n(\mathbf{z}_\nu)$. While for each $\nu \in N^*(\Lambda_n)$, one needs $O((n-1)^2 + \sum_{k=1}^{n} \text{Cost}\{H_{\nu_k}(\mathbf{z}_\nu)\})$ operations to get $\boldsymbol{\gamma}_n(\mathbf{z}_\nu)$. The total computational cost is

$$O\left((n-1)^2 \times n\right) + O\left(n \times \left((n-1)^2 + n^2\right)\right). \tag{4.36}$$

In step ②, for each $\nu \in \tilde{N}(\Lambda_{n-1})$, since one already has $B_{\nu_k} f_{\nu_k} - B_\nu f_{\nu_k}$ from the previous step, so one only needs to do the weighted sum operations given $\boldsymbol{\gamma}_{\Lambda_n}(\mathbf{z}_\nu)$, which requires

$O(nM)$ computational cost. For each $\nu \in N^*(\Lambda_n)$, one needs to calculate $B_\nu f_{\nu_k}$ first then does the summation, whose computational cost is $O(M^2 + nM)$. Therefore the total computational cost in ② is

$$O((n-1)^2 \times nM) + O(n \times (M^2 + nM)). \tag{4.37}$$

At last, Obtaining $\|S(T, \mathbf{z}_\nu)\|^2$ and finding the minimum among all $\nu \in \tilde{N}(\Lambda_n)$ requires computational cost of order

$$O(n^2 \times M + n^2). \tag{4.38}$$

To sum up, the total cost at n-th step of the RASPI is

$$O\left(n^3 M + nM^2\right). \tag{4.39}$$

Plugging in $M = O(\delta^{-2})$ gives the total cost of Algorithm 4.2 at the n-th step

$$O(n\delta^{-4} + n^3\delta^{-2}). \tag{4.40}$$

Summing (4.35) and (4.40) over $1 \le n \le O\left(\epsilon^{-1/s}\right)$, and based on the fact that $1/s \le 2$, one has

$$\text{Computational cost of ASPI} : O(\delta^{-7-2/s}), \tag{4.41}$$

$$\text{Computational cost of RASPI} : \begin{cases} O(\delta^{-2-4/s}), & \frac{1}{2} \le s \le 1 \\ O(\delta^{-4-2/s}), & s \ge 1 \end{cases} \tag{4.42}$$

The ratio of the two costs is

$$\frac{\text{Computational cost of ASPI}}{\text{Computational cost of RASPI}} = \begin{cases} O(\delta^{-5+2/s}), & \frac{1}{2} \le s \le 1, \\ O(\delta^{-3}), & s \ge 1. \end{cases} \tag{4.43}$$

From (4.43), one can see that the computational cost of the ASPI is $O(\delta^{-5+2/s})$ times that of RASPI for $s \le 1$ and $O(\delta^{-3})$ times that of the RASPI for $s \ge 1$. Since $s \ge \frac{1}{2}$, therefore the RASPI is always more efficient. In addition, the faster $\|\phi_j(x)\|_{W_x^{1,\infty}}$ decays, the more computational cost the RASPI saves.

Remark 4.3. *1. The inversion of a matrix in (4.22) does not necessarily need the cost of $O(M^3)$. For example, when the matrix is positive definite, one can invert an $\mathbb{R}^{M\times M}$ matrix by the conjugate gradient method with computational cost of $O(M^2)$. Also, when the collision operator $Q(f)$ is time independent, then the matrix B^{m+1} in the numerical method (4.22) is the same constant matrix for all m, so one only needs to invert the matrix once. In both cases, the computational cost of calculating $f(T, x, v, \mathbf{z}_\nu)$ for a specific $\mathbf{z}_\nu \in U$ is $O(M^2 N_t)$, which reduces the total computational cost of the ASPI to $O(\delta^{-5-2/s})$. Then the ratio of the two costs becomes*

$$\frac{\textit{Computational cost of ASPI}}{\textit{Computational cost of RASPI}} = \begin{cases} O(\delta^{3-2/s}), & \frac{1}{2} \le s < \frac{2}{3}, \\ O(\delta^{-(3-2/s)}), & \frac{2}{3} \le s \le 1, \\ O(\delta^{-1}), & s \ge 1. \end{cases} \tag{4.44}$$

When $\frac{1}{2} \le s < \frac{2}{3}$, the ASPI method is more efficient than the RASPI, while $s \ge \frac{2}{3}$, RASPI is still better than ASPI.

2. *Another deterministic method called Quasi Monte Carlo (QMC) is also widely used in parametric PDEs. However, in general, since the convergence rate of QMC is* $O(\frac{log(N)^d}{N})$ *[31, 1], which depends on dimensionality of the parameter, so it is not comparable in high dimension. Nevertheless, as discussed in [25, 24] for parametric elliptic equation using modified QMC method, it enjoys the same convergence rate as the best N approximation when the randomness* $\|\psi_j\| \in \ell^p$ *for* $2/3 \leq p \leq 1$*. Its performance for kinetic equations remain to be investigated.*

3. *In general whether the ASPI method can achieve the error estimates we get in Section 3 is still an open question. However, under stronger assumptions, [41] showed that a certain type of adaptive sparse grid interpolation will produce sequences of active index sets in polynomial basis function space which will give a dimension independent convergence rate.*

5 Numerical examples

In this section, we conduct some numerical experiments for the linear Vlasov-Fokker-Planck equation with random electric field $E(t, x, \mathbf{z})$,

$$\epsilon \partial_t f + v\partial_x f - E\partial_v f = \frac{1}{\epsilon}\mathcal{F}f, \quad x, v \in \Omega = (0, 2\pi) \times \mathbb{R} \tag{5.1}$$

with periodic condition on $x \in [0, 2\pi]$, and initial data

$$f(0, x, v) = \frac{\sin(x)}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + F, \tag{5.2}$$

where $F = e^{-\phi^\infty} M$ is defined in (2.8). We consider $\mathbf{z} \in [-1, 1]^{100}$, and set the electric field as,

$$E(t, x, \mathbf{z}) = \frac{\sin(x)}{2} + \sum_{j=1}^{100} E_j(t, x) z_j \tag{5.3}$$

with different choices of $E_j(t, x)$ in the experiments. We solve (5.1) by finite difference method with unified meshes in space and velocity $\delta_x = \frac{2\pi}{Nx}$ on $[0, 2\pi]$ and $\delta_v = \frac{12}{N_v}$ on $[-6, 6]$. The scheme we use here is from [21, 20]

$$\epsilon \frac{f_{i,j}^{m+1} - f_{i,j}^m}{\delta_t} + \frac{f_{i+\frac{1}{2},j}^{m+1} - f_{i-\frac{1}{2},j}^{m+1}}{\delta_x} = \frac{1}{\epsilon} P_{\mathbf{z}} \left(\frac{f_{i,j}^{m+1}}{\sqrt{M_{i,j}}} \right), \tag{5.4}$$

where $f_{i,j}^m = f(m\delta_t, i\delta_x, j\delta_v - 6)$. The transport term $v\partial_x f$ is approximated by the upwind scheme

$$f_{i+\frac{1}{2},j}^m = \frac{|v_j| + v_j}{2} f_{i,j}^m - \frac{|v_j| - v_j}{2} f_{i+1,j}^m.$$

For the other terms, since

$$\frac{1}{\epsilon}\left(\epsilon E\partial_v f + \mathcal{F}(f)\right) = \frac{1}{\epsilon}\partial_v \left(M_l \partial_v \left(\frac{f}{M_l} \right) \right)$$

with

$$M_l(\mathbf{z}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{|v - \epsilon E|^2}{2}},$$

depending on $\mathbf{z}$, so one can define an operator $P_{\mathbf{z}}(f)$ as the discretization of $\partial_v \left(M_l \partial_v \left(\frac{f}{M_l} \right) \right)$ as following,

$$P_{\mathbf{z}}(f_j) = \frac{1}{\delta_v^2} \left(\sqrt{(M_l)_{j+1}(M_l)_j} \left(\frac{f_{j+1}}{(M_l)_{j+1}} - \frac{f_j}{(M_l)_j} \right) - \sqrt{(M_l)_{j-1}(M_l)_j} \left(\frac{f_j}{(M_l)_j} - \frac{f_{j-1}}{(M_l)_{j-1}} \right) \right). \tag{5.5}$$

Since the scheme is in implicit form, for the operator above, one needs to do $N_x N_t$ times inversion of a $\mathbb{R}^{N_v \times N_v}$ matrix for each numerical solution $f(T, x, v, \mathbf{z})$ at a sample point $\mathbf{z}$. One efficient way to reduce the computational cost is to set [21]

$$g_{i,j}^m = \frac{f_{i,j}^m}{\sqrt{(M_l)_{i,j}^m}}, \tag{5.6}$$

then

$$\begin{aligned} \tilde{P}_{\mathbf{z}}(g_j) =& \frac{\sqrt{(M_l)_j}}{\delta_v^2 \sqrt{(M_l)_j}} \left(\sqrt{(M_l)_{j+1}(M_l)_j} \left(\frac{g_{j+1}}{\sqrt{(M_l)_{j+1}}} - \frac{g_j}{\sqrt{(M_l)_j}} \right) \right. \\ & \left. - \sqrt{(M_l)_{j-1}(M_l)_j} \left(\frac{g_j}{\sqrt{(M_l)_j}} - \frac{g_{j-1}}{\sqrt{(M_l)_{j-1}}} \right) \right) \\ =& \frac{\sqrt{(M_l)_j}}{\delta_v^2} \left(g_{j+1} - \left(\frac{\sqrt{(M_l)_{j+1}}}{\sqrt{(M_l)_j}} + \frac{\sqrt{(M_l)_{j-1}}}{\sqrt{(M_l)_j}} \right) g_j + g_{j-1} \right). \end{aligned} \tag{5.7}$$

In this way, scheme (5.4) becomes,

$$\epsilon \frac{g_{i,j}^{m+1} - f_{i,j}^m / \sqrt{(M_l)_{i,j}^{m+1}}}{\delta_t} + \frac{g_{i+\frac{1}{2},j}^{m+1} - g_{i-\frac{1}{2},j}^{m+1}}{\delta_x} = \frac{1}{\epsilon} \tilde{P}_{\mathbf{z}} \left(g_{i,j}^{m+1} \right), \tag{5.8}$$

therefore we can get a symmetric positive definite matrix multiplied to g_i^{m+1}, which can be inverted with less computational cost, for example, by the conjugate gradient method.

For all numerical experiments, we set $N_x = 32, N_v = 64, \delta_t = \frac{\delta_x}{8}, T = 0.1, \epsilon = 1$.

5.1 Convergence rate

We test three different time independent random electric fields in the form of (5.3), where $E_j(x)$ is given by the following functions:

$$a)\ E_j(x) = \frac{\cos(jx)}{2^j}; \quad b)\ E_j(x) = \frac{\cos(jx)}{j^2}; \quad c)\ E_j(x) = \frac{\cos(jx)}{j}. \tag{5.9}$$

Let $f(T, \mathbf{z})$ represent the numerical solution obtained from scheme (5.4), $\hat{f}(T, \mathbf{z})$ represents the approximate solution obtained by sparse polynomial interpolation on sample points Γ_{Λ_n}. Specifically, for the ASPI algorithm, we get α_{ν_n}, $1 \le n \le N$, then $\hat{f}(\mathbf{z}_i) = \sum_{n=1}^N \alpha_{\nu_n} H_{\nu_n}(\mathbf{z}_i)$. Similarly, for the RASPI algorithm, we get $H_{\Lambda_N}^{-1}(\mathbf{z}_{\Lambda_N})$ and f_{Λ_N}, then

$$[\alpha_{\nu_1}, \cdots, \alpha_{\nu_N}] = H_{\Lambda_N}^{-1}(\mathbf{z}_{\Lambda_N}) f_{\Lambda_N},$$

hence one can get $\hat{f}(T, \mathbf{z}_i) = \sum_{n=1}^N \alpha_{\nu_n} H_{\nu_n}(\mathbf{z}_i)$.

For the convergence rate, we check the mean square error defined as following,

$$\text{Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{f}(T,\mathbf{z}_i) - f(T,\mathbf{z}_i)\right\|^2_{L^2_{x,v}}} \tag{5.10}$$

with $N = 10^5$, where $\mathbf{z}_i$ is uniformly drawn from $[-1,1]^{100}$, to test the accuracy of the sparse interpolation.

The left column of Figure 2 shows how the error decays when adding sample points adaptively by the ASPI method and the RASPI method. From the numerical results one can see that both methods enjoy almost the same convergence rate. The convergence rates are different for three different electric fields. By comparing the decay rate of error for each example, one finds that if $\|E_j(x)\|_V$ decays faster, then the approximation also converges with a faster rate. We further show the algebraic decay rate s in the slope. For $E_j(x)$ given in (5.9) that decays in the order of $O\left(2^{-j}\right)$, $O\left(j^{-2}\right)$, $O\left(j^{-1}\right)$ respectively, the decay rate of the error in terms of the number of basis or number of sampling points is about $O\left(N^{-2,7}\right)$, $O\left(N^{-1.2}\right)$, $O\left(N^{-0.7}\right)$ respectively for N basis or sample points, which are all faster than the Monte Carlo method of $O\left(N^{-0.5}\right)$.

However, as stated at the end of Section 4.1, for the ASPI method, in order to get the optimal N bases, one needs to compare all possible a_ν for $\nu \in \tilde{N}(\Lambda_n)$, which involves the computation of the solution $f(T,x,v,\mathbf{z})$ at z_ν for $\nu \in \tilde{N}(\Lambda_n)$. In other words, the number of sample points used is $\#(N(\Lambda_{100}))$, which is much larger than 100. For example, it is 300, 2325, 3933 for case (a), (b), (c) respectively. By taking all of these sample points into account, the decay rate of ASPI corresponding to the number of sample points are shown in Figure 3. For each sample the decay rate for each example is $O\left(N^{-2}\right)$, $O\left(N^{-0.9}\right)$, $O\left(N^{-0.3}\right)$ respectively. In particular, for the case $E_j = O\left(j^{-1}\right)$, it converges slower than the Monte Carlo method.

The right column of Figure 2 shows the projection of the 100 selected sample points on each dimension. One finds that for all three cases, the number of projection points gets smaller as the dimension gets higher. One also notes that all the dimensions larger than 17, 66, 88 only have one projection point for three cases respectively. This indicates that when $\|E_j\|$ decay slower, then more points are projected to higher dimension.

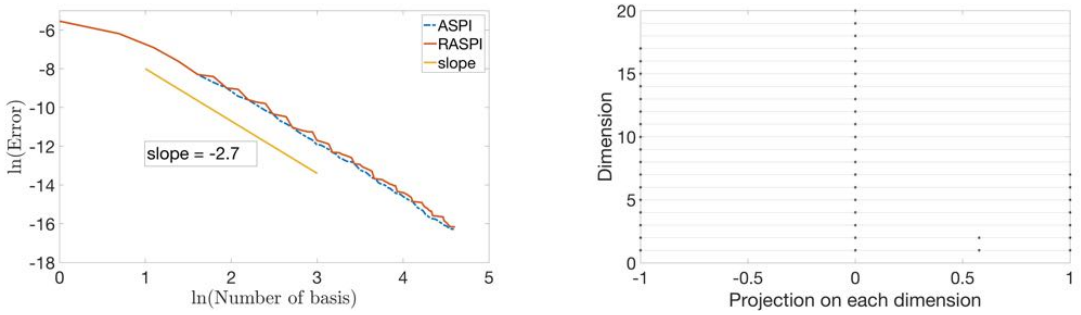
5.2 Efficiency of the greedy search

For both methods, one needs to do greedy search for $\nu \in \tilde{N}(\Lambda_n)$ at the n-th step. In Figure 4, we show that the greedy search is more efficient than just randomly choosing ν_{n+1} from $\tilde{N}(\Lambda_n)$. We call the method without greedy search the anisotropic Monte Carlo method. At the n-th step, one does the following,

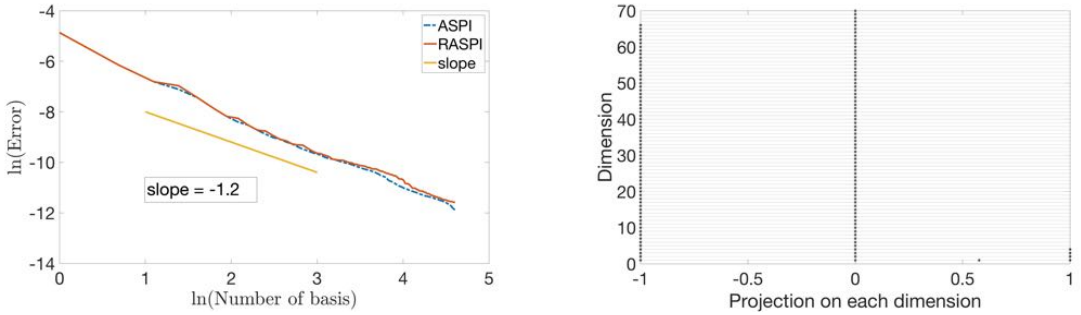
Algorithm 5.1. *(Anisotropic MC)*

- *At n-th step, one has* Λ_n, I_{Λ_n}.
  - *Construct* $\tilde{N}(\Lambda_n)$.
  - *Uniformly draw* ν_{n+1} *from* $\tilde{N}(\Lambda_n)$, *compute* $f_{\nu_{n+1}} = f(T,\mathbf{z}_{\nu_{n+1}})$, *then construct* $\alpha_{\nu_{n+1}} = f_{\nu_{n+1}} - I_{\Lambda_n}(\mathbf{z}_{\nu_{n+1}})$.

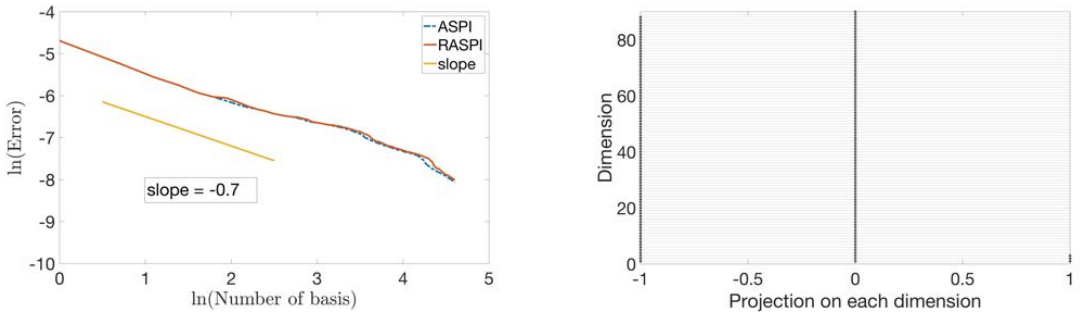
Since we have shown in Figure 2 that ASPI and the RASPI have almost the same decay rate, so only the decay rate of the RASPI is shown in Figure 4. Figure 4 shows that for the same



(a) The convergence rate for the case of $E(x, \mathbf{z}) = \dfrac{\sin(x)}{2} + \displaystyle\sum_{j=1}^{100} \frac{\cos(jx)}{2^j} z_j$

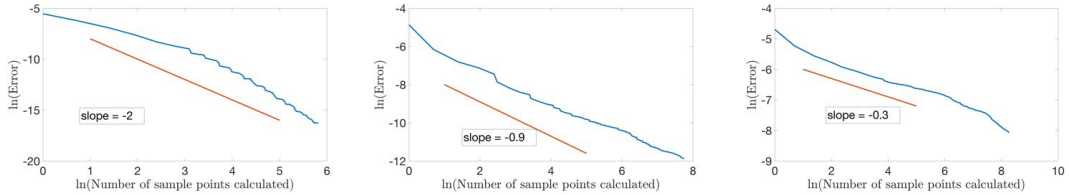


(b) The convergence rate for the case of $E(x, \mathbf{z}) = \dfrac{\sin(x)}{2} + \displaystyle\sum_{j=1}^{100} \frac{\cos(jx)}{j^2} z_j$



(c) The convergence rate for the case of $E(x, \mathbf{z}) = \dfrac{\sin(x)}{2} + \displaystyle\sum_{j=1}^{100} \frac{\cos(jx)}{j} z_j$

Figure 2: The convergence rate for error defined in (5.10) of the approximate solution to (5.1) at $t = 1$ with different E obtained by the ASPI method and the RASPI method, for $\epsilon = 1$.



(a) **(b)** **(c)**

Figure 3: The convergence rate for error defined in (5.10) of the approximate solution by the ASPI method with respect to the number of sample points actually calculated. (a), (b), (c) are the cases of different E_j in (5.9).

number of sample points one calculated, including those in the greedy search, the adaptive greedy search methods (ASPI and RASPI) have faster decay of error compared with the anisotropic Monte Carlo method. And one notices that as s becomes bigger, that is $\|\phi_j\|_{W_x^{1,\infty}}$ decays faster, the benefit one gains from the greedy search becomes more significant.

5.3 Time dependent electric field

Now, we will test some examples where the electric field is time dependent,

$$\begin{aligned} &a).\ E_j(t,x) = \frac{1}{2^j}\left(\cos(jx) + \frac{1}{(1+t)^2}\right), \\ &b).\ E_j(x) = \frac{1}{j^2}\left(\cos(jx) + \frac{1}{(1+t)^2}\right), \\ &c).\ E_j(x) = \frac{1}{j}\left(\cos(jx) + \frac{1}{(1+t)^2}\right). \end{aligned} \tag{5.11}$$

The decay rate of the RASPI is shown in Figure 5. We see the decay rate is slower than the case independent of t as shown in Figure 2, which is expected from the theoretical result. Since the decay rate in Theorem 3.5 depends on $\{\|\partial^\nu f\|\}_{\nu\in\mathcal{F}}$, and $\|\partial^\nu f\| = \|\partial^\nu h\|$, for $|\nu| > 0$. Notice that the upper bound of $\|\partial^\nu h\|$ is

$$\|\partial^\nu h(t)\|_{L^\infty(U,V)} \le Q(t)\,(|\nu|!)\,d^\nu,$$

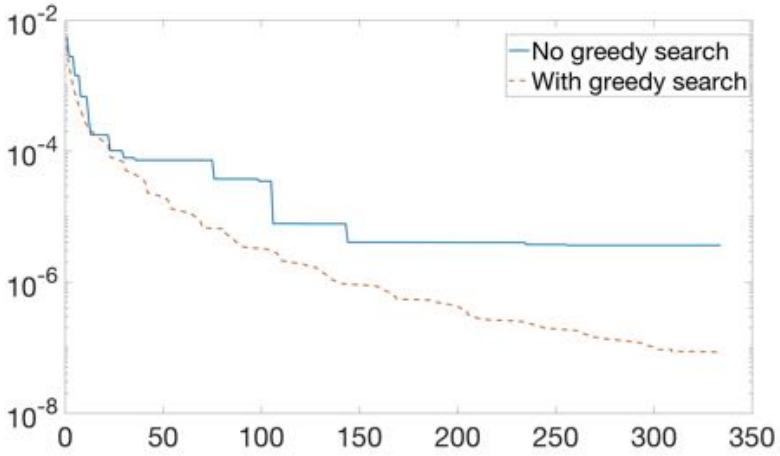
where

$$Q(t) = \min\left\{\frac{1}{\epsilon}e^{-\frac{\xi}{\epsilon^2}t}, e^{-\xi t}\right\} 2\left(\|h(0)\|_V + \sqrt{\frac{\lambda\bar{D}}{C_E}}\right).$$

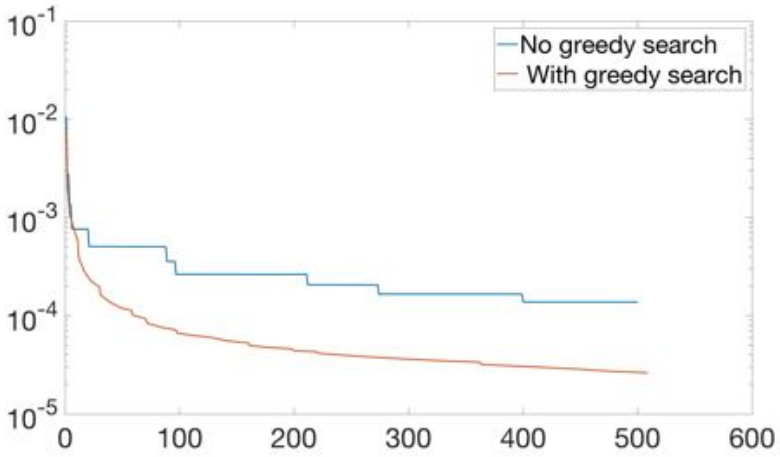
For the different cases of $\mathbb{E}_j$ in (5.9) and (5.11), since $\lim_{t\to\infty} E_j(t,x)$ in (5.11) is equal to $\mathbb{E}_j$ in (5.9), so the only difference in the upper bound of $\|\partial^\nu h\|$ is on $\bar{D}$ appearing in $Q(t)$ and defined in (3.10). Specifically, $\bar{D} = 0$ for the time independent case and $\bar{D} > 0$ for the time dependent case. So the time dependent case should have slower decay rate compared to the time independent case.

5.4 ϵ dependency

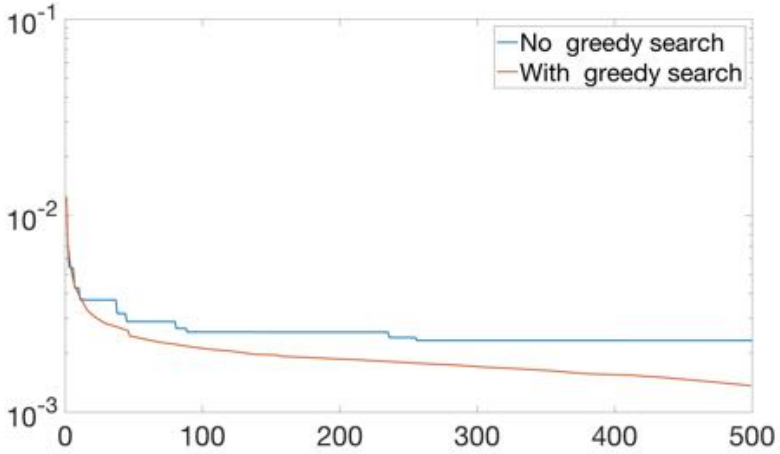
Finally, we will check the error dependence on different ϵ. The error of the RASPI method is shown in Figure 6. The decay rate for different ϵ is similar, but smaller ϵ yields smaller error.



(a) The error for the case of $E(x, \mathbf{z}) = \frac{\sin(x)}{2} + \sum_{j=1}^{100} \frac{\cos(jx)}{2^j} z_j$

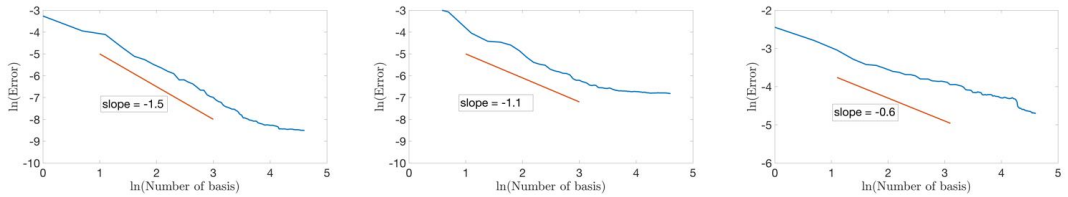


(b) The error for the case of $E(x, \mathbf{z}) = \frac{\sin(x)}{2} + \sum_{j=1}^{100} \frac{\cos(jx)}{j^2} z_j$



(c) The error for the case of $E(x, \mathbf{z}) = \frac{\sin(x)}{\frac{2}{25}} + \sum_{j=1}^{100} \frac{\cos(jx)}{j} z_j$

Figure 4: The error of the approximate solution to (5.1) at $t = 1$ with different E_j obtained by Algorithm 5.1 and the RASPI, where we set $\epsilon = 1$.



(a) **(b)** **(c)**

Figure 5: The convergence rate for error of the approximate solution by the RASPI with respect to the number of sample points. (a), (b), (c) are the cases of different E_j in (5.11).

Our explanation on this is that for smaller ϵ, the solution is closer to the global Maxwellian (2.3), which is deterministic thus the sampling error is less relevant.

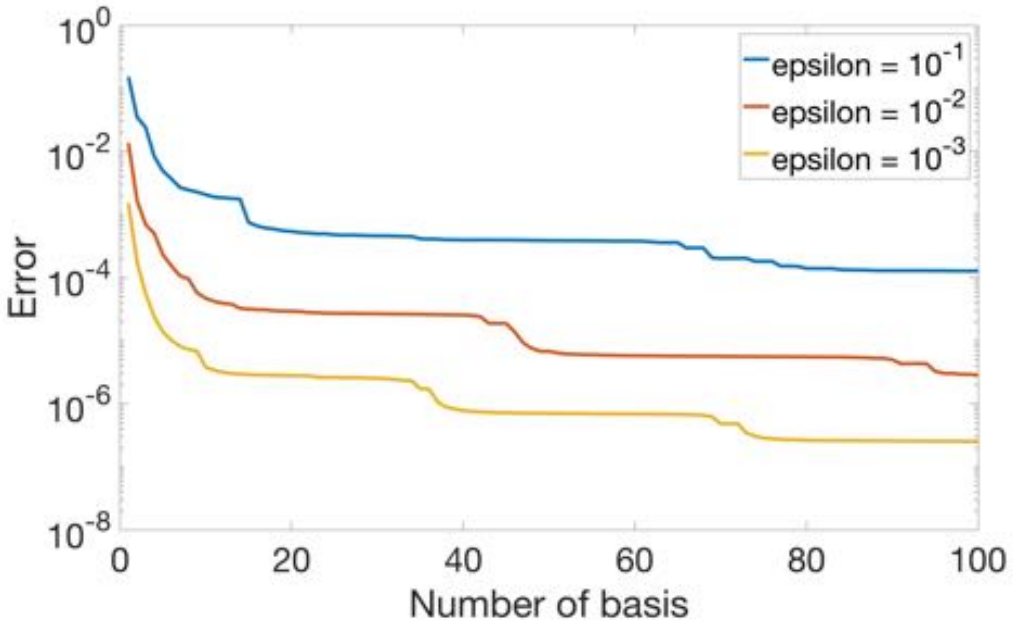


Figure 6: The error of the RASPI with respect to the number of sample points, with E_j defined in (5.9) b).

6 Conclusion

In this paper, we first showed theoretically that if the forcing term $E(t,x,z) = \bar{E}(t,x) + \sum_{j\geq} E_j(t,x)z_j$ has anisotropic property in random space, converges to the steady state $E^{\infty}(x,z)$ fast enough and is bounded above, then the best N approximation based on the Legendre basis converges to the solution of the Vlasov-Fokker-Planck equation in random space with an error of $O\left(N^{-\left(\frac{1}{p}-\frac{1}{2}\right)}\right)$, for $\left(\|E_j\|_{W^{1,\infty}}\right)_{j\geq 1} \in \ell^p$.

Numerically, we develop the residual based adaptive sparse polynomial interpolation (RASPI) method based on the adaptive sparse polynomial interpolation (ASPI). We show through nu-

merical examples that RASPI converges to the solution independent of dimension of the random variables. We also show that for general linear kinetic equation, or equivalently, for general time dependent and implicit scheme, the ratio of the computation cost of the ASPI to the RASPI is $O(\delta^{-5+2/s})$ for $\frac{1}{2} \leq s \leq 1$ and $O(\delta^{-3})$ for $s \geq 1$, which means that the faster $\|E_j\|_{W^{1,\infty}}$ decays, the more the RASPI saves.

There are still several open questions worthy of study in the future. For example, the rigorous convergence rate of the RASPI and the ASPI method remain to be established. Another important problem is whether nonlinear kinetic equations, such as the Boltzmann equation and Vlasov-Poisson-Fokker-Planck equations with high dimensional uncertain parameters, can be solved by these methods.

Acknowledgement

The first author was partially supported by NSF grants DMS-1522184, DMS-1819012, DMS-1107291: RNMS KI- Net, and NSFC grant No. 31571071. The third author has received funding of the Alexander von Humboldt-Professorship program, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement NO. 694126- DyCon9, the Air Force Office of Scientific Research under Award NO: FA9550-18-1-0242, the Grant MTM2017-92996-C2-1-R COSNET of MINECO (Spain), the ELKARTEK project KK-2018/00083 ROAD2DC of the Basque Government, and the European Unions Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement NO. 765579-ConFlex.

Appendices

A The proof of Lemma 3.7

A.1 The proof of the time independent random electric field

We will first prove the case when the random electric field is time independent, that is, $E(t,x,z) \equiv E(x,z)$.

Lemma A.1. *Under Condition 3.2, for $\forall \mathbf{z} \in U$, the following estimates hold,*

$$|\nu| = 0 : \partial_t G_i^0 + \frac{\eta}{\epsilon} \|h\|_V^2 \leq 0; \tag{A.1}$$

$$|\nu| > 1 : \partial_t G_i^\nu + \frac{\eta}{\epsilon} \|\partial^\nu h\|_V^2 \leq \frac{2}{\lambda\epsilon} \sum_{\nu_j \neq 0} \nu_j^2 C_j \left\|\partial^{\nu - e_j} h\right\|_V^2, \tag{A.2}$$

where $i = 1, 2$, $\eta = \frac{C_s}{10}$, and

$$\frac{\epsilon}{2\lambda} \|\partial^\nu h\|_V^2 \leq G_1^\nu \leq \frac{3}{2\lambda\epsilon} \|\partial^\nu h\|_V^2, \quad \frac{1}{2\lambda\epsilon} \|\partial^\nu h\|_V^2 \leq G_2^\nu \leq \frac{3}{2\lambda\epsilon} \|\partial^\nu h\|_V^2. \tag{A.3}$$

Proof. For $E(t,x,\mathbf{z}) \equiv E(x,\mathbf{z})$, the microscopic equation (2.13) for the perturbative solution $h(t,x,v,\mathbf{z})$ is simplified to,

$$\epsilon\partial_t h + v\partial_x h - \frac{1}{\epsilon}\mathcal{L}h = E\left(\partial_v - \frac{v}{2}\right) h, \tag{A.4}$$

where $\mathcal{L}$ is the linearized Fokker Planck operator defined in (2.14) that satisfies the local coercivity property as in (2.15). If one multiplies $\sqrt{M}$ and $v\sqrt{M}$ to (A.4) respectively, and integrates them over v, then one gets the *macroscopic equations*,

$$\begin{cases} \epsilon\partial_t\sigma + \partial_x u = 0 & \text{(A.5)} \\ \epsilon\partial_t u + \partial_x\sigma + \int v^2\sqrt{M}(1-\Pi)\partial_x h dv + \dfrac{1}{\epsilon}u = -E\sigma. & \text{(A.6)} \end{cases}$$

The first equation is the perturbative continuity equation, while the second one is the perturbative momentum equation. Notice the operators Π and $1-\Pi$ are perpendicular to each other in $L^2_{x,v}$, that is,

$$\|h\|^2 = \|\Pi h\|^2 + \|(1-\Pi)h\|^2 = \|\sigma\|^2 + \|(1-\Pi)h\|^2 . \tag{A.7}$$

If one takes ∂^ν and $\partial^\nu\partial_x$ to (A.4), and multiplies $\partial^\nu_{\mathbf{z}} h$ and $\partial^\nu\partial_x h$ respectively, then integrates them over x, v and adds the two equations together, one has

$$\begin{aligned} &\frac{\epsilon}{2}\partial_t \|\partial^\nu h\|_V^2 + \frac{\lambda}{\epsilon}\|(1-\Pi)\partial^\nu h\|^2_{V,\omega} \\ \le &\underbrace{\left\langle \partial^\nu\left(E\left(\partial_v - \frac{v}{2}\right)h\right), \partial^\nu h\right\rangle}_{I} + \underbrace{\left\langle \partial^\nu\partial_x\left(E\left(\partial_v - \frac{v}{2}\right)h\right), \partial^\nu\partial_x h\right\rangle}_{II}, \end{aligned} \tag{A.8}$$

where the second term II comes from the hypocoercivity of $\mathcal{L}$ in (2.15). If one takes ∂^ν to (A.6) , and multiplies $\partial^\nu\partial_x\sigma$, then integrates it over x, v, one has

$$\begin{aligned} &\partial_t\left(\epsilon\langle\partial^\nu u, \partial^\nu\partial_x\sigma\rangle + \frac{1}{2}\|\partial^\nu\sigma\|^2\right) + \frac{1}{2}\|\partial^\nu\partial_x\sigma\|^2 \\ \le &\|\partial^\nu\partial_x u\|^2 + \frac{1}{2}\|(1-\Pi)\partial^\nu h\|^2_{V,\omega} - \underbrace{\langle\partial^\nu(E\sigma), \partial^\nu\partial_x\sigma\rangle}_{III}. \end{aligned} \tag{A.9}$$

Here the first term on the LHS and the first term on the RHS come from

$$\begin{aligned} &\langle\partial_t\partial^\nu u, \partial^\nu\partial_x\sigma\rangle = \partial_t\langle\partial^\nu u, \partial^\nu\partial_x\sigma\rangle - \langle\partial^\nu u, \partial_t\partial^\nu\partial_x\sigma\rangle \\ =&\partial_t\langle\partial^\nu u, \partial^\nu\partial_x\sigma\rangle + \frac{1}{\epsilon}\langle\partial^\nu u, \partial^\nu\partial_x^2 u\rangle = \partial_t\langle\partial^\nu u, \partial^\nu\partial_x\sigma\rangle - \frac{1}{\epsilon}\|\partial^\nu\partial_x u\|^2 , \end{aligned} \tag{A.10}$$

where the second equality is because of the continuity equation (A.5). The second term on the LHS of (A.9) is because of

$$\frac{1}{\epsilon}\langle\partial^\nu u, \partial^\nu\partial_x\sigma\rangle = \left\langle -\frac{1}{\epsilon}\partial^\nu\partial_x u, \partial^\nu\sigma\right\rangle = \langle\partial_t\partial^\nu\sigma, \partial^\nu\sigma\rangle = \frac{1}{2}\partial_t\|\partial^\nu\sigma\|^2 . \tag{A.11}$$

Furthermore since,

$$\langle\partial^\nu\partial_x\sigma, \partial^\nu\partial_x\sigma\rangle + \left\langle\int v^2(1-\Pi)\partial_x\partial^\nu h\sqrt{M}\,dv, \partial^\nu\partial_x\sigma\right\rangle \ge \frac{1}{2}\|\partial^\nu\partial_x\sigma\|^2 - \frac{1}{2}\|(1-\Pi)\partial^\nu\partial_x h\|^2_\omega , \tag{A.12}$$

this gives the third term on the LHS and the second term on the RHS of (A.9).

Furthermore, integrate (A.5) over x, by the periodic boundary condition, one has,

$$\partial_t\int\sigma dx = 0.$$

Since from (2.12), we know that

$$\int \sigma(0,x,z)dx = \int h\sqrt{M}dxdv = 0,$$

so

$$\int \sigma(t,x,z)dx = \int \sigma(0,x,z)dx = 0.$$

Similar equality can be obtained for $\partial^\nu\sigma$. Therefore, one can apply the Poincare inequality (1.1) to $\|\partial_x\sigma\|^2$ to get,

$$\|\partial^\nu\partial_x\sigma\|^2 \geq C_s \|\partial^\nu\sigma\|_V^2 .$$

By adding $\theta_i(A.8) + \frac{1}{2\epsilon}(A.9)$, and the fact that

$$\|\partial^\nu\partial_x u\|^2 \leq \|(1-\Pi)\partial^\nu\partial_x h\|^2 \leq \|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 ,$$

one has,

$$\partial_t G_i^\nu + \frac{\lambda\theta_i}{\epsilon}\|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 + \frac{C_s}{4\epsilon}\|\partial^\nu\sigma\|_V^2 \leq \frac{3}{4\epsilon}\|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 + \theta_i(I+II) - \frac{1}{2\epsilon}III, \tag{A.13}$$

where C_s comes from the Poincare inequality (1.1). In order to find an estimate for G_i^ν, one needs to estimate terms $\theta_i(I+II) - \frac{1}{2\epsilon}III$. First notice that

$$\left(\partial_v + \frac{v}{2}\right)\partial^\nu\partial_x^i(\Pi h) = -\frac{v}{2}\left(\partial^\nu\partial_x^i\sigma\right)\sqrt{M} + \frac{v}{2}\left(\partial^\nu\partial_x^i\sigma\right)\sqrt{M} = 0, \quad \text{for } i = 0,1, \tag{A.14}$$

which implies that terms I and II can be simplified to, for $i = 0, 1$,

$$\begin{aligned} I,II =& -\left\langle \partial^\nu\partial_x^i(Eh), \left(\partial_v + \frac{v}{2}\right)\partial^\nu\partial_x^i h\right\rangle = -\left\langle \partial^\nu\partial_x^i(Eh), \left(\partial_v + \frac{v}{2}\right)\left(\partial^\nu\partial_x^i(\Pi h) + (1-\Pi)\partial^\nu\partial_x^i h\right)\right\rangle \\ =& -\left\langle \partial^\nu\partial_x^i(Eh), \left(\partial_v + \frac{v}{2}\right)(1-\Pi)\partial^\nu\partial_x^i h\right\rangle . \end{aligned} \tag{A.15}$$

Another inequality that will be used frequently later is that for $i = 0, 1$,

$$\begin{aligned} &\left\|\left(\partial_v + \frac{v}{2}\right)(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 \\ =& \left\|\partial_v(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 + \frac{1}{4}\left\|v(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 + \int_\Omega \frac{v}{2}\partial_v\left((1-\Pi)\partial^\nu\partial_x^i h\right)^2 dxdv \\ =& \left\|\partial_v(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 + \frac{1}{4}\left\|v(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 - \frac{1}{2}\left\|(1-\Pi)\partial^\nu\partial_x^i h\right\|^2 \leq \left\|(1-\Pi)\partial^\nu\partial_x^i h\right\|_\omega^2 . \end{aligned} \tag{A.16}$$

Based on (A.15) and (A.16), we will bound the term $\theta_i(I+II) - \frac{1}{2\epsilon}III$ for the cases $|\nu| = 0, |\nu| > 1$ respectively. Firstly, for the case $|\nu| = 0$,

$$\begin{aligned} &\theta_i(I+II) - \frac{1}{2\epsilon}III \\ =& \theta_i\left\langle Eh, \left(\partial_v + \frac{v}{2}\right)(1-\Pi)h\right\rangle + \theta_i\left\langle \partial_x(Eh), \left(\partial_v + \frac{v}{2}\right)(1-\Pi)h\right\rangle - \frac{1}{2\epsilon}\langle E\sigma, \partial_x\sigma\rangle \\ \leq& \frac{\theta_i}{2}\|E\|_{L_x^\infty}\left(\epsilon\|h\|_V^2 + \frac{1}{\epsilon}\|(1-\Pi)h\|_{V,\omega}^2\right) + \frac{\theta_i}{2}\|\partial_x E\|_{L_x^\infty}\left(\epsilon\|h\|^2 + \frac{1}{\epsilon}\|(1-\Pi)\partial_x h\|_\omega^2\right) \\ &+ \frac{1}{4\epsilon}\|E\|_{L_x^\infty}\left(\|\sigma\|^2 + \|\partial_x\sigma\|^2\right). \end{aligned} \tag{A.17}$$

Since $\|E\|_{L^\infty(U,W_x^\infty)} \le C_E$, and $\|h\|_V^2 = \|\sigma\|_V^2 + \|(1-\Pi)h\|_V^2$, one can further simplify (A.17) to

$$(A.17) \le C_E \left(\left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \right) \|\sigma\|_V^2 + \frac{\theta_i}{\epsilon} \|(1-\Pi)h\|_{V,\omega}^2 \right). \tag{A.18}$$

Plug the above estimate into (A.13), one has,

$$\begin{aligned} &\partial_t G_i^0 + \frac{\lambda\theta_i}{\epsilon} \|(1-\Pi)h\|_{V,\omega}^2 + \frac{C_s}{4\epsilon} \|\sigma\|_V^2 \\ \le &\frac{3}{4\epsilon} \|(1-\Pi)h\|_{V,\omega}^2 + C_E \left(\left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \right) \|\sigma\|_V^2 + \frac{\theta_i}{\epsilon} \|(1-\Pi)h\|_{V,\omega}^2 \right), \end{aligned} \tag{A.19}$$

which implies,

$$\partial_t G_i^0 + C_h^0 \|(1-\Pi)h\|_{V,\omega}^2 + C_\sigma^0 \|\sigma\|_V^2 \le 0, \tag{A.20}$$

where

$$C_h^0 = \frac{\lambda\theta_i}{\epsilon} - \frac{3}{4\epsilon} - \frac{C_E\theta_i}{\epsilon}, \quad C_\sigma^0 = \frac{C_s}{4\epsilon} - C_E \left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \right). \tag{A.21}$$

Since $C_E \le \frac{\lambda C_s}{8} \le \frac{\lambda}{16}$, so $\lambda - C_E \ge \frac{15}{16}\lambda$, which implies $(\lambda - C_E)\frac{\theta_i}{\epsilon} \ge \frac{4}{\lambda}$ for both $i = 1, 2$. Therefore $C_h^0 \ge \frac{3}{\epsilon}$. Since $C_s - C_E \ge \frac{7}{8}C_s$, and $\epsilon C_E\theta_i \le \frac{\epsilon C_s}{7}$ for both $i = 1, 2$, therefore $C_\sigma^0 \ge \frac{7C_s}{32\epsilon} - \frac{\epsilon C_s}{14} \ge \frac{C_s}{10\epsilon}$. Hence plug back $C_h^0, C_\sigma^0 \ge \frac{C_s}{10\epsilon}$ to (A.20), one has,

$$\partial_t G_i^0 + \frac{\eta}{\epsilon} \|h\|_{V,\omega}^2 \le 0, \quad \text{for } \eta = \frac{C_s}{10}, \tag{A.22}$$

which is exactly (A.1) in Lemma A.1.

For the case where $|\nu| > 0$, based on (A.15), (A.16) and the definition of $E = \bar{E}(x) + \sum_{j\ge1} E_j(x) z_j$, one can bound $\theta_i (I + II) - \frac{1}{2\epsilon} III$ by

$$\begin{aligned} &\theta_i (I + II) - \frac{1}{2\epsilon} III \\ \le &\theta_i \left\langle E\partial^\nu h, \left(\partial_v + \frac{v}{2}\right)(1-\Pi)\partial^\nu h \right\rangle + \theta_i \left\langle \partial_x (E\partial^\nu h), \left(\partial_v + \frac{v}{2}\right)(1-\Pi)\partial^\nu \partial_x h \right\rangle - \frac{1}{2\epsilon} \langle E\partial^\nu\sigma, \partial^\nu\partial_x\sigma \rangle \\ &+ \sum_{\nu_j \ne 0} \left(\theta_i \left\langle \nu_j E_j \partial^{\nu-e_j} h, (\partial_v + \frac{v}{2})(1-\Pi)\partial^\nu h \right\rangle + \theta_i \left\langle \partial_x \left(\nu_j E_j \partial^{\nu-e_j} h\right), (\partial_v + \frac{v}{2})(1-\Pi)\partial^\nu \partial_x h \right\rangle \right. \\ &\left. - \frac{1}{2\epsilon} \left\langle \nu_j E_j \partial^{\nu-e_j}\sigma, \partial^\nu\partial_x\sigma \right\rangle \right). \end{aligned} \tag{A.23}$$

Since the three terms in the second line of the above equation is similar to the case where $|\nu| = 0$, hence one can get similar estimates as in (A.18). In addition, by assumption $\|E_j\|_{L^\infty(U,W_x^{1,\infty})} \le C_j$, one has

$$\begin{aligned} &(A.23) \\ \le &C_E \left(\left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \right) \|\partial^\nu\sigma\|_V^2 + \frac{\theta_i}{\epsilon} \|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 \right) \\ &+ \frac{\theta_i}{2} \sum_{\nu_j \ne 0} C_j \left(\epsilon\nu_j^2 \left\|\partial^{\nu-e_j} h\right\|_V^2 + \frac{1}{\epsilon} \|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 \right) + \frac{1}{4\epsilon} \sum_{\nu_j \ne 0} C_j \left(\nu_j^2 \left\|\partial^{\nu-e_j}\sigma\right\|^2 + \|\partial^\nu\partial_x\sigma\|^2 \right) \\ \le &C_E \left(\frac{1}{2} \left(\epsilon\theta_i + \frac{1}{\epsilon} \right) \|\partial^\nu\sigma\|_V^2 + \frac{3\theta_i}{2\epsilon} \|(1-\Pi)\partial^\nu h\|_{V,\omega}^2 \right) + \sum_{\nu_j \ne 0} \nu_j^2 C_j \left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \right) \left\|\partial^{\nu-e_j} h\right\|_V^2, \end{aligned} \tag{A.24}$$

where the second inequality is because that $\sum_{j\geq 1} C_j \leq C_E$. Hence plug (A.24) into (A.13), one has,

$$\partial_t G_i^\nu + C_h^\nu \left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 + C_\sigma^\nu \left\|\partial^\nu \sigma\right\|_V^2 \leq \sum_{\nu_j \neq 0} \nu_j^2 C_j \left(\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon}\right) \left\|\partial^{\nu - e_j} h\right\|_V^2, \tag{A.25}$$

where

$$C_h^\nu = \frac{\lambda\theta_i}{\epsilon} - \frac{3}{4\epsilon} - \frac{3C_E\theta_i}{2\epsilon} \geq \frac{\eta}{\epsilon}, \quad C_\sigma^\nu = \frac{C_s}{4\epsilon} - \frac{C_E}{2}\left(\epsilon\theta_i + \frac{1}{\epsilon}\right) \geq \frac{\eta}{\epsilon} \tag{A.26}$$

and

$$\frac{\epsilon\theta_i}{2} + \frac{1}{4\epsilon} \leq \frac{2}{\lambda\epsilon}, \tag{A.27}$$

for both $i = 1, 2$, which gives (A.2) in Lemma A.1.

■

A.2 The proof of Lemma 3.7

If one takes ∂^ν and $\partial^\nu \partial_x$ to (2.13), and multiplies $\partial_{\mathbf{z}}^\nu h$ and $\partial^\nu \partial_x h$ respectively, then integrates them over x, v and adds the two equations together, one has

$$\begin{aligned}
&\frac{\epsilon}{2}\partial_t \left\|\partial^\nu h\right\|_V^2 + \frac{\lambda}{\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 \\
\leq &\left\langle \partial^\nu \left(E\left(\partial_v - \frac{v}{2}\right)h\right), \partial^\nu h\right\rangle + \left\langle \partial^\nu \partial_x \left(E\left(\partial_v - \frac{v}{2}\right)h\right), \partial^\nu \partial_x h\right\rangle, \\
&\underbrace{- \left\langle v\sqrt{M}\partial^\nu \left(\left(E - E^\infty\right)e^{-\phi^\infty}\right), \partial^\nu h\right\rangle}_{IV} \underbrace{- \left\langle v\sqrt{M}\partial^\nu \partial_x \left(\left(E - E^\infty\right)e^{-\phi^\infty}\right), \partial^\nu \partial_x h\right\rangle}_{V}.
\end{aligned} \tag{A.28}$$

If one multiplies $\sqrt{M}$ and $v\sqrt{M}$ to (2.13), and integrates it over v, then one gets

$$\begin{cases}
\epsilon\partial_t \sigma + \partial_x u = 0 & \text{(A.29)} \\
\epsilon\partial_t u + \partial_x \sigma + \int v^2\sqrt{M}(1-\Pi)\partial_x h dv + \dfrac{1}{\epsilon}u = -E\sigma - \left(E - E^\infty\right)e^{-\phi^\infty}. & \text{(A.30)}
\end{cases}$$

Then if one takes ∂^ν to (A.30), and multiplies $\partial^\nu \partial_x \sigma$, then integrates it over x, v, one has

$$\begin{aligned}
&\partial_t \left(\epsilon\left\langle \partial^\nu u, \partial^\nu \partial_x \sigma\right\rangle + \frac{1}{2}\left\|\partial^\nu \sigma\right\|^2\right) + \frac{1}{2}\left\|\partial^\nu \partial_x \sigma\right\|^2 \\
\leq &\left\|\partial^\nu \partial_x u\right\|^2 + \frac{1}{2}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 - \left\langle \partial^\nu (E\sigma), \partial^\nu \partial_x \sigma\right\rangle - \underbrace{\left\langle \partial^\nu \left(\left(E - E^\infty\right)e^{-\phi^\infty}\right), \partial^\nu \partial_x \sigma\right\rangle}_{VI}.
\end{aligned} \tag{A.31}$$

By comparing (A.28) and (A.31) with (A.8) and (A.9), we actually only need to bound terms IV, V, VI. Similar to (A.13), one has the following estimates for G_i^ν,

$$\begin{aligned}
&\partial_t G_i^\nu + \frac{\lambda\theta_i}{\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 + \frac{C_s}{4\epsilon}\left\|\partial^\nu \sigma\right\|_V^2 \\
\leq &\frac{3}{4\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 + \left[\theta_i\left(I + II\right) - \frac{1}{2\epsilon}III\right] + \left[\theta_i\left(IV + V\right) - \frac{1}{2\epsilon}VI\right].
\end{aligned} \tag{A.32}$$

First notice, for $i = 0, 1$,

$$\begin{aligned}&\left\langle v\sqrt{M}\partial^\nu\partial_x^i\left((E-E^\infty)\,e^{-\phi^\infty}\right),\partial^\nu\partial_x^i\sigma\sqrt{M}\right\rangle\\=&\left(\int vM\,dv\right)\left\langle\partial^\nu\partial_x^i\left((E-E^\infty)\,e^{-\phi^\infty}\right),\partial^\nu\partial_x^i\sigma\right\rangle=0.\end{aligned}\tag{A.33}$$

So one can break

$$\partial^\nu\partial_x^i h=(1-\Pi)\partial^\nu\partial_x^i h+\partial^\nu\partial_x^i\sigma\sqrt{M}$$

in $(IV+V)$, therefore

$$\begin{aligned}IV+V=&\sum_{i=0,1}\left\langle v\sqrt{M}\partial^\nu\partial_x^i\left((E-E^\infty)\,e^{-\phi^\infty}\right),(1-\Pi)\partial^\nu\partial_x^i h\right\rangle\\\leq&\frac{\epsilon}{2C_E}\left\|\partial^\nu\left((E-E^\infty)\,e^{-\phi^\infty}\right)\right\|_V^2+\frac{C_E}{2\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_V^2,\end{aligned}\tag{A.34}$$

For the term VI, one can bound it by

$$VI\leq\frac{1}{C_E}\left\|\partial^\nu\left((E-E^\infty)\,e^{-\phi^\infty}\right)\right\|^2+\frac{C_E}{4}\left\|\partial^\nu\partial_x\sigma\right\|^2.\tag{A.35}$$

For $|\nu|=0$, plug (A.34) and (A.35) into (A.32), and based on the estimates (A.19) we have already got in Appendices A, one has

$$\begin{aligned}&\partial_t G_i^0+\frac{\lambda\theta_i}{\epsilon}\left\|(1-\Pi)h\right\|_{V,\omega}^2+\frac{C_s}{4\epsilon}\left\|\sigma\right\|_V^2\\\leq&\frac{3}{4\epsilon}\left\|(1-\Pi)h\right\|_{V,\omega}^2+C_E\left(\left(\frac{\epsilon\theta_i}{2}+\frac{3}{8\epsilon}\right)\left\|\sigma\right\|_V^2+\frac{2\theta_i}{\epsilon}\left\|(1-\Pi)h\right\|_{V,\omega}^2\right)\\&+\frac{\epsilon\theta_i}{2C_E}\left\|(E-E^\infty)\,e^{-\phi^\infty}\right\|_V^2+\frac{1}{2\epsilon C_E}\left\|(E-E^\infty)\,e^{-\phi^\infty}\right\|^2,\end{aligned}\tag{A.36}$$

which implies,

$$\partial_t G_i^0+C_h^0\left\|(1-\Pi)h\right\|_{V,\omega}^2+C_\sigma^0\left\|\sigma\right\|_V^2\leq\frac{1}{\epsilon C_E}\left\|(E-E^\infty)\,e^{-\phi^\infty}\right\|_V^2,\tag{A.37}$$

where

$$C_h^0=\frac{\lambda\theta_i}{\epsilon}-\frac{3}{4\epsilon}-\frac{2C_E\theta_i}{\epsilon},\quad C_\sigma^0\leq\frac{C_s}{4\epsilon}-C_E\left(\frac{\epsilon\theta_i}{2}+\frac{5}{8\epsilon}\right).\tag{A.38}$$

Since $C_E\leq\frac{\lambda C_s}{8}\leq\frac{\lambda}{16}$, so $\lambda-2C_E\geq\frac{7}{8}\lambda$, which implies $(\lambda-C_E)\,\frac{\theta_i}{\epsilon}\geq 1$ for both $i=1,2$. Therefore $C_h^0\geq\frac{1}{4\epsilon}$. Since $\frac{C_s}{4}-\frac{3C_E}{8}\geq\frac{11}{64}C_s$, and $\epsilon C_E\theta_i\leq\frac{\epsilon C_s}{7}$ for both $i=1,2$, therefore $C_\sigma^0\geq\frac{11C_s}{64\epsilon}-\frac{\epsilon C_s}{14}\geq\frac{C_s}{10\epsilon}$. Hence plug back $C_h^0, C_\sigma^0\geq\frac{C_s}{10\epsilon}$ to (A.37), one has,

$$\partial_t G_i^0+\frac{\eta}{\epsilon}\left\|h\right\|_{V,\omega}^2\leq\frac{1}{\epsilon C_E}\left\|(E-E^\infty)\,e^{-\phi^\infty}\right\|_V^2,\quad\text{for }\eta=\frac{C_s}{10}.\tag{A.39}$$

For $|\nu|>0$, plug (A.34) and (A.35) into (A.32), and based on the estimates (A.24), one has

$$\begin{aligned}&\partial_t G_i^\nu+\frac{\lambda\theta_i}{\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2+\frac{C_s}{4\epsilon}\left\|\partial^\nu\sigma\right\|_V^2\\\leq&\frac{3}{4\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2+C_E\left(\left(\frac{\epsilon\theta_i}{2}+\frac{5}{8\epsilon}\right)\left\|\partial^\nu\sigma\right\|_V^2+\frac{2\theta_i}{\epsilon}\left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2\right)\\&+\frac{1}{\epsilon C_E}\left\|\partial^\nu\left((E-E^\infty)\,e^{-\phi^\infty}\right)\right\|_V^2+\frac{2}{\lambda\epsilon}\sum_{\nu_j\neq 0}\nu_j^2 C_j\left\|\partial^{\nu-e_j}h\right\|_V^2,\end{aligned}\tag{A.40}$$

which implies

$$\begin{aligned}&\partial_t G_i^\nu + C_h^\nu \left\|(1-\Pi)\partial^\nu h\right\|_{V,\omega}^2 + C_\sigma^\nu \left\|\partial^\nu \sigma\right\|_V^2 \\ \leq &\frac{2}{\lambda\epsilon}\sum_{\nu_j\neq 0} \nu_j^2 C_j \left\|\partial^{\nu-e_j} h\right\|_V^2 + \frac{1}{\epsilon C_E}\left\|\partial^\nu\left(\left(E-E^\infty\right)e^{-\phi^\infty}\right)\right\|_V^2,\end{aligned} \tag{A.41}$$

for the same C_h^ν, C_h^ν defined in (A.38). This completes the proof.

References

[1] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[2] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric pdes. *Foundations of Computational Mathematics*, 14(4):601–633, 2014.

[3] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric pdes. *Journal de Mathématiques Pures et Appliquées*, 103(2):400–428, 2015.

[4] Moulay Abdellah Chkifa. On the lebesgue constant of leja sequences for the complex unit disk and of their real projection. *Journal of Approximation Theory*, 166:176–200, 2013.

[5] Albert Cohen and Ronald DeVore. Approximation of high-dimensional parametric pdes. *Acta Numerica*, 24:1–159, 2015.

[6] Albert Cohen, Ronald DeVore, and Christoph Schwab. Convergence rates of best n-term galerkin approximations for a class of elliptic spdes. *Foundations of Computational Mathematics*, 10(6):615–646, 2010.

[7] Albert Cohen, Ronald DeVore, and Christoph Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic pde's. *Analysis and Applications*, 9(01):11–47, 2011.

[8] Esther Daus, Shi Jin, and Liu Liu. On the multi-species boltzmann equation with uncertainty and its stochastic galerkin approximation. *Preprint*, 2019.

[9] Renjun Duan, Massimo Fornasier, and Giuseppe Toscani. A kinetic flocking model with diffusion. *Communications in Mathematical Physics*, 300(1):95–145, 2010.

[10] Francis Filbet and Shi Jin. A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources. *Journal of Computational Physics*, 229(20):7625–7648, 2010.

[11] Markus Hansen and Christoph Schwab. Analytic regularity and nonlinear approximation of a class of parametric semilinear elliptic pdes. *Mathematische Nachrichten*, 286(8-9):832–860, 2013.

[12] Markus Hansen and Christoph Schwab. Sparse adaptive approximation of high dimensional parametric initial value problems. *Vietnam Journal of Mathematics*, 41(2):181–215, 2013.

[13] Víctor Hernández-Santamaría, Martin Lazar, and Enrique Zuazua. Greedy optimal control for elliptic problems and its application to turnpike problems. 2017.

[14] Jingwei Hu, Shi Jin, and Ruiwen Shu. On stochastic galerkin approximation of the nonlinear boltzmann equation with uncertainty in the fluid regime. *preprint*, 2019.

[15] Hyung Ju Hwang and Juhi Jang. On the vlasov-poisson-fokker-planck equation near maxwellian. *Discrete & Continuous Dynamical Systems-Series B*, 18(3), 2013.

[16] Shi Jin. Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations. *SIAM J. Sci. Comput.*, 21:441–454, 1999.

[17] Shi Jin. Asymptotic preserving (ap) schemes for multiscale kinetic and hyperbolic equations: a review. *Lecture Notes for Summer School on Methods and Models of Kinetic Theory(M&MKT), Porto Ercole (Grosseto, Italy). Rivista di Matematica della Universita di Parma*, 3:177–216, 2012.

[18] Shi Jin, Jian-Guo Liu, and Zheng Ma. Uniform spectral convergence of the stochastic galerkin method for the linear transport equations with random inputs in diffusive regime and a micro-macro decomposition based asymptotic preserving method. *Research in Math. Sci.,(a special issue in honor of the 70th birthday of Bjorn Engquist)*, 4:15, 2017. DOI 10.1186/s40687-017-0105-1.

[19] Shi Jin and Liu Liu. An asymptotic-preserving stochastic galerkin method for the semiconductor boltzmann equation with random inputs and diffusive scalings. *SIAM Multiscale Model. and Simult.*, 15:157–183, 2017.

[20] Shi Jin and Li Wang. An asymptotic preserving scheme for the vlasov-poisson-fokker-planck system in the high field regime. *Acta Mathematica Scientia*, 31(6):2219–2232, 2011.

[21] Shi Jin and Bokai Yan. A class of asymptotic-preserving schemes for the fokker–planck–landau equation. *Journal of Computational Physics*, 230(17):6420–6437, 2011.

[22] Shi Jin and Yuhua Zhu. Hypocoercivity and uniform regularity for the vlasov-poisson-fokker-planck system with uncertainty and multiple scales. *SIAM J. Math. Anal.*, 50:1790–1816, 2018.

[23] Angela Kunoth and Christoph Schwab. Analytic regularity and gpc approximation for control problems constrained by linear parametric elliptic and parabolic pdes. *SIAM Journal on Control and Optimization*, 51(3):2442–2471, 2013.

[24] Frances Y Kuo, Christoph Schwab, and Ian H Sloan. Quasi-monte carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, 50(6):3351–3374, 2012.

[25] Frances Y Kuo, Christoph Schwab, and Ian H Sloan. Multi-level quasi-monte carlo finite element methods for a class of elliptic pdes with random coefficients. *Foundations of Computational Mathematics*, 15(2):411–449, 2015.

[26] Martin Lazar and Enrique Zuazua. Greedy controllability of finite dimensional linear systems. *Automatica*, 74:327–340, 2016.

[27] Qin Li and Li Wang. Uniform regularity for linear kinetic equations with random input based on hypocoercivity. *SIAM/ASA J. Uncertainty Quantification*, 5(1):1193–1219, 2017.

[28] Liu Liu and Shi Jin. Hypocoercivity based sensitivity analysis and spectral convergence of the stochastic galerkin approximation to collisional kinetic equations with multiple scales and random inputs. *Multiscale Modeling & Simulation*, 16(3):1085–1114, 2018.

[29] Michel Loeve. *Probability theory, I.* Springer-Verlag New York-Heidelberg, 1978.

[30] Michel Loeve. *Probability theory, II.* Springer-Verlag New York-Heidelberg, 1978.

[31] William J Morokoff and Russel E Caflisch. Quasi-monte carlo integration. *Journal of computational physics*, 122(2):218–230, 1995.

[32] Fabio Nobile, Raul Tempone, and Clayton G Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.

[33] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[34] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1, 2007.

[35] Cl Schillings and Ch Schwab. Sparsity in bayesian inversion of parametric operator equations. *Inverse Problems*, 30(6):065007, 2014.

[36] Claudia Schillings and Christoph Schwab. Sparse, adaptive smolyak quadratures for bayesian inverse problems. *Inverse Problems*, 29(6):065011, 2013.

[37] Christoph Schwab and Andrew M Stuart. Sparse deterministic approximation of bayesian inverse problems. *Inverse Problems*, 28(4):045003, 2012.

[38] Christoph Schwab and Radu Alexandru Todor. Karhunen–loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122, 2006.

[39] Ruiwen Shu and Shi Jin. Uniform regularity in the random space and spectral accuracy of the stochastic galerkin method for a kinetic-fluid two-phase flow model with random initial inputs in the light particle regime. *ESAIM: Mathematical Modelling and Numerical Analysis*, 52(5):1651–1678, 2018.

[40] Radu Alexandru Todor. Robust eigenvalue computation for smoothing operators. *SIAM journal on numerical analysis*, 44(2):865–878, 2006.

[41] J. Zech, D. Dung, and Ch. Schwab. Multilevel approximation of parametric and stochastic pdes. Technical Report 2018-05, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2018.

[42] Yuhua Zhu. Sensitivity analysis and uniform regularity for the boltzmann equation with uncertainty and its stochastic galerkin approximation. Preprint.