Universitat Autònoma de Barcelona

Master Thesis [1]

An energy preserving discontinuous Galerkin method for Vlasov-Poisson system

Author:
Soheil Hajian

Supervisor:
Dr. B. Ayuso de Dios

Abstract. One of the simplest model problems in the kinetic theory of plasma–physics is the Vlasov-Poisson (VP) system with periodic boundary conditions. Such system describes the evolution of a plasma of charged particles (electrons and ions) under the effects of the transport and self-consistent electric field Φ_x. We consider a family of semi-discrete numerical schemes for the approximation of the Vlasov-Poisson system. The methods are based on the coupling of discontinuous Galerkin (DG) approximation to the Vlasov equation and several finite element (conforming, non-conforming and mixed) approximations for the Poisson problem. We investigate the numerical performance of all the schemes in challenging questions such as the Landau damping and two stream instability. We study and validate the conservation of physical properties such as L_p-norms, mass and total energy.

September 25, 2011

[1] part of Erasmus Mundus program (MathMods) sponsored by European Commission scholarship and partially by CRM grant.

Contents

Chapter 1

Introduction

Plasma is a kind of medium where electrons are separated from their nuclei and created a mixture of interacting charged particles. In kinetic theory, the above system is described using a statistical approach by assigning a probability density function (PDF) to the electrons, $f(t, x, v)$, which depends on time, t and the phase space (x, v). The mathematical model that describes the evolution of the system assumes that the plasma is a collision-less system where the only interaction is produced by mean-field force created by electrostatic interference and neglect electromagnetic effects. This simplified model is called the Vlasov-Poisson system which consists of a transport equation coupled with a Poisson problem. More precisely the V-P system reads

$$\begin{cases} f_t + v f_x - E(t,x) f_v = 0 \qquad x \in \Omega_x, v \in \mathbb{R} \\ \\ -\Phi_{xx} = -E_x = \rho(t,x) - 1 \end{cases} \tag{1.0.1}$$

where $E(t, x)$ is the electric field produced by electrostatic effect of ions and electrons and $\rho(t, x)$ is mass density function

$$\rho(t,x) = \int_{\mathbb{R}} f(t,x,v)\, dv \tag{1.0.2}$$

As ions are heavier than electrons, one can assume that their distribution is uniform. Assuming also plasma is in a neutral background one has

$$\int_{\Omega_x} \rho(t,x)\, dx = \int_{\Omega_x} \int_{\mathbb{R}} f(t,x,v)\, dv\, dx = 1 \tag{1.0.3}$$

There are many numerical schemes proposed for Vlasov-Poisson system. Finite volume schemes are used for hyperbolic problems and conservation laws but achieving high order solution is very difficult. On the other hand, Classical finite element (FEM) schemes can be high order approximation but they suffer from numerical oscillation when applied them to hyperbolic problems. The discontinuous Galerkin FEM (DG-FEM) is a finite element method where the continuity of the solution across each element is relaxed and hence the computations can be done locally within each element. Due to this properties *hp*-adaptivity can be achieved very easily and parallelization is built-in. Moreover inversion of the mass matrices is very cheap since they are block diagonal. One should also mention that DG-FEM requires much more degrees of freedom.

Vlasov-Poisson system has some physical properties such as energy conservation, mass conservation and L^p-norms conservation. One would like to design schemes that are able to preserve such properties. Although it is not clear if having such conservation properties at the discrete level might improve the accuracy of the scheme, it might be expected that the overall performance in the numerical simulation of physical application is enhanced.

In this thesis, we consider a family of DG schemes recently introduced in [3]. We study the actual implementation of the numerical methods. In particular, we discuss how schemes should be modified to ensure the overall efficiency of the methods. We also carry out the convergence validation of the theoretical results given in [3]. The methods are compared for the approximation of linear, non-linear Landau damping and two stream instability. Special attention is devoted to study the quality of the approximation and the ability of the schemes for preserving mass and energy. We finally present some results for the approximation of a plane diode with different boundary condition.

Chapter 2

Vlasov-Poisson system properties

We consider the Vlasov equation coupled with Poisson equation

$$\begin{cases} f_t + v f_x - E(t,x)\, f_v = 0 & x \in \Omega_x := [x_l, x_r], v \in \mathbb{R} \\ \\ -\Phi_{xx} = -E_x = \rho(t,x) - 1 & \\ \\ f(t,0,v) = f(t,1,v) & \text{periodic boundary condition in } x, v \in \mathbb{R} \\ f(0,x,v) = f_0(x,v) & \text{initial data}, supp(f_0(.,v)) \subset \Omega_v := [v_d, v_u], x \in \Omega_x \end{cases} \tag{2.0.1}$$

Here the first equation is the Vlasov PDE where $f(t,x,v)$ is a distribution function (non-negative) which depends on $t \in \mathbb{R}^+, (x,v) \in \Omega_x \times \mathbb{R}$ and is periodic with respect to x. According to [**?**] if the initial data, f_0 has compact support with respect to v then the solution, f will remain compact, (in other words f is zero for large speeds). Moreover, one may note that the independent variables are (x,v) and hence the problem is already in 2D. We start by writing the weak formulation of the (2.0.1) by introducing the space of functions $C_0^\infty(\Omega_x \times \Omega_v)$ and the test function

$$\varphi \in C_0^\infty(\Omega_x \times \Omega_v);\ \text{which ‘0’ here means periodic in } x \text{ and compact w.r.t. } v.$$

Multiplying the PDE and integrating with respect to both x and v, we will have

$$\begin{aligned} 0 &= \underbrace{\int_{\Omega_x}\int_{\mathbb{R}} f_t\,\varphi\,dx\,dv}_{A(t)} + \int_{\Omega_x}\int_{\mathbb{R}} v\,f_x\,\varphi\,dv\,dx - \int_{\Omega_x}\int_{\mathbb{R}} E(x)\,f_v\,\varphi\,dv\,dx \qquad (2.0.2) \\ &= A(t) + \int_{\Omega_x}\int_{\Omega_v} v\,f_x\,\varphi\,dv\,dx - \int_{\Omega_x}\int_{\Omega_v} E(x)\,f_v\,\varphi\,dv\,dx \\ &= A(t) + \int_{\Omega_v} v\left(\int_{\Omega_x} f_x\,\varphi\,dx\right) dv - \int_{\Omega_x} E(x)\left(\int_{\Omega_v} f_v\,\varphi\,dv\right) dx \\ &= A(t) + \left(\int_{\Omega_v} v\,\underbrace{[f\varphi]_{\partial\Omega_x}}_{=0}\,dv - \int_{\Omega_v}\int_{\Omega_x} v\,f\varphi_x dx\,dv\right) \\ &\quad - \left(\int_{\Omega_x} E(x)\,\underbrace{[f\varphi]_{\partial\Omega_v}}_{=0}\,dx - \int_{\Omega_v}\int_{\Omega_x} E(x) f\varphi_v dx\,dv\right) \\ 0 &= \int_{\Omega_x}\int_{\Omega_v} f_t\,\varphi\,dx\,dv - \int_{\Omega_x}\int_{\Omega_v} v f\varphi_x dv\,dx + \int_{\Omega_x}\int_{\Omega_v} E(x) f\varphi_v dv\,dx \quad \forall\varphi \in C_0^\infty(\Omega_x \times \Omega_v) \end{aligned}$$

We now check some other physical properties of V-P system: mass and energy conservations as well as L^2 conservation.

2.1 Mass conservation

The mass conservation reads:

$$\int_{\Omega_x}\int_{\Omega_v} f(t,x,v)dv\,dx = \int_{\Omega_x}\int_{\Omega_v} f_0(x,v)dv\,dx \quad \forall t>0$$

Proof. Take φ as

$$\varphi(x,v) = \begin{cases} 1 & v\in\Omega_v, x\in\Omega_x \\ 0 & v\in\mathbb{R}\setminus\Omega_v, x\in\Omega_x \end{cases}$$

then we will have:

$$\begin{aligned} 0 &= \int_{\Omega_x}\int_{\Omega_v} f_t\,1\,dx\,dv - \underbrace{\int_{\Omega_x}\int_{\Omega_v} v\,f\,\varphi_x\,dv\,dx}_{=0} + \underbrace{\int_{\Omega_x}\int_{\Omega_v} E(x)\,f\,\varphi_v\,dv\,dx}_{=0} \\ 0 &= \frac{d}{dt}\int_{\Omega_x}\int_{\Omega_v} f\,dx\,dv \end{aligned}$$

and hence

$$\int_{\Omega_x}\int_{\Omega_v} f(t,x,v)\,dv\,dx = \int_{\Omega_x}\int_{\Omega_v} f_0(x,v)\,dv\,dx \quad \forall t>0$$

□

2.2 L^2-conservation

$$\int_{\Omega_x}\int_{\Omega_v} |f(t,x,v)|^2\,dv\,dx = \int_{\Omega_x}\int_{\Omega_v} |f_0(x,v)|^2\,dv\,dx \quad \forall t>0$$

Proof. Take $\varphi = f$ and then

$$\begin{aligned} 0 &= \int_{\Omega_x}\int_{\Omega_v} f_t\,f\,dx\,dv - \int_{\Omega_x}\int_{\Omega_v} v\,f\,f_x\,dv\,dx + \int_{\Omega_x}\int_{\Omega_v} E(x)\,f\,f_v\,dv\,dx \\ 0 &= \frac{d}{dt}\int_{\Omega_x}\int_{\Omega_v} |f|^2\,dx\,dv - \frac{1}{2}\int_{\Omega_v} v\left(\int_{\Omega_x}(f^2)_x\,dx\right)dv + \frac{1}{2}\int_{\Omega_x}\int_{\Omega_v} E(x)\,(f^2)_v\,dv\,dx \\ 0 &= \frac{1}{2}\frac{d}{dt}\int_{\Omega_x}\int_{\Omega_v} |f|^2dx\,dv - \frac{1}{2}\int_{\Omega_v} v\,\underbrace{f^2\big|_{\partial\Omega_x}}_{=0}dv + \frac{1}{2}\int_{\Omega_x} E(x)\,\underbrace{f^2\big|_{\partial\Omega_v}}_{=0}dx \\ 0 &= \frac{d}{dt}\int_{\Omega_x}\int_{\Omega_v} |f|^2dx\,dv \end{aligned}$$

where $f^2\big|_{\partial\Omega_x}$ is zero because of periodicity of f in x and $f^2\big|_{\partial\Omega_v}$ is zero because of compact support of f with respect to v.

$$\int_{\Omega_x}\int_{\Omega_v} |f(t,x,v)|^2dv\,dx = \int_{\Omega_x}\int_{\Omega_v} |f_0(x,v)|^2dv\,dx \quad \forall t>0$$

□

2.3 Energy conservation

Defining the mass flux

$$j(t,x) := \int_{\mathbb{R}} v\, f(t,x,v)\, dv \tag{2.3.1}$$

and integrating once w.r.t. v the Vlasov equation (2.0.1), we will have the continuity equation

$$\partial_t \rho(t,x) + \partial_x j(t,x) = 0 \tag{2.3.2}$$

now if one take ∂_t from the Poisson equation in (2.0.1), we will get

$$\begin{aligned} \partial_t E_x &= -\partial_t \rho \\ \partial_t E_x &= \partial_x j(t,x) \\ \int \partial_t E_x \, dx &= \int \partial_x j(t,x)\, dx \\ \partial_t E &= j(t,x) \end{aligned} \tag{2.3.3}$$

moreover the energy measures of the V-P system are

$$\begin{aligned} &\mathcal{E}_k(t) = \int_{\Omega_x} \int_{\Omega_v} \frac{|v|^2}{2} f(t,x,v) dv\, dx && \text{kinetic energy} \\ &\mathcal{E}_p(t) = \frac{1}{2} \int_{\Omega_x} |E(t,x)|^2 dx && \text{potential energy} \\ &\mathcal{E}_{tot}(t) = \mathcal{E}_k(t) + \mathcal{E}_p(t) && \text{total energy} \end{aligned}$$

where the energy conservation reads

$$\mathcal{E}_{tot}(t) = \mathcal{E}_{tot}(0) \quad \forall t > 0$$

Proof. Take $\varphi = \frac{|v|^2}{2}$ and then

$$\begin{aligned} 0 &= \int_{\Omega_x} \int_{\Omega_v} f_t \frac{|v|^2}{2}\, dx\, dv - \underbrace{\int_{\Omega_x} \int_{\Omega_v} v\, f \left(\frac{|v|^2}{2}\right)_x dv\, dx}_{=0} + \int_{\Omega_x} \int_{\Omega_v} E(t,x)\, f \left(\frac{|v|^2}{2}\right)_v dv\, dx \\ 0 &= \frac{d}{dt} \int_{\Omega_x} \int_{\Omega_v} f \frac{|v|^2}{2}\, dx\, dv + \int_{\Omega_x} E(t,x) \int_{\Omega_v} f\, v\, dv\, dx \\ 0 &= \frac{d}{dt} \int_{\Omega_x} \int_{\Omega_v} f \frac{|v|^2}{2}\, dx\, dv + \int_{\Omega_x} E(t,x)\, j(t,x)\, dx \quad \text{substituting (2.3.3)} \\ 0 &= \frac{d}{dt} \left(\int_{\Omega_x} \int_{\Omega_v} f \frac{|v|^2}{2}\, dx\, dv + \int_{\Omega_x} \frac{E(t,x)^2}{2}\, dx \right) \end{aligned}$$

□

Chapter 3

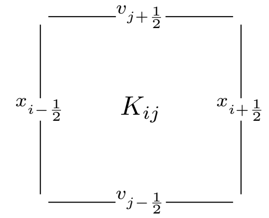
Discontinuous Galerkin Method

3.1 Vlasov equation

In this section, we introduce the discontinuous Galerkin finite element method (DG-FEM) for the V-P system. The method is different from the conforming finite element in the sense that the solutions on the boundaries of each element are not necessarily continuous and hence we have more degree of freedom compare to the FEM. Moreover, due to this discontinuity across the element boundaries the computations will be done locally inside each element (instead of whole domain in the FEM).

In order to formulate the DG-FEM for the Vlasov equation, we introduce some notation that will be used. For the first step we need to introduce the mesh (here we use a rectangular grid). Consider the coordinate of elements boundaries as

$$\begin{array}{c} x_l = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{i+\frac{1}{2}} < x_{N_x+\frac{1}{2}} = x_r \\ v_d = v_{\frac{1}{2}} < v_{\frac{3}{2}} < \ldots < v_{j+\frac{1}{2}} < v_{N_v+\frac{1}{2}} = v_u \end{array}$$



and in an element K_{ij}

$$\begin{array}{ll} I_i := [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] & \forall i = 1..N_x \\ J_j := [v_{j-\frac{1}{2}}, v_{j+\frac{1}{2}}] & \forall j = 1..N_v \\ K_{ij} := I_i \times J_j & \forall i, j \end{array} \tag{3.1.1}$$

We define finite element space

$$V_h \quad := \quad \{w | w \in L^2(\Omega_x \times \Omega_v), w|_{K_{ij}} \in \mathcal{Q}^k(K_{ij}) \quad \forall i, j\} \tag{3.1.2}$$

Furthermore, we define average and jump operators

$$\{\{f_{i+1/2}\}\} \quad := \quad \frac{f^+_{i+1/2} + f^-_{i+1/2}}{2} \quad \text{(average)} \tag{3.1.3}$$

$$[\![f_{i+1/2}]\!] \quad := \quad f^+_{i+1/2} - f^-_{i+1/2} \quad \text{(jump)} \tag{3.1.4}$$

where

$$\begin{aligned} f^+_{i+1/2} &:= f(x^+_{i+1/2}, .) \\ f^-_{i+1/2} &:= f(x^-_{i+1/2}, .) \end{aligned}$$

We multiply (2.0.1) by $\varphi \in V_h$ and integrate by parts in an element K_{ij}

$$\begin{aligned} 0 &= \underbrace{\int_{K_{ij}} f^h_t \, \varphi \, dx \, dv}_{A(t)} + \int_{K_{ij}} v \, f^h_x \, \varphi \, dv \, dx - \int_{K_{ij}} E_h(x) \, f^h_v \, \varphi \, dv \, dx \\ &= A(t) + \int_{J_j} v \left(\int_{I_i} f^h_x \, \varphi \, dx \right) dv - \int_{I_i} E_h(x) \left(\int_{J_j} f^h_v \, \varphi \, dv \right) dx \\ &= A(t) + \left(\int_{J_j} \underbrace{v \;\; f^h \varphi \big|_{\partial I_i}}_{?} dv - \int_{J_j} \int_{I_i} v \, f^h \, \varphi_x \, dx \, dv \right) \qquad (3.1.5) \\ & \quad - \left(\int_{I_i} \underbrace{E_h(x) \;\; f^h \varphi \big|_{\partial J_j}}_{?} dx - \int_{J_j} \int_{I_i} E_h(x) \, f^h \, \varphi_v \, dx \, dv \right) \end{aligned}$$

Now by defining value of f^h at the boundaries as indicated by “?”, we would determine the DG method of our interest so that the resulting method is L^2 stable.

Proposition 1. *Consider the Vlasov equation described in* (2.0.1) *and weak form* (3.1.5)*. Define the numerical fluxes as*

$$\widehat{vf}(x_{i+\frac{1}{2}}, v) := \begin{cases} v \, f^h(x^-_{i+\frac{1}{2}}, v) & v \geq 0 \\ v \, f^h(x^+_{i+\frac{1}{2}}, v) & v \leq 0 \end{cases} \qquad (3.1.6)$$

$$\widehat{Ef}(x, v_{j+\frac{1}{2}}) := \begin{cases} E_h(x) \, f^h(x, v^+_{j+\frac{1}{2}}) & E_h(x) \geq 0 \\ E_h(x) \, f^h(x, v^-_{j+\frac{1}{2}}) & E_h(x) \leq 0 \end{cases} \qquad (3.1.7)$$

then the numerical solution is L^2 stable

$$\| f^h(t, ., .) \|_{L^2} \leq \| f_0(., .) \|_{L^2} \quad \forall t > 0$$

Proof. In order to prove the above proposition we set the test function to be $\varphi \equiv f$ in (3.1.5). hence the term $A(t)$ will be

$$\begin{aligned} \int_{K_{ij}} f_t \varphi dx \, dv &= \int_{K_{ij}} f_t f \, dx \, dv \\ &= \int_{K_{ij}} \frac{1}{2} (f^2)_t dx \, dv \\ &= \frac{1}{2} \frac{d}{dt} \int_{K_{ij}} f^2 dx \, dv \\ &= \frac{1}{2} \frac{d}{dt} \| f(t, ., .) \|^2_{L^2(K_{ij})} \end{aligned}$$

Carrying on with the other terms of (3.1.5), we will get

$$
\begin{aligned}
-\frac{1}{2}\frac{d}{dt}\|f\|^2 \quad = \quad & \int_{K_{ij}} E_h(x)\frac{(f^2)_v}{2}dx\,dv - \int_{K_{ij}} v\frac{(f^2)_x}{2}dx\,dv \\
& + \int_{J_j}\left(\widehat{vf}(x_{i+\frac{1}{2}},v)f^-_{i+\frac{1}{2},\cdot} - \widehat{vf}(x_{i-\frac{1}{2}},v)f^+_{i-\frac{1}{2},\cdot}dv\right) \\
& - \int_{I_i}\left(\widehat{Ef}(x,v_{j+\frac{1}{2}})f^-_{\cdot,j+\frac{1}{2}} - \widehat{Ef}(x,v_{j-\frac{1}{2}})f^+_{\cdot,j-\frac{1}{2}}\right)dx
\end{aligned}
$$

where for simplicity of notation we set $\widehat{Ef}_{\cdot,j+\frac{1}{2}} := \widehat{Ef}(x,v_{j+\frac{1}{2}})$ and $\widehat{vf}_{i-\frac{1}{2},\cdot} := \widehat{vf}(x_{i-\frac{1}{2}},v)$. Integrating first terms and collecting all terms gives

$$
\begin{aligned}
-\frac{1}{2}\frac{d}{dt}\|f\|^2 \quad = \quad & \int_{I_i}\left[\frac{E_h(x)}{2}(f^-_{j+\frac{1}{2}})^2 - (f^+_{j-\frac{1}{2}})^2\right]dx \\
& - \int_{J_j}\frac{v}{2}\left[(f^-_{i+\frac{1}{2}})^2 - (f^+_{i-\frac{1}{2}})^2\right]dv \\
& + \int_{J_j}\left[\widehat{vf}_{i+\frac{1}{2},\cdot}\, f^-_{i+\frac{1}{2}} - \widehat{vf}_{i-\frac{1}{2},\cdot}\, f^+_{i-\frac{1}{2}}dv\right] \\
& - \int_{I_i}\left[\widehat{Ef}_{\cdot,j+\frac{1}{2}}\, f^-_{j+\frac{1}{2}} - \widehat{Ef}_{\cdot,j-\frac{1}{2}}\, f^+_{j-\frac{1}{2}}\right]dx
\end{aligned}
$$

We introduce the auxiliary expressions

$$
\widetilde{F}_{i+\frac{1}{2},\cdot} \quad := \quad \int_{J_j}\left[\frac{v}{2}(f^-_{i+\frac{1}{2}})^2 - \widehat{vf}_{i+\frac{1}{2},\cdot}\, f^-_{i+\frac{1}{2}}\right]dv \tag{3.1.8}
$$

$$
\widetilde{G}_{\cdot,j+\frac{1}{2}} \quad := \quad \int_{I_i}\left[\frac{E_h(x)}{2}(f^-_{j+\frac{1}{2}})^2 - \widehat{Ef}_{\cdot,j+\frac{1}{2}}\, f^-_{j+\frac{1}{2}}\right]dx \tag{3.1.9}
$$

and so we can rewrite our identity as

$$
-\frac{1}{2}\frac{d}{dt}\|f\|^2 \quad = \quad \widetilde{G}_{\cdot,j+\frac{1}{2}} - \widetilde{G}_{\cdot,j-\frac{1}{2}} - \left(\widetilde{F}_{i+\frac{1}{2},\cdot} - \widetilde{F}_{i-\frac{1}{2},\cdot}\right)
$$

$$
+\int_{I_i}\widehat{Ef}_{\cdot,j-\frac{1}{2}}\left[f^+_{j-\frac{1}{2}} - f^-_{j-\frac{1}{2}}\right]dx \tag{3.1.10}
$$

$$
-\int_{J_j}\widehat{vf}_{i-\frac{1}{2},\cdot}\left[f^+_{i-\frac{1}{2}} - f^-_{i-\frac{1}{2}}\right]dv \tag{3.1.11}
$$

$$
+\int_{I_i}\frac{E_h(x)}{2}\left[(f^+_{j-\frac{1}{2}})^2 - (f^-_{j-\frac{1}{2}})^2\right]dx \tag{3.1.12}
$$

$$
-\int_{J_j}\frac{v}{2}\left[(f^+_{i-\frac{1}{2}})^2 - (f^-_{i-\frac{1}{2}})^2\right]dv \tag{3.1.13}
$$

Now considering the terms (3.1.10) to (3.1.13), we can choose the $\widehat{Ef}$ and $\widehat{vf}$ such that those terms become positive. Following [3], we define

$$
\begin{aligned}
\widehat{vf}_{i+\frac{1}{2},\cdot} \quad &:= \quad \{\{f_{i+\frac{1}{2}}\}\} - \frac{\texttt{sign}[v]}{2}\llbracket f_{i+\frac{1}{2}}\rrbracket = \begin{cases} v\,f(x^-_{i+\frac{1}{2}},v) & v \geq 0 \\ v\,f(x^+_{i+\frac{1}{2}},v) & v \leq 0 \end{cases} \\
\widehat{Ef}_{\cdot,j+\frac{1}{2}} \quad &:= \quad \{\{f_{j+\frac{1}{2}}\}\} + \frac{\texttt{sign}[E_h(x)]}{2}\llbracket f_{j+\frac{1}{2}}\rrbracket = \begin{cases} E_h(x)\,f(x,v^+_{j+\frac{1}{2}}) & E_h(x) \geq 0 \\ E_h(x)\,f(x,v^-_{j+\frac{1}{2}}) & E_h(x) \leq 0 \end{cases}
\end{aligned}
$$

then the sum of terms (3.1.10) to (3.1.13) will be positive and hence

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\|f\|^2 + \widetilde{G}_{.,j+\frac{1}{2}} - \widetilde{G}_{.,j-\frac{1}{2}} - \left(\widetilde{F}_{i+\frac{1}{2},.} - \widetilde{F}_{i-\frac{1}{2},.}\right) &\leq 0 \quad \forall i,j \\
\sum_{i,j}\left(\frac{1}{2}\frac{d}{dt}\|f\|^2_{L^2(K_{ij})} + \widetilde{G}_{.,j+\frac{1}{2}} - \widetilde{G}_{.,j-\frac{1}{2}} - \left(\widetilde{F}_{i+\frac{1}{2},.} - \widetilde{F}_{i-\frac{1}{2},.}\right)\right) &\leq 0 \\
\frac{1}{2}\frac{d}{dt}\|f\|^2_{L^2(\Omega_x\times\Omega_v)} + \underbrace{\widetilde{G}_{.,N_v+\frac{1}{2}} - \widetilde{G}_{.,\frac{1}{2}}}_{\equiv 0} - \underbrace{\left(\widetilde{F}_{N_x+\frac{1}{2},.} - \widetilde{F}_{\frac{1}{2},.}\right)}_{\equiv 0} &\leq 0
\end{aligned}$$

hence

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\|f\|^2_{L^2(\Omega_x\times\Omega_v)} &\leq 0 \\
\|f(t,.,.)\|^2_{L^2(\Omega_x\times\Omega_v)} &\leq \|f_0(.,.)\|^2 \quad \forall t > 0
\end{aligned}$$

Note that

$$\widetilde{G}_{.,N_v+\frac{1}{2}} - \widetilde{G}_{.,\frac{1}{2}} \text{ and } \widetilde{F}_{N_x+\frac{1}{2},.} - \widetilde{F}_{\frac{1}{2},.}$$

vanish because of compact support in Ω_v and periodicity of the solution at the boundaries of Ω_x. □

3.2 Poisson equation

In the 1D V-P system we have Vlasov equation in 2D and Poisson equation in 1D, hence we have to consider a Poisson solver for 1D. More precisely we have to solve

$$-\Phi_{xx} = \rho(t,x) - 1 \quad x \in \Omega_x \tag{3.2.1}$$

but actually we need the first derivative of Φ, $E(t,x)$, in the Vlasov equation

$$\Phi_x = E(t,x) \tag{3.2.2}$$

Hence we consider methods that approximate the Poisson problem but also give an approximation to its first derivative. In the following subsections we discuss three different methods for this purpose. First in §3.2.1 we discuss about integrating (3.2.1) instead of solving Poisson since we just need Φ_x and we are solving a 1D problem. In §3.2.2, we introduce a DG method for solving a system of first order ODEs instead of a second order equation. Later on in §3.2.3, we consider a mixed finite element method with less degrees of freedom that gives conforming electric field and a discontinues Φ. All of these methods are discussed in [3] in detail.

3.2.1 Direct integration for Poisson equation

The first approach to the Poisson equation that we illustrate here is a simple integrator for the Poisson equation since we just need $E(t,x) = \Phi_x$ for Vlasov equation (hence it is not suitable for 4D V-P system). Note that in this approach $E_h \in \mathcal{P}^{k+1}(\Omega_x)$, so by integrating the Poisson equation, we will obtain

$$\Phi_h(t,x) = C_E^h\, x + \frac{x^2}{2} - \int_0^x \int_0^s \rho_h(t,z)\, dz\, ds \quad x \in [0,1] \tag{3.2.3}$$

where

$$C_E^h = \int_0^1 \int_0^s \rho_h(t,s)\, ds\, dz - \frac{1}{2} \tag{3.2.4}$$

$$E_h(t,x) = C_E^h + x - \int_0^x \rho_h(t,s)\, ds \tag{3.2.5}$$

where in order to calculate

$$\int_0^x \rho_h(t,s)\,ds$$

we first express the ρ_h by polynomials of degree $k+1$ (note that ρ_h is a polynomial of degree k) since $E_h \in \mathcal{P}^{k+1}(\Omega_x)$. Denoting N_{Gauss} the Gauss quadrature points of a numerical integrator, we know that it can evaluate an integrand of degree $\leq 2N_{Gauss} - 1$ exactly. Hence we choose

$$N_{Gauss} = \frac{k+2}{2}$$

for integrating.

3.2.2 Mixed-FEM for Poisson equation

In this section we introduce a method to approximate Φ and $E(t,x)$ simultaneously. Before that we simplify the notation by changing the right-hand side of Poisson equation with $g(t,x) = \rho(t,x) - 1$. The idea now is to re-write the second order problem as a system of first order equations

$$-\Phi_{xx} = g \Longrightarrow \begin{cases} E = \Phi_x \\ E_x = -g \end{cases} \tag{3.2.6}$$

where $E(t,x)$ and Φ are unknowns. Next step is to write the weak formulation of this system and define the spaces for approximating $E(t,x)$ and Φ. We introduce the following finite element spaces

$$\begin{aligned} \Phi_h \in V_h &:= \left\{ w | w \in L^2(\Omega_x),\, w|_{I_i} \in \mathcal{P}^k(I_i) \right\} \quad \text{(same as before)} && (3.2.7) \\ E_h \in \Sigma_h &:= \left\{ \sigma | \sigma \in H^1(\Omega_x),\, \sigma|_{I_i} \in \mathcal{P}^{k+1}(I_i) \right\}. && (3.2.8) \end{aligned}$$

The weak formulation reads of (3.2.6): find $(E_h, \Phi_h) \in \Sigma_h \times V_h$ that satisfy

$$\begin{aligned} \int_{\Omega_x} E_h\,\tau + \int_{\Omega_x} \Phi_h\,\tau_x &= 0 \qquad \forall \tau \in \Sigma_h && (3.2.9) \\ -\int_{\Omega_x} (E_h)_x\,\nu &= \int_{\Omega_x} g\,\nu \quad \forall \nu \in V_h && (3.2.10) \end{aligned}$$

next we express Φ_h, E_h in terms of basis functions of V_h, Σ_h respectively. Notice that E_h belongs to the space of conforming polynomials while Φ_h is discontinuous. In fact we have a saddle point system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{E}_h \\ \mathbf{\Phi}_h \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} \tag{3.2.11}$$

where $\mathbf{E}_h \in \mathbb{R}^{N_x(k+1)+1}$ and $\mathbf{\Phi}_h \in \mathbb{R}^{N_x(k+1)}$. We use an exact solver to solver (3.2.11) in the code since it is a 1D problem.

3.2.3 DG-FEM for Poisson equation

The DG method for the Poisson equation is very similar to the one introduced in the previous section but here we choose same space for E_h and Φ_h. More precisely we choose

$$\mathcal{Z}_h := \left\{ w | w \in L^2(\Omega_x),\, w|_{I_i} \in \mathcal{P}^{k+1}(I_i) \right\} \tag{3.2.12}$$

where k is the order of polynomial used for Vlasov equation. We take test functions $\tau, \nu \in \mathcal{Z}_h$, the weak formulation to the system of equations in an element, I_i, reads: find $(E_h(t,x), \Phi_h) \in (\mathcal{Z}_h \times \mathcal{Z}_h)$ that satisfy

$$\begin{aligned} \int_{I_i} E_h\,\tau &= -\int_{I_i} \Phi_h\,\tau_x + \widehat{\Phi}\,\tau\Big|_{\partial I_i} \qquad \forall \tau \in \mathcal{Z}_h && (3.2.13) \\ \int_{I_i} E_h\,\nu_x &= \int_{I_i} g\,\nu + \widehat{E}\,\nu\Big|_{\partial I_i} \qquad \forall \nu \in \mathcal{Z}_h && (3.2.14) \end{aligned}$$

method	c_{11}	c_{12}	c_{22}
LDG2	$(k+1)^2/h_x$	$sign(v)/2$	0
LDG3	$(k+1)^2/h_x$	$1/2$	0

Table 3.1: coefficients for LDG2 and LDG3 method.

where the numerical fluxes $\hat{E}$ and $\hat{\Phi}$ are defined as

$$\widehat{E}_{i+1/2} := \{\{E_h\}\} + c_{12}[\![E_h]\!] + c_{11}[\![\Phi_h]\!] \tag{3.2.15}$$

$$\widehat{\Phi}_{i+1/2} := \{\{\Phi_h\}\} - c_{12}[\![E_h]\!] + c_{22}[\![E_h]\!] \tag{3.2.16}$$

where following [3], we choose $c_{22} \equiv 0$ and $c_{11} \geq \frac{(k+1)^2}{h_x}$. Then we show two methods where c_{12} are chosen differently

1. *LDG2*

   In this case we choose the flux as

$$\begin{aligned} \widehat{E}_{i+1/2} &:= \{\{E_h\}\} + \frac{1}{2}[\![E_h]\!] + \frac{(k+1)^2}{h_x}[\![\Phi_h]\!] \\ \widehat{\Phi}_{i+1/2} &:= \{\{\Phi_h\}\} - \frac{1}{2}[\![E_h]\!] \end{aligned}$$

2. *LDG3; energy conservative method*

   It is proved in [3], choosing Poisson solver as above with $c_{12} = sign(v)/2$, coupled with the DG approximation to Vlasov equation discussed before will yield a numerical method that preserves energy

$$\begin{aligned} \widehat{E}_{i+1/2} &:= \{\{E_h\}\} + \frac{sign(v)}{2}[\![E_h]\!] + \frac{(k+1)^2}{h_x}[\![\Phi_h]\!] \\ \widehat{\Phi}_{i+1/2} &:= \{\{\Phi_h\}\} - \frac{sign(v)}{2}[\![E_h]\!] \end{aligned}$$

To simplify, we present the coefficients of LDG2/3 in Table 3.1. For computation, we express the unknown functions in terms of basis functions of $\mathcal{Z}_h$ (i.e. using Lagrange polynomials $\{l_m(x)\}$) and choosing test functions to be one of those basis functions, $l_n(x)$, we obtain

$$\begin{aligned} \sum_{m=1}^{k+1+1} E_h^m \underbrace{\int_{I_i} l_n(x) l_m(x)}_{M_{nm}} - \Phi_h^m \underbrace{\int_{I_i} l_m(x) \frac{d}{dx} l_n(x)}_{(S^T)_{nm}} - \hat{\Phi}\, l_n(x)\Big|_{\partial I_i} &= 0 \\ \sum_{m=1}^{k+1+1} E_h^m \underbrace{\int_{I_n} l_m(x) \frac{d}{dx} l_n(x)}_{(S^T)_{nm}} - \hat{E}\, l_n(x)\Big|_{\partial I_i} &= \int_{I_i} g\, l_n(x) \quad \forall n = 1, ..., (k+1+1) \end{aligned}$$

Note that a method to calculate matrices **M**, **S** will be introduced in chapter 4 for an approximation of order k. The system described above will finally look like

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{E}_h \\ \mathbf{\Phi}_h \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} \tag{3.2.17}$$

the size of left hand-side matrix in 1D will be $\{2N_x(k+1+1)\}^2$. In the actual code we are using an exact solver for this system

$$\begin{bmatrix} \mathbf{E}_h \\ \mathbf{\Phi}_h \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{C} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} \tag{3.2.18}$$

since the size of matrix is not very big. But generally, it is not a good way to solve the above system as when we extend the problem to 4D (2D space + 2D velocity) then the size of the system grows very much and inverting is very expensive. One may note that mixed-FEM has an advantage compare to the DG approach to Poisson problem since it has $2N_x$ less degree of freedoms and hence is faster to solve.

Chapter 4

Implementation

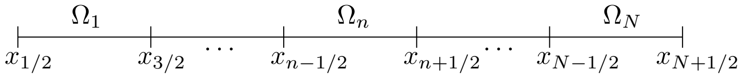
The goal of this section is to describe the actual implementation of DG scheme for the Vlasov-Poisson system. To ease the presentation, we begin with a linear transport equation and derive all local operators (here matrices) that will be used in the code. Due to the structure of Vlasov-Poisson system, we restrict our attention to a Cartesian grid.

4.1 Linear transport equation in 1D

Consider the following system:

$$\begin{cases} u_t + a\, u_x = \psi(t,x) & x \in \Omega := [0, 2\pi], a > 0, t > 0 \\ u(t, x=0) = u(t, x=2\pi) & \forall t \\ u(t=0, x) = u_0(x) & \forall x \in \Omega \end{cases} \tag{4.1.1}$$

We first partition the domain, Ω, to N non-overlapping elements, Ω_n, defined as



$$\Omega_n := [x_{n-1/2}, x_{n+1/2}], \text{ where } x_{1/2} = 0, x_{N+1/2} = 2\pi$$

and $\{x_{1/2}, ..., x_{N+1/2}\}$ is the set of physical coordinates. Now, we introduce the finite element space

$$V_h := \{v | v \in L^2(\Omega), v|_{\Omega_i} \in \mathcal{P}^k(\Omega_n), \forall n \in 1..N\}$$

where $\mathcal{P}^k(\Omega_n)$ is the set of polynomials of degree $\leq k$. Our approximate solution to (4.1.1), is a polynomial of degree k within each element Ω_n but is not necessarily continuous across each element. Inside each element, we define some basis functions for the space $\mathcal{P}^k(\Omega_n)$. We can consider Lagrange (nodal representation) or Legendre (modal representation) basis functions. We illustrate the Lagrange and Legendre polynomial in the reference element $I := [-1, 1]$ by

$$\text{Lagrange Polynomial} \qquad \hat{l}_i(r) := \prod_{1 \leq j \leq k+1, j \neq i} \frac{r - r_j}{r_i - r_j} r \in I \quad \forall i = 1, ..., k+1 \tag{4.1.2}$$

$$\text{Normalized Legendre Polynomial} \qquad \tilde{P}_i(r) \text{ s.t. } \int_I \tilde{P}_i(r) \tilde{P}_j(r) dr = \delta_{ij} \quad \forall i, j = 0, ..., k \tag{4.1.3}$$

$$\tilde{P}_i(r) := \sqrt{\frac{2i+1}{2}} P_i(r)$$

where δ_{ij} is the Kronecker delta, $\{r_1, ..., r_{k+1}\}$ is the set of distinct nodal coordinates in the reference element and $P_i(r)$ is the i-th Legendre polynomial. They can be derived from the recursion

$$\begin{aligned}(i+1)P_{i+1}(r) = (2i+1)rP_i(r) - iP_{i-1}(r) \qquad & \forall r \in I, i \geq 1 \\ & P_0(r) = 1, P_1(r) = r\end{aligned} \tag{4.1.4}$$

Note that Lagrange polynomials have the property:

$$\hat{l}_i(r_j) = \delta_{ij} \quad \forall i, j = 1..k+1$$

where the above basis functions are defined for the reference element, I. We can define the basis functions for an arbitrary element Ω_n using the affine map

$$\begin{aligned} x(r) = F(r) &:= \frac{h_n}{2} r + c_n \\ c_n &:= \frac{x_{n-1/2} + x_{n+1/2}}{2} \\ h_n &:= x_{n+1/2} - x_{n-1/2} = |\Omega_n| \end{aligned}$$

where $x(r) \in \Omega_n, r \in I$, and the basis functions in element Ω_n will look like

$$l_i^n(x) := \begin{cases} \hat{l}_i(r) & r \in I \\ 0 & r \notin I \end{cases} \tag{4.1.5}$$

$$\tilde{P}_i^n(x) := \begin{cases} \tilde{P}_i(r) & r \in I \\ 0 & r \notin I \end{cases} \tag{4.1.6}$$

Here $l_i^n(x)$ is the i-th Lagrange basis function in element Ω_n and $\tilde{P}_i^n(x)$ the corresponding normalized i-th Legendre basis function in Ω_n. The approximate solution to (4.1.1) belongs to V_h so it can be expressed in terms of the basis functions

$$\begin{aligned} \text{Legendre (modal) } u_h^n(t,x) &= \sum_{i=1}^{k+1} \tilde{u}_i^n(t) \tilde{P}_{i-1}^n(x) \quad \forall x \in \Omega_n \\ \text{Lagrange (nodal) } u_h^n(t,x) &= \sum_{j=1}^{k+1} u_h^n(t, r_j^n) l_j^n(x) \quad \forall x \in \Omega_n \end{aligned}$$

with $\{r_1^n, ..., r_{k+1}^n\}$ is the set of nodal coordinates in element n, where $r_1^n = x_{n-1/2}$ and $r_{k+1}^n = x_{n+1/2}$. In order to simplify the notation, we denote the solution in an element n evaluated at r_i^n using

$$u_i^n := u_h^n(t, r_i^n)$$

Remark 1. *in order to see why the coefficient of the nodal form is chosen as above one may set $x = r_i^n$ where r_i^n is one of the nodal coordinates within the Ω_n and get*

$$\begin{aligned} u_i^n(t) &= \sum_{j=1}^{k+1} u_h^n(t, r_j^n) l_j^n(r_i^n) \\ &= \sum_{j=1}^{k+1} u_h^n(t, r_j^n) \delta_{ij} \\ &= u_h^n(t, r_i^n) \end{aligned}$$

Hence the coefficients of Lagrange basis functions are the values of the solution evaluated at the nodes within each element. Finally the global solution of this approximation will be the sum of the local solutions

$$u_h(t,x) = \sum_{n=1}^{N} u_h^n(t,x)$$

The weak form of the (4.1.1), in each element is: find $u_h \in V_h$ s.t.

$$\int_{\Omega_n} \partial_t u_h \varphi - \int_{\Omega_n} a u_h \varphi_x + \int_{\partial\Omega_n} \widehat{au_h}\varphi = \int_{\Omega_n} f(t,x)\varphi \quad \forall \varphi \in V_h, \forall \Omega_n \tag{4.1.7}$$

Expressing the u_h in terms of basis functions of V_h (i.e. Lagrange polynomials) and choosing the test function as one of those, we will get in n-th element

$$\sum_{j=1}^{k+1} \partial_t u_j^n \int_{\Omega_n} l_j^n(x) l_i^n(x) dx - \int_{\Omega_n} u_j^n l_j^n(x) \frac{d}{dx} l_i^n(x) dx + [\widehat{au_h} l_i^n(x)]_{x_{n-1/2}}^{x_{n+1/2}} = \int_{\Omega_n} f(t,x) l_i^n(x); \forall i, n \tag{4.1.8}$$

Defining

$$\begin{aligned}
M_{ij}^n &= \textstyle\int_{\Omega_n} l_j^n(x) l_i^n(x) dx && \text{mass matrix} \\
S_{ij}^n &= \textstyle\int_{\Omega_n} l_i^n(x) \frac{d}{dx} l_j^n(x) dx && \text{gradient matrix} \\
\mathbf{u}_n &= [u_1^n(t), u_2^n(t), \ldots, u_{k+1}^n(t)] && \text{vector of solution in element } n \\
\mathbf{flux}^n &= [-u_h(x_{n-1/2}^-), 0, ..., 0, u_h(x_{n+1/2}^-)] && \text{flux vector of size } k+1 \text{ designed for DG}
\end{aligned} \tag{4.1.9}$$

then *lhs* of (4.1.8) will look like

$$\begin{aligned}
\textstyle\sum_{j=1}^{k+1} \partial_t u_j^n \int_{\Omega_n} l_j^n(x) l_i^n(x) dx &\equiv \mathbf{M}^n . \partial_t \mathbf{u}_n \\
u_j^n \int_{\Omega_n} l_j^n(x) \frac{d}{dx} l_i^n(x) dx &\equiv (\mathbf{S}^n)^T . \mathbf{u}_n \\
[\widehat{au_h} l_i^n(x)]_{x_{n-1/2}}^{x_{n+1/2}} &\equiv u_h(x_{n+1/2}^-)\delta_{i,k+1} - u_h(x_{n-1/2}^-)\delta_{i,1}
\end{aligned} \tag{4.1.10}$$

The third term in the *lhs* of (4.1.8) contains $\widehat{au_h}$, which is called numerical flux for DG. It can be defined as

$$\begin{aligned}
\widehat{au_h}(x_{i+1/2}) &= \{\{u_h\}\} - \mathtt{sgn}[a][\![u_h]\!]/2 = \{\{u_h\}\} - [\![u_h]\!]/2 \\
&= u_h(x_{i+1/2}^-)
\end{aligned}$$

and

$$\widehat{au_h}(x_{i-1/2}) = u_h(x_{i+1/2}^-)$$

where $\{\{.\}\}$ is the average function across the element and $[\![.]\!]$ is the jump function. Hence

$$[\widehat{au_h} l_i^n(x)]_{x_{n-1/2}}^{x_{n+1/2}} = \begin{bmatrix} -u_h(x_{n-1/2}^-) \\ 0 \\ \vdots \\ 0 \\ u_h(x_{n+1/2}^-) \end{bmatrix} = \mathbf{flux}^n$$

In other word, the flux vector of element n, in its first element has the value of the solution from the previous element, $n-1$, using $-u_{k+1}^{n-1}$ and its last element by u_{k+1}^n (the so-called upwind flux).

4.1.1 Local Operators

Mass matrix

We are interested in computing the elements of the mass matrix in the reference element, I, for an approximation of degree k

$$M^I_{ij} = \int_I \hat{l}_j(x)\hat{l}_i(x)dx \quad \forall i, j \in 1..(k+1)$$

Note that the mass matrix is symmetric and its relation with the mass matrix in element n is

$$\mathbf{M}^n = \frac{|\Omega_n|}{2}\mathbf{M}^I$$

Now in order to find a procedure to compute this matrix for any order we may use the advantage of orthogonal polynomials (modal form) instead of Lagrange basis functions (nodal representation). Recall that the approximate solution in these two forms will look like

$$u_h(x,t) = \sum_{j=1}^{k+1} \tilde{u}_j(t)\tilde{P}_{j-1}(x) = \sum_{m=1}^{k+1} u_h(r_m,t)\hat{l}_m(x), \forall x \in I$$

and if we set x to be r_i, we will get

$$u_h(r_i,t) = \sum_{j=1}^{k+1} \tilde{u}_j(t)\tilde{P}_{j-1}(r_i)$$

denoting the $V_{ij} := \tilde{P}_{j-1}(r_i)$, the so-called Vandermonde matrix, we will get the relations

$$\mathbf{u} = \mathbf{V}.\tilde{\mathbf{u}} \tag{4.1.11}$$

where $\tilde{\mathbf{u}}$ is the coefficient of Legendre basis functions and $\mathbf{u}$ is the vector of nodal values of Lagrange polynomials. Moreover if we change basis using

$$\tilde{P}_{i-1}(r) = \sum_{j=1}^{k+1} \tilde{P}_{i-1}(r_j)\hat{l}_j(r)$$

we will obtain the following matrix form

$$\begin{aligned} \mathbf{V}^T.\mathbf{l} &= \tilde{\mathbf{P}} \qquad (4.1.12) \\ \mathbf{l} &:= \begin{bmatrix} \hat{l}_1(r) \\ \hat{l}_2(r) \\ \vdots \\ \hat{l}_{k+1}(r) \end{bmatrix} \\ \tilde{\mathbf{P}} &:= \begin{bmatrix} \tilde{P}_0(r) \\ \tilde{P}_1(r) \\ \vdots \\ \tilde{P}_k(r) \end{bmatrix} \end{aligned}$$

then the mass matrix will be

$$\begin{aligned}
M^I_{ij} &= \int_I \sum_{m=1}^{k+1} (V^T)^{-1}_{im} \tilde{P}_{m-1} \sum_{n=1}^{k+1} (V^T)^{-1}_{jn} \tilde{P}_{n-1} \\
&= \sum_{m=1}^{k+1} \sum_{n=1}^{k+1} (V^T)^{-1}_{im} (V^T)^{-1}_{jn} \int_I \tilde{P}_{m-1} \tilde{P}_{n-1} \\
&= \sum_{m=1}^{k+1} \sum_{n=1}^{k+1} (V^T)^{-1}_{im} (V^T)^{-1}_{jn} \delta_{m,n} \\
&= \sum_{n=1}^{k+1} (V^T)^{-1}_{in} (V^T)^{-1}_{jn} \\
\mathbf{M}^I &= (\mathbf{V}\mathbf{V}^T)^{-1}
\end{aligned}$$

and

$$\mathbf{M}^n = \frac{|\Omega_n|}{2} (\mathbf{V}\mathbf{V}^T)^{-1}$$

Hence by constructing the Vandermonde matrix and using a procedure to iteratively construct the Legendre polynomials we can easily have the mass matrix of any order. Moreover the inversion of the $\mathbf{V}\mathbf{V}^T$ is not costly since it is a matrix of size $(k+1)^2$, where k is degree of polynomials and usually is very small (i.e. 1,2,3) and it is done once and will be used during computation.

Gradient matrix

In the reference element, we have

$$S^n_{ij} = \int_{\Omega_n} l^n_i \partial_x l^n_j dx = \int_I \hat{l}_i \partial_r \hat{l}_j dr = S^I_{ij}$$

The procedure to find high order gradient matrix will be the following. First we introduce a differentiation matrix $\mathbf{D}_r$ such that

$$D_{r,ij} = \left. \frac{d\hat{l}_j}{dx} \right|_{r_i}, \forall r_i \in I$$

Now in order to construct the gradient matrix we express the derivative in S_{ij} in terms of the Lagrange basis functions introduced for our space of functions

$$\begin{aligned}
S^I_{ij} &= \int_I \hat{l}_i \partial_x \hat{l}_j dx = \int_I \hat{l}_i \sum_{m=1}^{k+1} \hat{l}_m \left. \partial_x \hat{l}_j \right|_{r_m} dx \\
&= \sum_{m=1}^{k+1} \left(\int_I \hat{l}_i \hat{l}_m dx \right) \left. \partial_x \hat{l}_j \right|_{r_m} \\
&= \sum_{m=1}^{k+1} M_{im} D_{r,mj} \\
\mathbf{S}^I &= \mathbf{M}\mathbf{D}_r
\end{aligned}$$

Now the problem changed to find the $\mathbf{D}_r$ in a way for high order approximation. We start from the identity between the nodal and modal representation, $\mathbf{V}^T \mathbf{l} = \tilde{\mathbf{P}}$, which is derived in (4.1.12):

$$\mathbf{V}^T \frac{d}{dx} \hat{\mathbf{l}} = \frac{d}{dx} \tilde{\mathbf{P}}$$

evaluating the derivatives at r_i gives

$$\mathbf{V}^T \mathbf{D}^T_r = (\mathbf{V}_r)^T$$

where

$$V^T_{r,ij} := \frac{d}{dx}\tilde{P}_{j-1}\bigg|_{r_i}$$

and by using the Legendre's polynomial identities we compute $\mathbf{V}_r$ by

$$\frac{d}{dx}\tilde{P}_j = \sqrt{j(j+1)}\tilde{P}^{(1,1)}_{j-1}(x)$$

where $\tilde{P}^{(1,1)}_{j-1}$ is the Jacobi polynomial [**?**].

4.1.2 Mesh Generation

Here we introduce the objects of a simple mesh, generated for our DG method in 1D and later for 2D case. We start constructing the objects by numbering the physical coordinates:

$$\begin{array}{rcl} x_{1/2} & \leftrightarrow & \eta_1 \\ x_{3/2} & \leftrightarrow & \eta_2 \\ & \vdots & \\ x_{i-1/2} & \leftrightarrow & \eta_i \\ & \vdots & \\ x_{N+1/2} & \leftrightarrow & \eta_{N+1} \end{array}$$

as we are in 1D it is better to choose the set of vertices, $\{\eta_i\}$, as

$$Vertices := \{\eta_1 = 1, \eta_2 = 2, ..., \eta_{N+1} = N+1\}$$

Moreover we would like to label the set of elements $\{\Omega_n\}$ by the same rule as

$$Elements := \{\Omega_1 = 1, \Omega_2 = 2, ..., \Omega_{N+1} = N+1\}$$

after defining those sets, we would like to know the relation between them using a matrix that we call `E2V`:

$$\mathtt{E2V} := \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ \vdots & \vdots \\ N & N+1 \end{bmatrix} \in \mathcal{M}_{N\times 2}$$

where $\mathtt{E2V}_{i,j}$ means that the element Ω_i contains vertices $\{\mathtt{E2V}_{i,1}, \mathtt{E2V}_{i,2}\}$.

Furthermore, as we introduced before, we have some nodal coordinates defined in element n as $\{r^n_1, ..., r^n_{k+1}\}$. Remember that in this notation $r^n_1 = x_{n-1/2}$ and $r^n_{k+1} = x_{n+1/2}$. As we labeled the physical coordinates by the set of vertices, we would like also to label each nodal coordinates by a set which we call *faces*, $F := \{\xi_i\}$. The relation between each nodal coordinates in an element i and the corresponding face will be

$$r^i_j \leftrightarrow \xi_{(i-1)(k+1)+j}; \forall i = 1, ..., N, \quad j = 1, ..., k+1 \tag{4.1.13}$$

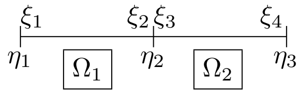
For instance, consider the case where $k = 1$ (first order) and $N = 2$ (number of elements), then the set of faces is

$$F = \{\underbrace{\xi_1, \xi_2}_{\in\Omega_1}, \underbrace{\xi_3, \xi_4}_{\in\Omega_2}\}$$

moreover as $N = 2$, there are $N + 1$ physical coordinates and hence the set of physical vertices is

$$Vertices = \{\eta_1, \eta_2, \eta_3\}$$

Note that the coordinates of the ξ_2 and ξ_3 are same. We call these faces adjacent to each other. The geometry of these objects is shown below



Observe that using the above definition for labeling the nodal coordinates in (4.1.13) we can easily extract the information regarding the relation of elements and faces. We denote by $\mathtt{E2F} \in \mathcal{M}_{N\times(k+1)}$ the matrix

$$\mathtt{E2F} := \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_{k+1} \\ \xi_{k+2} & \xi_{k+3} & \cdots & \xi_{2(k+1)} \\ & & \vdots & \\ & & \xi_{(i-1)(k+1)+j} & \\ & & \vdots & \\ & & \cdots & \xi_{N(k+1)} \end{bmatrix}$$

Then knowing E2F and E2V, we can construct the relation between faces and vertices denoted by $\mathtt{F2V} \in \mathcal{M}_{N(k+1)\times 2}$ and given by

$$\mathtt{F2V} := \begin{bmatrix} \xi_1 & \eta_1 \\ \xi_2 & \eta_2 \\ \xi_3 & \eta_2 \\ \xi_4 & \eta_3 \\ \vdots & \\ \xi_{N(k+1)} & \eta_{N+1} \end{bmatrix}$$

Now all objects that we need for implementation are constructed. Finally we express the vector of nodal values, $\mathbf{u}_h$, using the following vector $\mathtt{u}_h \in \mathbb{R}^{N(k+1)}$:

$$\mathtt{u}_h[\xi_{(i-1)(k+1)+j}] \equiv \mathbf{u}_h(r^i_j) \quad \forall i \in 1..N, j \in 1..k+1$$

4.1.3 Assembling

The discretized equation (4.1.8) in an element n reads in terms of our “local operators”

$$\mathbf{M}^n \frac{d\mathbf{u}^n_h}{dt} \;=\; (\mathbf{S}^n)^T\mathbf{u}^n_h - \mathbf{flux}^n + \Psi^n \tag{4.1.14}$$

where

$$\begin{array}{lcll} \mathbf{u}^n_h & = & [u^n_1(t), u^n_2(t), \ldots, u^n_{k+1}(t)] & \text{vector of solution in element } n \\ \mathbf{flux}^n & = & [-u_h(x^-_{n-1/2}), 0, ..., 0, u_h(x^-_{n+1/2})] & \text{flux vector of size } k+1 \text{ designed for DG} \\ \Psi^n & \equiv & \int_{\Omega_n} \psi(t,x)\mathbf{l}^n(x)dx & \text{right hand side; evaluated by Gauss quadrature} \end{array}$$

Moreover, one may note that in (4.1.14), we need an integrator in time to obtain the evolution of the solution. For this advection problem using a first order explicit integrator with carefully chosen dt may suffice.

$$\mathbf{u}^n_h(t+dt) \;=\; \mathbf{u}^n_h(t) + dt\cdot(\mathbf{M}^n)^{-1}\left((\mathbf{S}^n)^T\mathbf{u}^n_h - \mathbf{flux}^n + \Psi^n\right) \quad \forall n \tag{4.1.15}$$

Finally by having all recipes for this 1D case and using a proper data structure for retrieving information from the mesh and elements, we can update the values of the solution in each element (element-wise). Recall that inversion of $(\mathbf{M}^n)^{-1}$ is low cost and is done once for all time.

4.1.4 Simulation in 1D

We use the method described before for approximating the solution of the following advection problem

$$\begin{cases} u_t + a\, u_x = -2\pi \sin(x + 2\pi t) & x \in \Omega := [0, 2\pi], a = 2\pi, t > 0 \\ u(t, 0) = u(t, 2\pi) & \text{boundary condition} \\ u(0, x) = \cos(x) & \forall x \in \Omega, \text{initial data} \end{cases} \tag{4.1.16}$$

The analytical solution of the (4.1.16) is

$$u(t, x) = \cos(2\pi t) \cos(x)$$

The DG approximation solution and the exact one is shown in Fig. 4.1(left). Note that in Fig. 4.1(right) we show the order of convergence with respect to the L^2-norm versus refinement of the mesh using polynomials of degree 1 ($N = 1$) which yields an order of accuracy of magnitude 2.

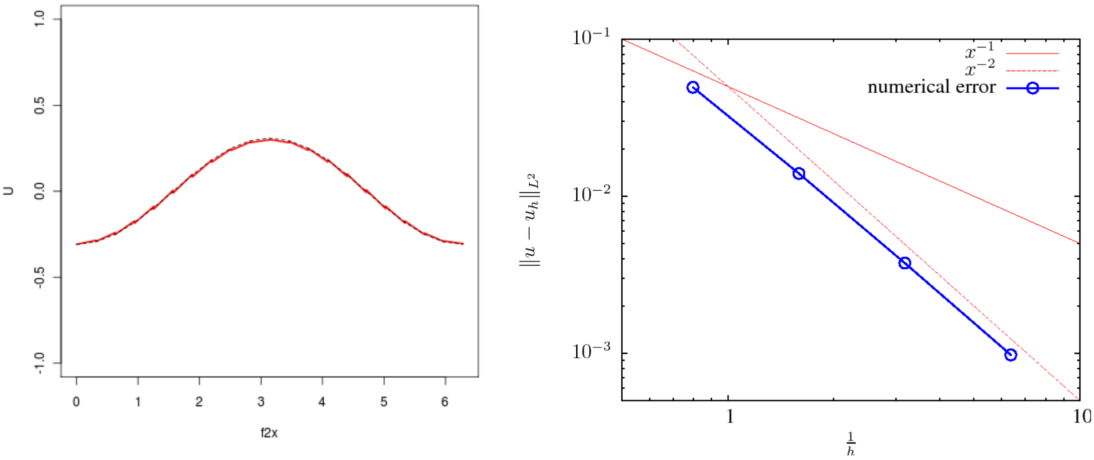


Figure 4.1: the approximate solution of (4.1.16) (red) at $t = 0.7$ (left). log-log plot of h^{-2} (dashed), L^2 error of approximate solution (solid) using linear polynomials (right).

4.2 Vlasov equation

We now use the the ingredients introduced in the previous section and considering extension of 1D advection problem to solve the Vlasov equation:

$$\begin{cases} f_t + v f_x - E(t, x) f_v = 0 & x \in \Omega_x := [x_l, x_r], v \in \Omega_v := [v_d, v_u] \\ f(t, 0, v) = f(t, 1, v) & \text{periodic boundary condition in } x, v \in \Omega_v \\ f(0, x, v) = f_0(x, v) & \text{initial data}, supp(f_0(., v)) \in \Omega_v \end{cases} \tag{4.2.1}$$

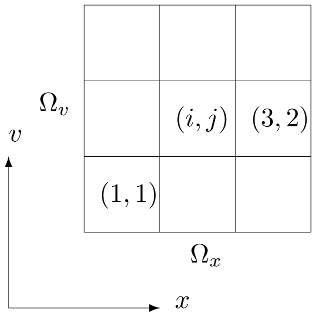
where $E(x)$ is a given electric field. Before introducing the space of functions, its basis and weak form of (4.2.1), we introduce basic notation that will be needed.

4.2.1 Notation and parameters

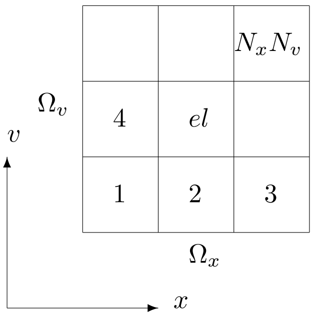
First of all, the DG weak form of the (4.2.1) with the defined numerical flux in Proposition 1 suggests to employ a Cartesian mesh. Hence we choose a Cartesian grid and will use same method for generating the objects of the mesh in 2D (as we did for the 1D case). We begin by defining some global parameters that will be used later very often. Regarding the mesh in 2D, we define:

1. N_x: number of elements in the x-coordinate.
2. N_v: number of elements in the v-coordinate.
3. $N_x + 1$: number of physical points in the x-coordinate.
4. $N_v + 1$: number of physical points in the v-coordinate.
5. Ω_x: computation domain in x-coordinate.
6. Ω_v: computation domain in v-coordinate.
7. I_i: computation domain in x-coordinate in element (i, j).
8. J_j: computation domain in y-coordinate in element (i, j).
9. $h_x(i) = x_{i+1/2} - x_{i-1/2}$: is the length of I_i.
10. $h_v(j) = v_{j+1/2} - v_{j-1/2}$: is the length of J_j.
11. $c_x(i) = \frac{x_{i+1/2}+x_{i-1/2}}{2}$: is the center of I_i.
12. $c_v(j) = \frac{v_{j+1/2}+v_{j-1/2}}{2}$: is the center of J_j.
13. k: is the degree of polynomial in an element.
14. $N_p^{2D} := N_p^2 = (k + 1)^2$: is the number of faces in an element in 2D case.

For a given computational domain $\Omega_x \times \Omega_v$, we partition the domain to $N_x \times N_v$ elements where the total number of elements is $N_x N_v$. Then we introduce two numbering procedure to label the elements produced by the partitioning. First, we introduce a simple Cartesian one, which is a 2D labeling to each element, for instance consider the following mesh with a 3 × 3 partition:



We assign an array like (i, j) to each element where $i = 1, ..., N_x$ represents the column of the element and $j \in 1..N_v$ is the row of an element. Another procedure consists in assigning an integer label (instead of (i, j)) to each element starting from most left-down element and counting toward right elements. When the most right element in a row is read, we go to the upper row and begin from the most left one to count:

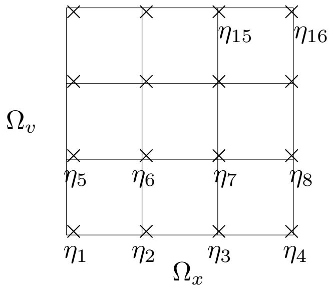


We define $el \in 1, ..., N_xN_v$ as the index of an element in a N_xN_v partition of a $\Omega_x \times \Omega_v$ domain in the integer labeling. The translation between these two approaches is

$$\begin{array}{rcl} (i,j) & \rightarrow & el := (j-1)N_x + i \\ el & \rightarrow & (i,j) := (\texttt{FLOOR}(el/N_x) + 1, el - \texttt{FLOOR}(el/N_x)) \end{array}$$

4.2.2 Mesh objects in 2D

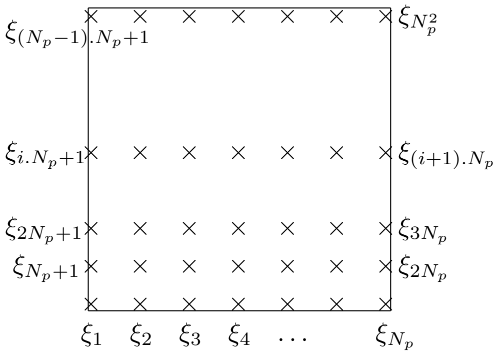
Here, we will introduce the objects of the 2D mesh similar to what we introduced in 1D case. We begin with the set of vertices in the computational domain by introducing $V := \{\eta_1, \eta_2, \ldots, \eta_{(N_x+1)(N_v+1)}\}$. We label mesh nodes by $\{\eta_i\}$ in the following way:



and with same procedure we can construct the matrix $\texttt{E2V} \in \mathcal{M}_{(N_xN_v),4}$

$$\texttt{E2V} := \begin{bmatrix} 1 & 2 & 5 & 6 \\ 2 & 3 & 6 & 7 \\ & & \vdots & \\ & \ldots & (N_x+1)(N_v+1)-1 & (N_x+1)(N_v+1) \end{bmatrix}$$

The set of faces is constructed using $F := \{\xi_1, \xi_2, \ldots, \xi_{N_p^2 N_x N_v}\}$. The structure of the faces in the first element, $el = 1$, can be represented in the following way



and the matrix $\texttt{E2F} \in \mathcal{M}_{(N_xN_v),N_p^2}$ is

$$\texttt{E2F} := \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_{N_p^2} \\ \xi_{N_p^2+1} & \xi_{N_p^2+2} & \cdots & \xi_{2N_p^2} \\ & & \vdots & \\ \xi_{(N_xN_v-1)N_p^2+1} & \cdots & & \xi_{N_xN_vN_p^2} \end{bmatrix}$$

We may also define four class of faces that represent the adjacent faces to the edges of an element. They will be very useful in calculation of local operators: $\mathbf{faces_{down}}$, $\mathbf{faces_{left}}$, $\mathbf{faces_{right}}$ and $\mathbf{faces_{up}}$. In the first element, $el = 1$, they are defined as:

$$\begin{aligned} faces_{down}^n &:= \xi_n \quad \forall n \in 1..N_p \\ faces_{left}^n &:= \xi_{1+N_p(n-1)} \\ faces_{right}^n &:= \xi_{N_p \cdot n} \\ faces_{up}^n &:= \xi_{N_p(N_p-1)+n} \end{aligned} \tag{4.2.2}$$

for instance in the case $k = 2$, we will have:

$$\begin{aligned} \mathbf{faces_{down}} &:= \{\xi_1, \xi_2, \xi_3\} \\ \mathbf{faces_{left}} &:= \{\xi_1, \xi_4, \xi_7\} \\ \mathbf{faces_{right}} &:= \{\xi_3, \xi_6, \xi_9\} \\ \mathbf{faces_{up}} &:= \{\xi_7, \xi_8, \xi_9\} \end{aligned}$$

4.2.3 Basis function in 2D

We define the DG finite element space:

$$V_h := \{w | w \in L^2(\Omega_x \times \Omega_v),\, w|_{K_{ij}} \in \mathcal{Q}^k(K_{ij})\}$$

where $\mathcal{Q}^k$ is the set of polynomials of degree at most k in each variables:

$$\mathcal{Q}^k := \texttt{span}\{w | w \in x^i y^j, 0 \le i, j \le k\}$$

where the dimension of $\mathcal{Q}^k$ is $(\dim\mathcal{P}^k)^2$ and $\dim\mathcal{P}^k = (k+1)^2 = N_p^2$.

Now consider we are working in the reference element, I^2, we can define a set of basis functions for $\mathcal{Q}^k$ such that basis functions are made by a tensor product of the bases relative to $\mathcal{P}^k$. For instance, consider $k = 1, (N_p = 2)$ then:

$$\begin{aligned}
\hat{\lambda}_1(x,y) &:= \hat{l}_1(x)\hat{l}_1(y) \\
\hat{\lambda}_2(x,y) &:= \hat{l}_2(x)\hat{l}_1(y) \\
\hat{\lambda}_3(x,y) &:= \hat{l}_1(x)\hat{l}_2(y) \\
\hat{\lambda}_4(x,y) &:= \hat{l}_2(x)\hat{l}_2(y)
\end{aligned}$$

and for general case (arbitrary k), we will get

$$\hat{\lambda}_n(x,y) := \hat{l}_r(x)\hat{l}_s(y) \qquad \begin{aligned} &r, s = 1, ..., N_p \\ &n = r + (s-1)N_p \end{aligned} \tag{4.2.3}$$

note that r, s indices runs from 1 to N_p, while the n-index of the basis function in 2 dimension runs from 1 to $N_p^{2D} = (N_p + 1)^2$. To clarify the index notation, we mention that here i, j are indices that will be used for elements in the computational domain and n, m to identify the basis function inside an element. Moreover, as we are using Cartesian grid for our computations and tensor product basis functions, then it is possible to decompose a 2D-integral in an element (i, j) into two 1D-integrals

$$\int_{I_i}\int_{J_j} \lambda_m \lambda_n dx\, dv \quad = \quad \left(\int_{I_i} \lambda_m^X \lambda_n^X dx\right)\left(\int_{J_j} \lambda_m^V \lambda_n^V dv\right) \quad \forall n, m \in 1..N_p^{2D}$$

where λ_n^V, λ_n^X are the x and v decomposed parts of the $\lambda_n(x, v)$. For instance if $k = 1$, we will have:

$$\begin{array}{lcl|lcl}
\hat{\lambda}_1^V(v) & := & \hat{l}_1(v) & \hat{\lambda}_1^X(x) & := & \hat{l}_1(x) \\
\hat{\lambda}_2^V(v) & := & \hat{l}_1(v) & \hat{\lambda}_2^X(x) & := & \hat{l}_2(x) \\
\hat{\lambda}_3^V(v) & := & \hat{l}_2(v) & \hat{\lambda}_3^X(x) & := & \hat{l}_1(x) \\
\hat{\lambda}_4^V(v) & := & \hat{l}_2(v) & \hat{\lambda}_4^X(x) & := & \hat{l}_2(x)
\end{array} \tag{4.2.4}$$

We express our approximate solution in an element K_{ij} by

$$f^h(t,x,v) = \sum_{n=1}^{N_p^2} u_h^n(t)\lambda_n(x,v) \quad (t,x,v) \in \mathbb{R}^+ \times K_{ij} \tag{4.2.5}$$

4.2.4 DG-FEM for Vlasov equation

as we discussed, the DG formulation of (4.2.1) in an element K_{ij} will be:

$$\begin{aligned}
0 = \int_{K_{ij}} f_t \varphi dx\, dv \quad &+ \quad \left(\int_{J_j}\int_{I_i} E(x) f \varphi_v dx\, dv - \int_{J_j}\int_{I_i} v f \varphi_x dx\, dv\right) \\
&- \quad \left(\int_{I_i} \widehat{Ef}\varphi\Big|_{\partial J_j} dx - \int_{J_j} \widehat{vf}\varphi\Big|_{\partial I_i} dv\right) \\
\varphi, f \quad &\in \quad V_h := \{w | w \in L^2(\Omega_x \times \Omega_v), w|_{K_{ij}} \in \mathcal{Q}^k(K_{ij})\}
\end{aligned} \tag{4.2.6}$$

where the numerical fluxes are

$$\begin{aligned}
\widehat{vf}(x_{i+\frac{1}{2}}, v) \quad &:= \quad \begin{cases} vf(x^-_{i+\frac{1}{2}}, v) & v \geq 0 \\ vf(x^+_{i+\frac{1}{2}}, v) & v \leq 0 \end{cases} \\
\widehat{Ef}(x, v_{j+\frac{1}{2}}) \quad &:= \quad \begin{cases} E(x)f(x, v^+_{j+\frac{1}{2}}) & E(x) \geq 0 \\ E(x)f(x, v^-_{j+\frac{1}{2}}) & E(x) \leq 0 \end{cases} \\
\{\{f_{i+\frac{1}{2}}\}\} \quad &:= \quad \frac{f^+_{i+\frac{1}{2}} + f^-_{i+\frac{1}{2}}}{2} \quad \text{average operator} \\
[\![f_{i+\frac{1}{2}}]\!] \quad &:= \quad f^+_{i+\frac{1}{2}} - f^-_{i+\frac{1}{2}} \quad \text{jump operator}
\end{aligned}$$

Taking $\varphi = \lambda_n(x,y)$ we will get:

$$\begin{aligned} 0 = \int_{K_{ij}} f_t \lambda_n dx\,dv \quad &+ \quad \left(\int_{J_j}\int_{I_i} E(x) f \partial_v \lambda_n dx\,dv - \int_{J_j}\int_{I_i} v f \partial_x \lambda_n dx\,dv\right) \\ &- \quad \left(\int_{I_i} \widehat{Ef}\lambda_n\Big|_{\partial J_j} dx - \int_{J_j}\widehat{vf}\lambda_n\Big|_{\partial I_i} dv\right) \quad \forall n \in 1..N_p^2 \end{aligned} \tag{4.2.7}$$

Mass matrix

$$\begin{aligned} \int_{K_{ij}} f_t \lambda_n dx\,dv &= \sum_{m=1}^{N_p^2} \frac{du_h^m}{dt} \int_{K_{ij}} \lambda_m \lambda_n dx\,dv \\ &= \frac{|I_i|.|J_j|}{4} \sum_{m=1}^{N_p^2} \frac{du_h^m}{dt} \int_{I^2} \hat{\lambda}_m \hat{\lambda}_n dx\,dv \end{aligned}$$

now defining $M^{2D}_{n\,m} := \int_{I^2} \hat{\lambda}_m \hat{\lambda}_n dx\,dv$ as the mass matrix in 2D in the reference element, we are keen to compute it for an arbitrary k. For the beginning we choose $k = 1$ ($N_p^2 = 4$) and define

$$\hat{\lambda}_n := \hat{\lambda}_n^X(x)\hat{\lambda}_n^Y(y)$$

where $\hat{\lambda}_n^X$ is the part of $\hat{\lambda}_n$ that depends only on x. Hence we will have

$$\int_{I^2} \hat{\lambda}_m \hat{\lambda}_n dx\,dv = \left(\int_I \hat{\lambda}_m^X \hat{\lambda}_n^X dx\right)\left(\int_I \hat{\lambda}_m^V \hat{\lambda}_n^V dv\right)$$

then

$$\begin{array}{rcl|rcl} \hat{\lambda}_1^V(v) & := & \hat{l}_1(v) & \hat{\lambda}_1^X(x) & := & \hat{l}_1(x) \\ \hat{\lambda}_2^V(v) & := & \hat{l}_1(v) & \hat{\lambda}_2^X(x) & := & \hat{l}_2(x) \\ \hat{\lambda}_3^V(v) & := & \hat{l}_2(v) & \hat{\lambda}_3^X(x) & := & \hat{l}_1(x) \\ \hat{\lambda}_4^V(v) & := & \hat{l}_2(v) & \hat{\lambda}_4^X(x) & := & \hat{l}_2(x) \end{array} \tag{4.2.8}$$

therefore

$$\begin{aligned} \left(\int_I \hat{\lambda}_m^X \hat{\lambda}_n^X dx\right) &\equiv \left[\begin{array}{c|c} \mathbf{M}^{1D} & \mathbf{M}^{1D} \\ \hline \mathbf{M}^{1D} & \mathbf{M}^{1D} \end{array}\right]_{4\times 4} \\ \left(\int_I \hat{\lambda}_m^V \hat{\lambda}_n^V dv\right) &\equiv \left[\begin{array}{c|c} M_{11}\mathbf{F} & M_{12}\mathbf{F} \\ \hline M_{21}\mathbf{F} & M_{22}\mathbf{F} \end{array}\right]_{4\times 4} \\ \mathbf{F} &:= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

finally

$$M^{2D}_{nm} := \int_{I^2} \hat{\lambda}_n \hat{\lambda}_m dx\,dv \tag{4.2.9}$$

$$\mathbf{M}^{2D} = \left[\begin{array}{c|c} \mathbf{M}^{1D} & \mathbf{M}^{1D} \\ \hline \mathbf{M}^{1D} & \mathbf{M}^{1D} \end{array}\right] \otimes \left[\begin{array}{c|c} M_{11}\mathbf{F} & M_{12}\mathbf{F} \\ \hline M_{21}\mathbf{F} & M_{22}\mathbf{F} \end{array}\right] \tag{4.2.10}$$

and in general case (arbitrary k)

$$\begin{aligned} \mathbf{M}^{1D} &\in \mathcal{M}_{N_p \times N_p} \\ \mathbf{F} &\in \mathcal{M}_{N_p \times N_p} \\ \mathbf{M}^{2D} &:= \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{M}^{1D} & \mathbf{M}^{1D} & \dots & \mathbf{M}^{1D} \\ \hline & & \vdots & \\ \hline \mathbf{M}^{1D} & \mathbf{M}^{1D} & \dots & \mathbf{M}^{1D} \end{array}\right]}_{N_p^{2D}} \otimes \left[\begin{array}{c|c|c|c} M_{11}\mathbf{F} & M_{12}\mathbf{F} & \dots & M_{1,N_p}\mathbf{F} \\ \hline & & \vdots & \\ \hline M_{N_p,1}\mathbf{F} & M_{N_p,2}\mathbf{F} & \dots & M_{N_p,N_p}\mathbf{F} \end{array}\right] \end{aligned} \tag{4.2.11}$$

where $\mathbf{M}^{1D}$ is the mass matrix generated using polynomials of degree k ($N_p = k+1$, $N_p^{2D} = N_p^2$) and $\mathbf{F}$ is a full matrix with elements equal to 1. Finally the Mass matrix in an element $K_{ij} = I_i \times J_j$ will be

$$\mathbf{M}^{2D}_{(i,j)} = \frac{|I_i|.|J_j|}{4}\mathbf{M}^{2D}$$

V-gradient matrix

$$\begin{aligned}
\int_{J_j}\int_{I_i} E(x) f \partial_v \lambda_n dx\, dv &= \sum_{m=1}^{N_p^2}\left(\int_{J_j}\int_{I_i} E(x)\lambda_m \partial_v \lambda_n dx\, dv\right) u_h^m \\
&=: \sum_{m=1}^{N_p^2} VolE_{n,m}\, u_h^m \\
&= \sum_{m=1}^{N_p^2}\left(\int_{I^2} E\left(c_x(i) + \frac{h_x(i)}{2}x\right)\hat{\lambda}_m \partial_v \hat{\lambda}_n dx\, dv\right) u_h^m \\
&= \sum_{m=1}^{N_p^2}\left(\int_{I} E\left(c_x(i) + \frac{h_x(i)}{2}x\right)\hat{\lambda}^X_m \hat{\lambda}^X_n dx\right)\left(\int_I \hat{\lambda}^V_m \frac{d\hat{\lambda}^V_n}{dv} dv\right) u_h^m \\
&= \sum_{m=1}^{N_p^2} IntE^{(i)}_{nm}\, SV^T_{nm} u_h^m \quad \forall n \in 1..N_p^2
\end{aligned}$$

where

$$VolE_{n,m} := \left(\int_{J_j}\int_{I_i} E(x)\lambda_m \partial_v \lambda_n dx\, dv\right) \tag{4.2.12}$$

$$SV_{n,m} := \int_I \hat{\lambda}^V_n \frac{d\hat{\lambda}^V_m}{dv} dv \tag{4.2.13}$$

$$IntE^{(i)}_{n,m} := \int_I E\left(c_x(i) + \frac{h_x(i)}{2}x\right)\hat{\lambda}^X_m \hat{\lambda}^X_n dx \tag{4.2.14}$$

Starting with the simplest case, $k = 1$, and using the identities in (4.2.4), we obtain

$$\mathbf{SV}^{2D} := \left[\begin{array}{c|c} S^{1D}_{11}\mathbf{F} & S^{1D}_{12}\mathbf{F} \\ \hline S^{1D}_{21}\mathbf{F} & S^{1D}_{22}\mathbf{F} \end{array}\right] \tag{4.2.15}$$

where $\mathbf{S}^{1D}$ is the gradient matrix we derived in §4.1.1 for 1D case and in the general case (arbitrary k) reads

$$\begin{aligned}
\mathbf{S}^{1D} &\in \mathcal{M}_{N_p \times N_p} \\
\mathbf{F} &\in \mathcal{M}_{N_p \times N_p} \\
\mathbf{SV}^{2D} &:= \left[\begin{array}{c|c|c|c} S^{1D}_{11}\mathbf{F} & S^{1D}_{12}\mathbf{F} & \dots & S^{1D}_{1,N_p}\mathbf{F} \\ \hline & & \vdots & \\ \hline S^{1D}_{N_p,1}\mathbf{F} & S^{1D}_{N_p,2}\mathbf{F} & \dots & S^{1D}_{N_p,N_p}\mathbf{F} \end{array}\right]_{N_p^{2D}\times N_p^{2D}}
\end{aligned} \tag{4.2.16}$$

The other term, $IntE^{(i)}_{n,m}$, which is the electric field integral in an element (i,j), can be evaluated by Gauss quadrature. Note that the electric field function is a polynomial of degree $k+1$ where the degree of solution, f_h, is k, so the integrand is a polynomial of degree $3k+1$. For the Gauss quadrature we should specify the number of Gauss nodes, N_{Gauss} which determines the degree of precision $2N_{Gauss}-1$ (i.e. integrates exactly any polynomial of degree $\leq 2N_{Gauss}-1$). We choose

$$N_{Gauss} := \texttt{CEILING}[\frac{3k+2}{2}]$$

The V-gradient matrix in an element $el \equiv (i,j)$ is

$$\mathbf{VolE}^{(i,j)} := \mathbf{IntE}^{(i)} \otimes \mathbf{SV}^T \tag{4.2.17}$$

X-gradient matrix

$$\begin{aligned}
\int_{J_j}\int_{I_i} vf\partial_x\lambda_n dx\, dv &= \sum_{m=1}^{N_p^2}\left(\int_{J_j}\int_{I_i} v\lambda_m\partial_x\lambda_n dx\, dv\right)u_h^m \\
&=: \sum_{m=1}^{N_p^2} VolV_{nm}^{(i,j)}u_h^m \\
&= \sum_{m=1}^{N_p^2}\left(\frac{h_y(j)}{2}\int_{I^2}\left(c_y(j)+\frac{h_y(j)}{2}v\right)\hat{\lambda}_m\partial_x\hat{\lambda}_n dx\, dv\right)u_h^m \\
&= \frac{h_y(j)}{2}\sum_{m=1}^{N_p^2}\left(\int_I\left(c_y(j)+\frac{h_y(j)}{2}v\right)\hat{\lambda}_m^V\hat{\lambda}_n^V dv\right)\left(\int_I \hat{\lambda}_m^X\frac{d\hat{\lambda}_n^X}{dx}dx\right)u_h^m \\
&= \frac{h_y(j)}{2}\sum_{m=1}^{N_p^2}\left(IntM_{nm}+IntR_{nm}\right)SX_{nm}^T u_h^m
\end{aligned}$$

where

$$\begin{aligned}
VolV_{nm}^{(i,j)} &:= \int_{J_j}\int_{I_i} v\lambda_m\partial_x\lambda_n dx\, dv && (4.2.18)\\
SX_{nm} &:= \int_I \hat{\lambda}_n^V\frac{d\hat{\lambda}_m^V}{dx}dv && (4.2.19)\\
IntM_{nm} &:= c_y(j)\int_I \hat{\lambda}_m^V\hat{\lambda}_n^V dv && (4.2.20)\\
IntR_{nm} &:= \frac{h_y(j)}{2}\int_I v\hat{\lambda}_m^V\hat{\lambda}_n^V dv && (4.2.21)
\end{aligned}$$

As before we begin with $k=1$. Using identities in (4.2.4), we obtain

$$\mathbf{SX}^{2D} := \left[\begin{array}{c|c} \mathbf{S}^{1D} & \mathbf{S}^{1D} \\ \hline \mathbf{S}^{1D} & \mathbf{S}^{1D} \end{array}\right] \tag{4.2.22}$$

where as before $\mathbf{S}^{1D}$ is the gradient matrix we derived for the 1D case. Generalizing to the higher order (arbitrary k) reads

$$\begin{aligned}
\mathbf{S}^{1D} &\in \mathcal{M}_{N_p\times N_p}\\
\mathbf{F} &\in \mathcal{M}_{N_p\times N_p}\\
\mathbf{SX}^{2D} &:= \left[\begin{array}{c|c|c|c} \mathbf{S}^{1D} & \mathbf{S}^{1D} & \dots & \mathbf{S}^{1D} \\ \hline & & \vdots & \\ \hline \mathbf{S}^{1D} & \mathbf{S}^{1D} & \dots & \mathbf{S}^{1D} \end{array}\right]_{N_p^{2D}\times N_p^{2D}}
\end{aligned} \tag{4.2.23}$$

Moreover, the $\int_I c_y(j)\hat{\lambda}^V_m\hat{\lambda}^V_n dv$ term in the X-gradient equation is same as the second term in (4.2.11) multiplied by $c_y(j)$:

$$\mathbf{IntM} \;=\; c_y(j)\left[\begin{array}{c|c|c|c} M_{11}\mathbf{F} & M_{12}\mathbf{F} & \dots & M_{1,N_p}\mathbf{F} \\ \hline & & \vdots & \\ \hline M_{N_p,1}\mathbf{F} & M_{N_p,2}\mathbf{F} & \dots & M_{N_p,N_p}\mathbf{F} \end{array}\right] \tag{4.2.24}$$

Now the last step is to find a way to evaluate the last term, $\frac{h_y(j)}{2}\int_I v\hat{\lambda}^V_m\hat{\lambda}^V_n dv$. We are going to use some properties of the Legendre polynomials for evaluating this integral. We begin with the recursive formula that generates the Legendre polynomials:

$$\begin{aligned}(n+1)P_{n+1}(r) = (2n+1)rP_n(r) - nP_{n-1}(r) \quad \forall r \in I, n \geq 1 \\ P_0(r) = 1\end{aligned}$$

collecting terms without r, we get

$$\frac{n}{2n+1}P_{n-1}(r) + \frac{n+1}{2n+1}P_{n+1}(r) = rP_n(r) \tag{4.2.25}$$

Now for the moment, we focus on evaluating the following integral

$$MR_{nm} := \int_I r\hat{\lambda}_n(r)\hat{\lambda}_n(r)dr \quad n, m \in 1..N_p$$

using (4.1.12), we obtain

$$\int_I r\hat{\lambda}_n(r)\hat{\lambda}_m(r)dr \;=\; \sum_{q=1}^{N_p}\sum_{s=1}^{N_p}(V^T)^{-1}_{nq}(V^T)^{-1}_{ms}\int_I r\tilde{P}_{q-1}\tilde{P}_{s-1}dr$$

now expanding and applying (4.2.25), we obtain

$$\begin{aligned}\int_I r\,\tilde{P}_q\tilde{P}_s\,dr &= \int_I r\frac{P_q}{\sqrt{\frac{2}{2q+1}}}\frac{P_s}{\sqrt{\frac{2}{2s+1}}}dr \\ &= \frac{\sqrt{(2s+1)(2q+1)}}{2}\int_I rP_qP_sdr \\ &= \frac{\sqrt{(2s+1)(2q+1)}}{2}\int_I P_q\left[\frac{s}{2s+1}P_{s-1}(r) + \frac{s+1}{2s+1}P_{s+1}(r)\right]dr \\ &= \frac{\sqrt{(2s+1)(2q+1)}}{2}\left[\frac{s}{2s+1}\int_I P_qP_{s-1}(r)dr + \frac{s+1}{2s+1}\int_I P_qP_{s+1}(r)dr\right] \\ &= \frac{\sqrt{(2s+1)(2q+1)}}{2}\left[\frac{s}{2s+1}\frac{2}{2q+1}\delta_{q,s-1} + \frac{s+1}{2s+1}\frac{2}{2q+1}\delta_{q,s+1}\right] \\ &= \frac{1}{\sqrt{(2s+1)(2q+1)}}\left[s\delta_{q,s-1} + (s+1)\delta_{q,s+1}\right]\end{aligned}$$

and hence

$$\begin{aligned}MR_{nm} &= \sum_{s=1}^{N_p}\sum_{q=1}^{N_p}(V^T)^{-1}_{nq}(V^T)^{-1}_{ms}\int_I r\tilde{P}_{q-1}\tilde{P}_{s-1}dr \\ &= \sum_{s=1}^{N_p}\sum_{q=1}^{N_p}(V^T)^{-1}_{nq}(V^T)^{-1}_{ms}\frac{[(s-1)\delta_{q-1,s-2} + s\delta_{q-1,s}]}{\sqrt{(2s-1)(2q-1)}} \\ &= \sum_{s=1}^{N_p-1}(V^T)^{-1}_{n,s+1}(V^T)^{-1}_{ms}\frac{s}{\sqrt{(2s-1)(2s+1)}} + \\ &\quad \sum_{s=2}^{N_p}(V^T)^{-1}_{n,s-1}(V^T)^{-1}_{ms}\frac{s-1}{\sqrt{(2s-1)(2s-3)}}\end{aligned}$$

and finally using (4.2.4), $IntR_{nm}$ is the following matrix

$$\mathbf{IntR} := \frac{h_y(j)}{2}\left[\begin{array}{c|c|c|c} \mathbf{MR} & \mathbf{MR} & \dots & \mathbf{MR} \\ \hline & & \vdots & \\ \hline \mathbf{MR} & \mathbf{MR} & \dots & \mathbf{MR} \end{array}\right]_{N_p^{2D}\times N_p^{2D}} \tag{4.2.26}$$

Coming back to give a short notation for our main equation in a given element $el \equiv (i,j)$, we obtain

$$\mathbf{IntV}^{(j)} := \frac{h_y(j)}{2}\left(\mathbf{IntR}+\mathbf{IntM}\right), \tag{4.2.27}$$

$$\mathbf{VolV}^{(i,j)} = \mathbf{IntV}^{(j)} \otimes \mathbf{SX}^T. \tag{4.2.28}$$

V-boundry term matrix

for this part we assume that the sign of v during integration remain constant (either $v > 0$ or $v < 0$). We are dealing with

$$\int_{J_j} \widehat{vf}\lambda_n\Big|_{\partial I_i} dv$$

1. Case $v < 0$: then $\widehat{vf}\Big|_{\partial I_i} = v(f(x^+_{i+1/2}, v) - f(x^+_{i-1/2}, v))$. Now consider we are in an element (i,j), then we denote the solution in the element on the left, $(i-1,j)$ by f^{left} and also for the right element by $(i+1,j)$ by f^{right}.

$$\begin{aligned}\int_{J_j} \widehat{vf}\lambda_n\Big|_{\partial I_i} dv &= \int_{J_j} v[f(x^+_{i+1/2},v)\lambda_n(x_{i+1/2},v) - f(x^+_{i-1/2},v)\lambda_n(x_{i-1/2},v)]dv \\ &= \int_{J_j} v\, f(x^+_{i+1/2},v)\lambda_n(x_{i+1/2},v)dv - \int_{J_j} v\, f(x^+_{i-1/2},v)\lambda_n(x_{i-1/2},v)dv \\ &= \int_{J_j} v\, f^{\text{right}}(x_{i+1/2},v)\lambda_n(x_{i+1/2},v)dv - \int_{J_j} v\, f(x_{i-1/2},v)\lambda_n(x_{i-1/2},v)dv \\ &= \sum_{m=1}^{N_p^2}\int_{J_j} v\,\lambda_m^{\text{right}}(x_{i+1/2},v)\lambda_n(x_{i+1/2},v)u_h^{m,\text{right}}dv \\ &\quad -\sum_{m=1}^{N_p^2}\int_{J_j} v\,\lambda_m(x_{i+1/2},v)\lambda_n(x_{i+1/2},v)u_h^{m}dv\end{aligned}$$

Now we define these two terms (keep in mind $v < 0$) in the following way

$$\begin{aligned}VFR_n &:= \sum_{m=1}^{N_p^2}\int_{J_j} v\,\lambda_m^{\text{right}}(x_{i+1/2},v)\lambda_n(x_{i+1/2},v)u_h^{m,\text{right}}dv \\ &= \sum_{m=1}^{N_p^2}\left[\lambda_m^{X,\text{right}}(x_{i+1/2})\lambda_n^X(x_{i+1/2})\right]\left(\int_{J_j} v\,\lambda_m^{V,\text{right}}\lambda_n^V dv\right)u_h^{m,\text{right}} \\ &= \sum_{m=1}^{N_p^2} B_{nm}\left(\int_{J_j} v\,\lambda_m^{V,\text{right}}\lambda_n^V dv\right)u_h^{m,\text{right}}\end{aligned}$$

where

$$B_{nm} = \lambda_m^{X,\text{right}}(x_{i+1/2})\lambda_n^X(x_{i+1/2}) \tag{4.2.29}$$

As before, we begin with the simplest case $k = 1$, and using (4.2.4). But we should also mention some more identities as:

$$\begin{array}{l|l} \lambda_1^{V,\mathrm{right}} = \lambda_2^V & \lambda_1^{X,\mathrm{right}}(x_{i+1/2}) = \lambda_2^X(x_{i+1/2}) = 1 \\ \lambda_3^{V,\mathrm{right}} = \lambda_4^V & \lambda_3^{V,\mathrm{right}}(x_{i+1/2}) = \lambda_4^V(x_{i+1/2}) = 1 \\ & \lambda_2^{X,\mathrm{right}}(x_{i+1/2}) = \lambda_1^X(x_{i+1/2}) = 0 \\ & \lambda_4^{X,\mathrm{right}}(x_{i+1/2}) = \lambda_3^X(x_{i+1/2}) = 0 \end{array} \tag{4.2.30}$$

and hence

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \\ VFR_n &= \sum_{m=1}^{N_p^2} B_{nm} \left(\int_{J_j} v \lambda_m^{V,\mathrm{right}} \lambda_n^V dv \right) u_h^{m,\mathrm{right}} \\ \mathbf{VFR} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ IntV_{22} & 0 & IntV_{24} & 0 \\ 0 & 0 & 0 & 0 \\ IntV_{42} & 0 & IntV_{44} & 0 \end{bmatrix} . \mathbf{u}_h^{\mathrm{right}} \end{aligned}$$

Now we focus on the easier term:

$$\begin{aligned} VFL_n &:= \sum_{m=1}^{N_p^2} \left(\int_{J_j} v \lambda_m(x_{i-1/2}, v) \lambda_n(x_{i-1/2}, v) dv \right) u_h^m \\ &= \sum_{m=1}^{N_p^2} \left[\lambda_m^X(x_{i-1/2}) \lambda_n^X(x_{i-1/2}) \right] \underbrace{\left(\int_{J_j} v \lambda_m^V \lambda_n^V dv \right)}_{IntV_{nm}} u_h^m \end{aligned}$$

$$\mathbf{VFL} = \begin{bmatrix} IntV_{11} & 0 & IntV_{13} & 0 \\ 0 & 0 & 0 & 0 \\ IntV_{31} & 0 & IntV_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \mathbf{u}_h \tag{4.2.31}$$

and hence the total expression for V-boundary term will be

$$\int_{J_j} \widehat{vf} \lambda_n \Big|_{\partial I_i} dv = VFR_n - VFL_n \quad \forall n = 1..N_p^2 \tag{4.2.32}$$

In order to get a general expression that works for any degree (arbitrary k), we have to generalize some concepts regarding the basis functions. The basis functions, $\{\hat{\lambda}_n\}$, we introduced in the reference element I^2 is related to the set of faces $\{f_n\}$ in the reference element shown in §4.2.2, where $\hat{\lambda}_n(\xi_m) = \delta_{nm}$ for all $n, m \in 1..N_p^2$. In this reference element we defined four class of faces, which correspond to the faces on the boundary of the element. Coming back to calculate **VFR** terms, we will get

$$B_{nm} = \lambda_m^{X,\mathrm{right}}(x_{i+1/2}) \lambda_n^X(x_{i+1/2}) = \begin{cases} 1 & n \in \mathbf{faces_{right}}, m \in \mathbf{faces_{left}} \\ 0 & \text{else} \end{cases}$$

and

$$VFR_{nm} = B_{nm} \int_{J_j} v \lambda_m^{V,\mathrm{right}} \lambda_n^V dv \quad = \quad \begin{cases} IntV_{nm} & n, m \in \mathbf{faces_{right}} \\ 0 & \text{else} \end{cases}$$

Moreover for the **VFL** terms we obtain

$$B'_{nm} := \lambda_m^X(x_{i-1/2}) \lambda_n^X(x_{i-1/2}) \quad = \quad \begin{cases} 1 & n, m \in \mathbf{faces_{left}} \\ 0 & \text{else} \end{cases} \tag{4.2.33}$$

$$\mathbf{VFL} = \mathbf{IntV} \otimes \mathbf{B}' \tag{4.2.34}$$

and finally

$$\int_{J_j} \widehat{vf}\lambda_n\Big|_{\partial I_i} dv = VFR_n - VFL_n \quad \forall n = 1..N_p^2 \tag{4.2.35}$$

2. Case $v > 0$: then $\widehat{vf}\Big|_{\partial I_i} = v(f(x^-_{i+1/2}, v) - f(x^-_{i-1/2}, v))$. Using the same procedure as above for calculating **VFL** and **VFR**, we will obtain

$$B_{nm} = \lambda_m^{X,\text{left}}(x_{i+1/2})\lambda_n^X(x_{i+1/2}) = \begin{cases} 1 & n \in \textbf{faces}_{\textbf{left}}, m \in \textbf{faces}_{\textbf{right}} \\ 0 & \text{else} \end{cases}$$

and

$$VFL_{nm} = B_{nm}\int_{J_j} v\lambda_m^{V,\text{left}}\lambda_n^V dv = \begin{cases} IntV_{nm} & n, m \in \textbf{faces}_{\textbf{left}} \\ 0 & \text{else} \end{cases}$$

Moreover for the **VFR** terms we obtain:

$$B'_{nm} := \lambda_m^X(x_{i-1/2})\lambda_n^X(x_{i-1/2}) \quad = \quad \begin{cases} 1 & n, m \in \textbf{faces}_{\textbf{right}} \\ 0 & \text{else} \end{cases} \tag{4.2.36}$$

$$\mathbf{VFR} \quad = \quad \mathbf{IntV} \otimes \mathbf{B}' \tag{4.2.37}$$

$E(x)$-boundry term matrix

Upto now, all terms could be calculated with little effort but the $E(x)$-boundary term is a little bit crucial because

$$\widehat{Ef}(v_{j+\frac{1}{2}}) \quad := \quad \begin{cases} E(x)f(x, v^+_{j+\frac{1}{2}}) & E(x) \geq 0 \\ E(x)f(x, v^-_{j+\frac{1}{2}}) & E(x) \leq 0 \end{cases}$$

$E(x)$ can change sign inside the interval. Hence

$$\int_{I_i} \widehat{Ef}\varphi\Big|_{\partial J_j} dx$$

cannot be calculated as before, since the integrand is defined in two different ways. One idea is to find roots of $E(t, x)$ and decompose the integral into

$$\begin{aligned} \int_{I_i} \widehat{Ef}\varphi\Big|_{\partial J_j} dx \quad &= \quad \int_{I_i^+} E(x)f^{\text{up}}(x, v_{j+1/2})\varphi(x, v_{j+1/2})dx + \int_{I_i^-} E(x)f(x, v_{j+1/2})\varphi(x, v_{j+1/2})dx \\ &- \quad \left(\int_{I_i^+} E(x)f(x, v_{j-1/2})\varphi(x, v_{j-1/2})dx + \int_{I_i^-} E(x)f^{\text{down}}(x, v_{j-1/2})\varphi(x, v_{j-1/2})dx\right) \end{aligned} \tag{4.2.38}$$

where

$$\begin{aligned} I_i^+ &= \{x | x \in I_i, E(x) \geq 0\} \\ I_i^- &= \{x | x \in I_i, E(x) \leq 0\} \end{aligned} \tag{4.2.39}$$

Moreover, If the $E(t, x)$ is known then one may find its roots analytically (but in very exceptional cases). Note that in V-P system the $E(t, x)$ is even not known.

In order to compute this term one may use a brute force procedure to find approximately the $E(x)$ roots in each interval and decompose domain to I_i^+ and I_i^- and calculate (4.2.38). Below, we begin to treat this integral by assuming that $E(x)$ does not change sign within an interval where we have an exact expression for any k. Later we will introduce two different methods to handle the problem of discontinuity of the integrand in an interval.

1. $E(x) > 0$ then

$$\begin{aligned}
\int_{I_i} \widehat{Ef}\lambda_n\Big|_{\partial J_j} dx &= \int_{I_i} E(x) f^{\mathrm{up}}(x, v_{j+1/2})\lambda_n(x, v_{j+1/2}) dx \\
&- \int_{I_i} E(x) f(x, v_{j-1/2})\lambda_n(x, v_{j-1/2}) dx \\
&= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x)\lambda_m^{\mathrm{up}}(x, v_{j+1/2})\lambda_n(x, v_{j+1/2}) dx \right) u_h^m \\
&- \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x)\lambda_m(x, v_{j-1/2})\lambda_n(x, v_{j-1/2}) dx \right) u_h^m \quad \forall n = 1..N_p^2
\end{aligned}$$

Defining ‘up’ and ‘down’ terms

$$\begin{aligned}
EFU_n &:= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x)\lambda_m^{\mathrm{up}}(x, v_{j+1/2})\lambda_n(x, v_{j+1/2}) dx \right) u_h^m & (4.2.40) \\
&= \sum_{m=1}^{N_p^2} \left[\lambda_m^{V,\mathrm{up}}(v_{j+1/2})\lambda_n^V(v_{j+1/2}) \right] \left(\int_{I_i} E(x)\lambda_m^{X,\mathrm{up}}\lambda_n^X dx \right) u_h^m & (4.2.41)
\end{aligned}$$

then

$$\begin{aligned}
\left[\lambda_m^{V,\mathrm{up}}(v_{j+1/2})\lambda_n^V(v_{j+1/2}) \right] &= \begin{cases} 1 & n \in \mathbf{faces_{up}}, m \in \mathbf{faces_{down}} \\ 0 & \text{else} \end{cases} & (4.2.42) \\
\left(\int_{I_i} E(x)\lambda_m^{X,\mathrm{up}}\lambda_n^X dx \right) &= IntE_{n,m} \quad \forall n, m \in \mathbf{faces_{up}} & (4.2.43)
\end{aligned}$$

and for $v_{j-1/2}$ (down)

$$\begin{aligned}
EFD_n &:= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x)\lambda_m(x, v_{j-1/2})\lambda_n(x, v_{j-1/2}) dx \right) u_h^m & (4.2.44) \\
&= \sum_{m=1}^{N_p^2} \left[\lambda_m^V(v_{j-1/2})\lambda_n^V(v_{j-1/2}) \right] \left(\int_{I_i} E(x)\lambda_m^X\lambda_n^X dx \right) u_h^m & (4.2.45)
\end{aligned}$$

then

$$\begin{aligned}
\left[\lambda_m^V(v_{j-1/2})\lambda_n^V(v_{j-1/2}) \right] &= \begin{cases} 1 & n, m \in \mathbf{faces_{down}} \\ 0 & \text{else} \end{cases} & (4.2.46) \\
\left(\int_{I_i} E(x)\lambda_m^X\lambda_n^X dx \right) &= IntE_{n,m} & (4.2.47)
\end{aligned}$$

finally

$$\int_{I_i} \widehat{Ef}\lambda_n\Big|_{\partial J_j} dx = EFU_n - EFD_n \quad \forall n = 1..N_p^2 \qquad (4.2.48)$$

2. $E(x) < 0$ then

$$\begin{aligned}\int_{I_i} \widehat{Ef}\lambda_n\Big|_{\partial J_j} dx &= \int_{I_i} E(x) f(x, v_{j-1/2}) \lambda_n(x, v_{j+1/2}) dx \\ &- \int_{I_i} E(x) f^{\text{down}}(x, v_{j+1/2}) \lambda_n(x, v_{j-1/2}) dx \\ &= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x) \lambda_m(x, v_{j+1/2}) \lambda_n(x, v_{j+1/2}) dx \right) u_h^m \\ &- \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x) \lambda_m^{\text{down}}(x, v_{j-1/2}) \lambda_n(x, v_{j-1/2}) dx \right) u_h^m \quad \forall n = 1..N_p^2\end{aligned}$$

Defining 'up' and 'down' terms

$$\begin{aligned}EFU_n &:= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x) \lambda_m(x, v_{j+1/2}) \lambda_n(x, v_{j+1/2}) dx \right) u_h^m && (4.2.49) \\ &= \sum_{m=1}^{N_p^2} \left[\lambda_m^V(v_{j+1/2}) \lambda_n^V(v_{j+1/2}) \right] \left(\int_{I_i} E(x) \lambda_m^X \lambda_n^X dx \right) u_h^m && (4.2.50)\end{aligned}$$

then

$$\begin{aligned}\left[\lambda_m^V(v_{j+1/2}) \lambda_n^V(v_{j+1/2}) \right] &= \begin{cases} 1 & n, m \in \mathbf{faces_{up}} \\ 0 & \text{else} \end{cases} && (4.2.51) \\ \left(\int_{I_i} E(x) \lambda_m^X \lambda_n^X dx \right) &= IntE_{n,m} \quad \forall n, m && (4.2.52)\end{aligned}$$

and for $v_{j-1/2}$ (down)

$$\begin{aligned}EFD_n &:= \sum_{m=1}^{N_p^2} \left(\int_{I_i} E(x) \lambda_m^{\text{down}}(x, v_{j-1/2}) \lambda_n(x, v_{j-1/2}) dx \right) u_h^m && (4.2.53) \\ &= \sum_{m=1}^{N_p^2} \left[\lambda_m^{V,\text{down}}(v_{j-1/2}) \lambda_n^V(v_{j-1/2}) \right] \left(\int_{I_i} E(x) \lambda_m^{X,\text{down}} \lambda_n^X dx \right) u_h^m && (4.2.54)\end{aligned}$$

then

$$\begin{aligned}\left[\lambda_m^V(v_{j-1/2}) \lambda_n^V(v_{j-1/2}) \right] &= \begin{cases} 1 & n \in \mathbf{faces_{down}}, m \in \mathbf{faces_{up}} \\ 0 & \text{else} \end{cases} && (4.2.55) \\ \left(\int_{I_i} E(x) \lambda_m^X \lambda_n^X dx \right) &= IntE_{n,m} \quad \forall n, m \in \mathbf{faces_{down}} && (4.2.56)\end{aligned}$$

hence

$$\int_{I_i} \widehat{Ef}\lambda_n\Big|_{\partial J_j} dx = EFU_n - EFD_n \quad \forall n = 1..N_p^2 \tag{4.2.57}$$

Approximating $E(x)$-boundary term

If $E_h(x) = 0$ for some $x \in I_i$ then the definition of the numerical flux (3.1.7) would require finding roots of $E_h(x)$ in I_i and then partitioning the interval into I_i^+ and I_i^-, introduced in (4.2.39). Denoting $f(x, v^+_{j+\frac{1}{2}})$ by f^+ and $f(x, v^-_{j+\frac{1}{2}})$ by f^-, we would have ideally

$$\int_{I_i} \widehat{Ef}\,\varphi = \int_{I_i^+} E_h(x) f^+\,\varphi + \int_{I_i^-} E_h(x) f^-\,\varphi \tag{4.2.58}$$

In practice E_h is a polynomial of degree $k+1$ and therefore except in very particular situations ($k=1$) finding its roots could be extremely difficult. We now introduce two methods to deal with this difficulty and as we will show in §5.3.3 they are very efficient to compute the expression (4.2.58).

One should mention that in order to determine whether or not $E_h(x)$ has some roots in I_i, we use a simple controller that checks if electric field is strictly positive or negative. The check is done in the following way, we first express $E_h(x)$ in an element I_i using Bernstein polynomials of degree $k+1$

$$E_h(t,x) \quad := \quad \sum_{n=1}^{N_p+1} \widetilde{E}_i^n(t) B_{k+1}^n(x) \quad x \in I_i$$

where

$$B_{k+1}^n(x) \geq 0 \quad \forall n$$

Hence if all coefficients, $\{\widetilde{E}_i^n\}$, are positive (or all negative) then we can conclude that $E_h(x) \neq 0$ for $x \in I_i$. But if at least one of them changes sign then, $E_h(x)$ vanishes in I_i, we need to compute the flux in a different way. We now consider the following two approaches.

1. *Projection to $\mathcal{P}^0$*

   A very simple idea is to project the electric field function onto the piecewise constants in the element where $E_h(x)$ changes sign (note we do the projection only in those elements where $E_h(x)$ changes sign not all). In that way we guarantee that $\mathbb{P}_0(E_h)$ is constant along I_i. The new numerical flux reads

$$\widehat{Ef}(x, v_{j+\frac{1}{2}}) \quad := \quad \begin{cases} E_h(x)\, f^h(x, v^+_{j+\frac{1}{2}}) & E_h(x) \geq 0 \\ E_h(x)\, f^h(x, v^-_{j+\frac{1}{2}}) & E_h(x) \leq 0 \\ \mathbb{P}_0(E_h) f^+ & \text{if}\, \exists\, x^* : E_h(x*) = 0 \,\text{and}\, \mathbb{P}_0(E_h) > 0 \\ \mathbb{P}_0(E_h) f^- & \text{if}\, \exists\, x^* : E_h(x*) = 0 \,\text{and}\, \mathbb{P}_0(E_h) < 0 \end{cases} . \tag{4.2.59}$$

   Therefore with this definition, we can proceed as before to compute the X-boundary term. We will show in numerical experiments that this approach should be used only for low order approximations.

2. *Weighted average*

   If $E_h(t,x) = 0$ for some $x \in I_i$ then we use the following definition for numerical flux

$$\widehat{Ef}(x, v_{j+\frac{1}{2}}) \quad := \quad \begin{cases} E_h(x) f(x, v^+_{j+\frac{1}{2}}) & \forall x \in I_i, E_h(x) > 0 \\ E_h(x) f(x, v^-_{j+\frac{1}{2}}) & \forall x \in I_i, E_h(x) < 0 \\ \{\{Ef\}\}_{\mathbf{w}} & \exists x^* \in I_i, E_h(x^*) = 0 \end{cases} \tag{4.2.60}$$

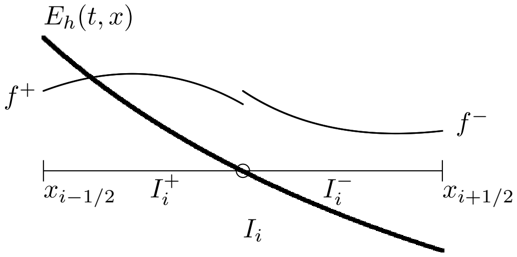
   where $\{\{Ef\}\}_{\mathbf{w}}$ is a weighted average function that we define as

$$\{\{Ef\}\}_{\mathbf{w}} = w^+\, E_h(x) f(x, v^+_{j+\frac{1}{2}}) + w^-\, E_h(x) f(x, v^-_{j+\frac{1}{2}}) \tag{4.2.61}$$
$$(w^+ + w^- = 1)$$

   Now the question is how to determine the optimal values for $w^\pm$ that minimize the difference between the new flux, $\widehat{Ef}$, and the old one in the sense of (4.2.58). From (4.2.58) we like to choose

$$\begin{aligned} w^+ \quad &\approx \quad |I_i^+| \\ w^- \quad &\approx \quad |I_i^+|. \end{aligned}$$

   However, obtaining information about $|I_i^+|$ or $|I_i^-|$ is as difficult as approximating the zeros of E_h (which we do not want to do). The idea now is to relate $|I_i^\pm|$ to computable quantities. To explain how $w^\pm$ are picked in practice, consider an element, $I_i = I_i^+ \cup I_i^-$, and the electric field behaves like



Since E_h is a polynomial (smooth function) then $|I_i^-|$ will be proportional to the $|\min_{I_i} E_h|$. Similarly $|I_i^+|$ is proportional to $|\max_{I_i} E_h|$. Hence we choose

$$\begin{aligned} w^+ &= \frac{|\max_{I_i} E_h|}{|\max_{I_i} E_h| + |\min_{I_i} E_h|} \\ w^- &= \frac{|\min_{I_i} E_h|}{|\max_{I_i} E_h| + |\min_{I_i} E_h|} \end{aligned} \tag{4.2.62}$$

with the property $w^+ + w^- = 1$ (see Appendix A). Note that (4.2.58) becomes

$$\int_{I_i} \widehat{Ef}\,\varphi = w^+ \int_{I_i} E_h(x) f^+\,\varphi + w^- \int_{I_i} E_h(x) f^-\,\varphi.$$

Moreover we wish to stress that obtaining w^+ and w^- is low cost. We will see in the following chapter that using this approximation will give really good result in simulation of V-P systems. Also the result for energy conservation is better than projecting to $\mathcal{P}^0$ approximation for several order of magnitudes (beside projecting to constant preserves energy better than other methods applied to V-P system in literature).

Chapter 5

Numerical experiments

In this section, we apply the numerical scheme that was described in the previous sections to a group of test cases that are generally used to verify the reliability and efficiency of the numerical schemes. We first begin with a linear transport problem to check the accuracy of the Vlasov solver then in §5.2, we check the convergence rate of the Vlasov-Poisson solver for a forced V-P system.

Pursuing this further, we also examine the method for two typical test cases, non-linear Landau damping in §5.3 and two stream instability in §5.5 where we compared the conservation of energy and L^p-norms for different Poisson solvers and the effect of mesh refinement and increase the degree of polynomials on conservation of those quantities.

$$
\begin{aligned}
&\mathcal{E}_k(t) = \int_{\Omega_x}\int_{\Omega_v} \frac{|v|^2}{2} f_h(t,x,v)dv\,dx && \text{discrete kinetic energy}\\
&\mathcal{E}_p(t) = \frac{1}{2}\int_{\Omega_x} |E_h(t,x)|^2 dx + \frac{(k+1)^2}{h_x}\sum_{i=0}^{N_x} [\![\Phi_h]\!]^2_{i+1/2} && \text{discrete potential energy}\\
&\mathcal{E}_{tot}(t) = \mathcal{E}_k(t) + \mathcal{E}_p(t) && \text{discrete total energy}\\
&\|f_h\|^p_{L^p} = \int_{\Omega_x}\int_{\Omega_v} |f_h(t,x,v)|^p \, dx\, dv && \text{discrete } L^p \text{ norm}\\
&\mathcal{E}^0_{tot} := \int_{\Omega_x}\int_{\mathbb{R}} \frac{|v|^2}{2} f_0(x,v)dx\,dv + \frac{1}{2}\int_{\Omega_x} |E_0(x)|^2 dx && \text{exact initial energy}\\
&\|f_0\|^p_{L^p} = \int_{\Omega_x}\int_{\mathbb{R}} |f_0(x,v)|^p \, dx\, dv && \text{exact initial } L^p \text{ norm}
\end{aligned}
\tag{5.0.1}
$$

In addition we show the effect of different approaches in evaluating $E(x)$-boundary term in §4.2.4 on physical quantities like total energy. For all test cases, we use a Runge-Kutta time integrator, so-called RK4 which is a fourth order integrator. Although this kind of integrator, RK4, is explicit and generally not energy preserving for physical systems, we prefer to use and avoid using energy preserving symplectic Runge-Kutta methods since symplectic RK methods are all implicit and expensive. We also mention that Poisson equation is solved in each stage of the Runge-Kutta time integration (not in each time step) and applied in the Vlasov equation in that stage.

In the following sections, whenever we solved the V-P system, we use the energy conservative LDG solver for Poisson equation in §3.2.2 and also “weighted average” flux when electric field changes sign in an element unless otherwise stated.

5.1 Simple linear advection

In order to check the DG scheme for the Vlasov equation (transport equation) with a given electric field and reliability of the algorithm, we perform a convergence test on the code with the following initial data

$$f(t=0,x,v)=f_0(x,v)=\sin(\pi x)e^{-\frac{v^2}{2}} \quad \forall x\in\Omega_x=[-1,1], v\in\Omega_v=[-10,10] \tag{5.1.1}$$

where the boundaries of the Ω_v are chosen such that $f_0(x,v)|_{v=\partial\Omega_v}\approx 0$ to ensure the compact support of the f_0 and so the solution.

5.1.1 $E(x)$ constant

First, we choose a constant electric field, $E(x)=5$. In this example, as the electric field is chosen constant, we can find the exact solution by solving the characteristics of (4.2.1):

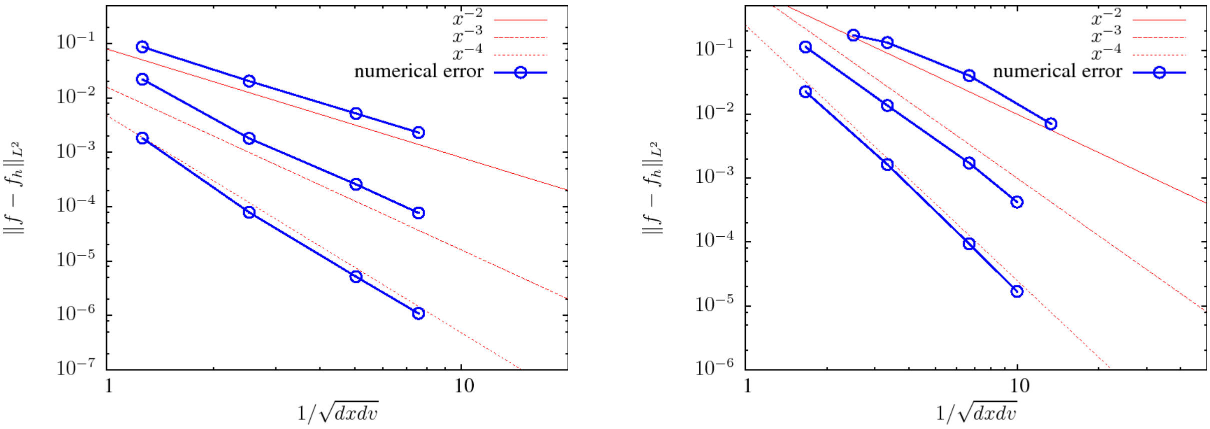


Figure 5.1: log-log plot of $x^{-(k+1)}$ (dashed), L^2 error of approximate solution (solid) using polynomials of degree $k=1,2,3$ where $E(x)=5$ (left) and $E(x)=x$ (right).

$$f(t,x,v)=f_0(x-vt-\frac{1}{2}Et^2, v+Et) \quad \forall x\in\Omega_x=[-1,1], v\in\Omega_v=[-10,10], t\geq 0 \tag{5.1.2}$$

We performed tests with this initial condition using polynomials of degree 1, 2 and 3. In Fig. 5.1(left), one may find accuracy of the scheme for $k=1,2,3$ which yields order of accuracy 2, 3 and 4.

5.1.2 $E(x)=x$

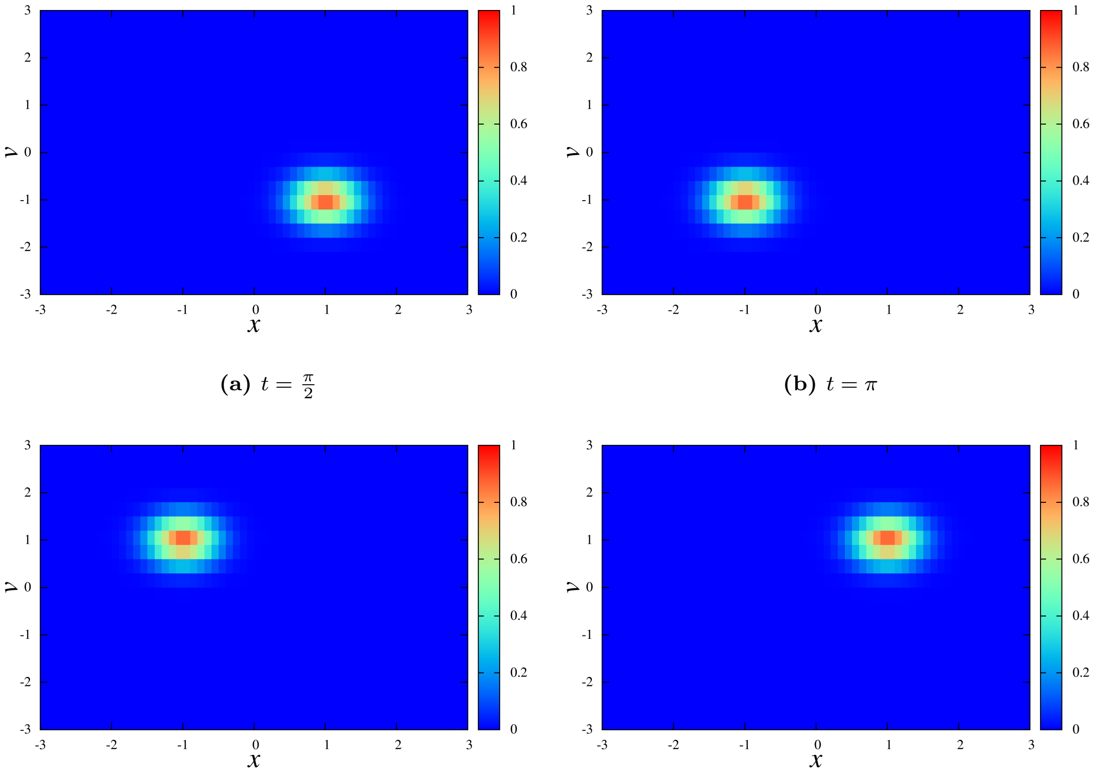
The second test we perform to check the accuracy of transport problem is formulated by choosing $E(x)=x$. Hence the Vlasov equation reads

$$\begin{aligned} f_t+vf_x-E(x)f_v &= 0 \\ f_t+vf_x-xf_v &= 0 \\ f_t+\mathbf{w}.\nabla f &= 0, \quad \mathbf{w}=(v,-x) \end{aligned}$$

note that $\mathbf{w}$ is a rotation vector that force the initial data to revolve around $(0,0)$ clockwise. The period of a complete rotation is 2π, so we can measure the error by calculating $\|f_h(0,.,.)-f_h(2\pi,.,.)\|_{L^2}$. The initial data is a 2D Gaussian function centered at $(1,1)$

$$f_0(x,v)=e^{-10\{(x-1)^2+(v-1)^2\}}, \quad \forall x\in\Omega_x=[-6,6], v\in\Omega_v=[-6,6] \tag{5.1.3}$$

We perform some simulation using polynomials of different degrees and a computational domain $[-6,6]\times[-6,6]$ upto $t = 2\pi$. At this time, we expect the approximate solution to be close to $f_h(0,.,.)$ in phase space. In Fig. 5.1(right) are given the convergence results for $k = 1, 2, 3$. In Fig. 5.2, we show the evolution of solution in phase space for $t = \pi/2$ (centered at $(1,-1)$), $t = \pi$ (centered at $(-1,-1)$), $t = 3\pi/2$ (centered at $(-1,1)$) and $t = 2\pi$ (centered at $(1,1)$).



(a) $t = \frac{\pi}{2}$

(b) $t = \pi$

(c) $t = \frac{3\pi}{2}$

(d) $t = 2\pi$

Figure 5.2: solution of the Vlasov equation with $E(x) = x$ at different times.

5.2 Convergence of Vlasov-Poisson

In this section we study the accuracy of DG-DG scheme for nonlinear Vlasov-Poisson system. We are able to perform this check by adding a right-hand side to the Vlasov equation and solving the following system:

$$\begin{cases} f_t + vf_x - E(t,x)f_v = \psi(t,x,v) \\ -\frac{\partial}{\partial x}E(t,x) = \rho(t,x) - \sqrt{\pi} \end{cases} \tag{5.2.1}$$

where

$$\psi(t,x,v) = e^{-\frac{1}{4}(4v-1)^2}\left(\left\{(4\sqrt{\pi}+2)v - (2\pi+\sqrt{\pi})\right\}\sin(2(x-\pi t)) + \left\{\frac{\sqrt{\pi}}{4} - \sqrt{\pi}v\right\}\sin(4(x-\pi t))\right)$$

then the general solution for this system will be:

$$
\begin{aligned}
f(t,x,v) &= \{2-\cos(2x-2\pi t)\}\, e^{-\frac{1}{4}(4v-1)^2} \qquad (5.2.2)\\
E(t,x) &= \frac{\sqrt{\pi}}{4}\sin(2x-2\pi t) \qquad (5.2.3)
\end{aligned}
$$

Notice that the general solution is periodic in time which means the solution will be at the same place at $t = 1$

$$f(t=1,x,v) = f(t=0,x,v)$$

Therefore, we performed the simulation until $t = 1$ on a computational domain $[-\pi,\pi]\times[-4,4]$ using polynomials of degree $k = 1, 2, ..., 6$. The results for accuracy can be found in Fig. 5.3(left). Moreover we show the convergence diagram varying the polynomial degree ($k = 1, 2, 4, 8, 12$) for different mesh 20×20, 40×40 and 80×80 in Fig. 5.3(right). From the latter convergence diagram it can be observed the exponential convergence as we plot it in a semi-log scale.

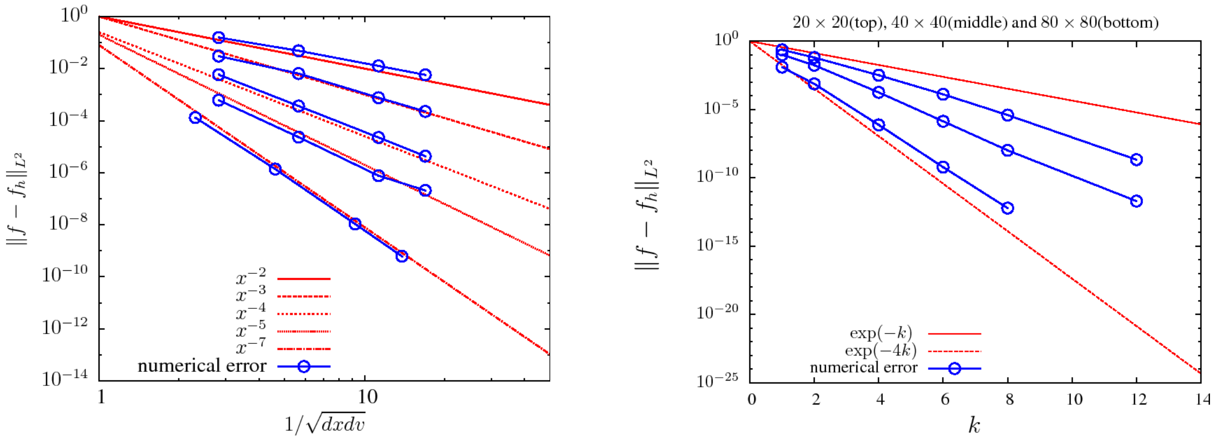


Figure 5.3: log-log plot of L^2 error for different mesh size (left) and different polynomial degree (right) for V-P system with right-hand side.

In order to check the conservative properties of the method, we perform a check on the physical quantities such as L^∞, L^2, mass $m(t)$ and total energy $\mathcal{E}_{tot}(t)$. We use $k = 6$ with a mesh 40×40. In Fig. 5.4, we show the relative error of the mass conservation (left-top), L^∞ (right-top), total energy (bottom-left) and L^2 (bottom-right). One may note that the errors are close to machine precision.

5.2.1 Convergence rate for different Poisson solver

In this section we compare the convergence rate for different Poisson solver applied to the V-P system. we use different meshes, 20×20, 40×40 and 80×80 together with polynomial degrees, $k = 2, 3$. In Table 5.1, is given the L^2 error for mixed-FEM, LDG2 and LDG3 Poisson solver (introduced in §3.2) where we do not see significant difference either in error or in the order of accuracy. But one may notice that LDG3 solver (the energy conservative method) seems to be more accurate since produce smaller errors.

5.3 Nonlinear Landau damping (strong case)

Nonlinear Landau damping is commonly used to check the reliability of the V-P system solvers since the properties of the L^2 norm of the electric field ($\sqrt{2\mathcal{E}_p(t)}$) are well known [7]. The initial data that we will

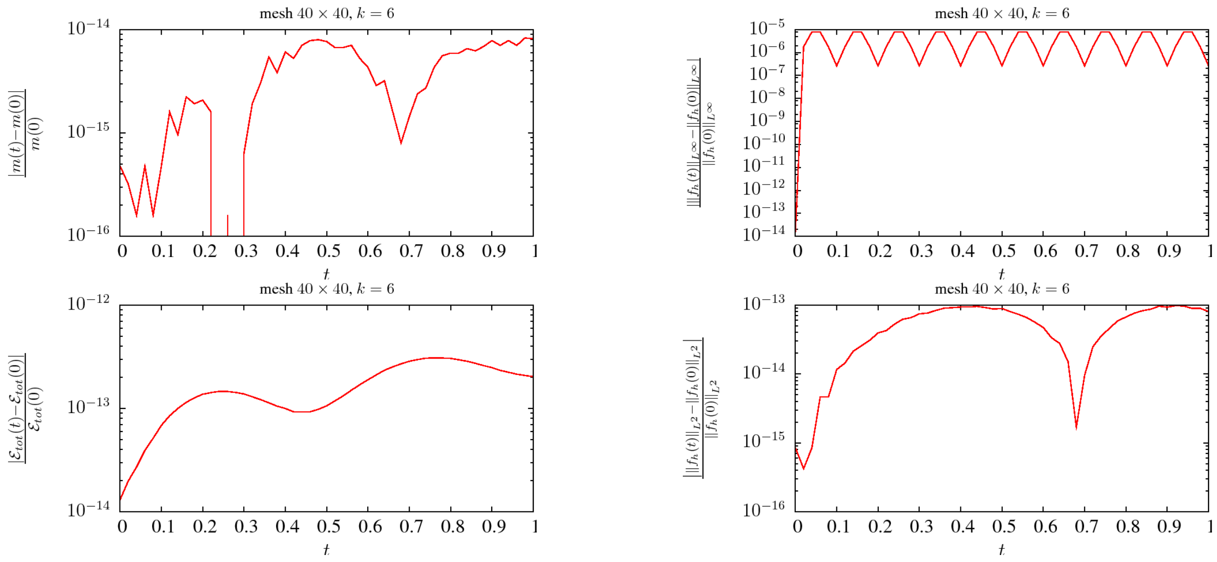


Figure 5.4: relative error for L^∞, L^2, mass $m(t)$ and total energy $\mathcal{E}_{tot}(t)$ for V-P system with *rhs* test case.

$k = 2$

mesh	L^2 error (mixed)	order	L^2 error (LDG3)	order	L^2 error (LDG2)	order
20^2	3.0154×10^{-2}	-	3.0134×10^{-2}	-	3.0134×10^{-2}	-
40^2	6.4640×10^{-3}	2.221873	6.4623×10^{-3}	2.2213132	6.4623×10^{-3}	2.2213157
80^2	7.5804×10^{-4}	3.092085	7.5775×10^{-4}	3.0922407	7.5775×10^{-4}	3.0922409

$k = 3$

mesh	L^2 error (mixed)	order	L^2 error (LDG3)	order	L^2 error (LDG2)	order
20^2	5.8300890×10^{-3}	-	5.8295897×10^{-3}	-	5.8295895×10^{-3}	-
40^2	3.6364382×10^{-4}	4.00291991	3.6361315×10^{-4}	4.00291807	3.6361315×10^{-4}	4.00291802
80^2	2.2582894×10^{-5}	4.00922378	2.2580954×10^{-5}	4.00922603	2.2580954×10^{-5}	4.00922603

Table 5.1: Order of accuracy using different Poisson solver for $k = 2, 3$ for V-P system with *rhs*.

use for this case is

$$f(t=0,x,v) = \frac{1}{\sqrt{2\pi}}\left(1+\alpha\cos(K\,x)\right)e^{-\frac{v^2}{2}} \tag{5.3.1}$$

where $K = 0.5$ and $\alpha = 0.5$. The Ω_x is $[0, 4\pi]$ and typically in literature the Ω_v is set to be $[-5, 5]$. We found that as $f(t, x, v)|_{v=\partial\Omega_v} \approx 10^{-5}$, so for high accurate methods as those considered here the solution cannot be longer considered of compact support. Therefore we consider $\Omega_v = [-10, 10]$ where $f(t, x, v)|_{v=\partial\Omega_v} \approx 10^{-22}$. We plot the evolution of the electric field norm in a semi-log scale in Fig. 5.5(left) for a mesh 100×160 using polynomials of degree 3 ($k = 3$). We also present fitted functions of the form $c\exp(-\gamma t)$ on the local maximums of $\|E(t, .)\|_{L^2}$ for $t \in [0, 10]$ (initial decay) and $t \in [20, 40]$ where the coefficients are estimated as $c_1 = 2.383814$, $\gamma_1 = -0.305920$, $c_2 = 0.015360$ and $\gamma_2 = 0.085241$. We also investigate the conservation properties of our scheme on the evolution of the L^1-norm and L^2-norm of the solution using

$$\frac{|\|f_h(t)\|_{L^1} - \|f_h(t=0)\|_{L^1}|}{\|f_h(t=0)\|_{L^1}} \tag{5.3.2}$$

$$\frac{|\|f_h(t)\|_{L^2} - \|f_h(t=0)\|_{L^2}|}{\|f_h(t=0)\|_{L^2}} \tag{5.3.3}$$

and the evolution in time of these quantities is depicted in Fig. 5.6.

As we claim that our scheme will preserve the total energy, $\mathcal{E}_{tot}(t)$, we plot the long-run behavior of

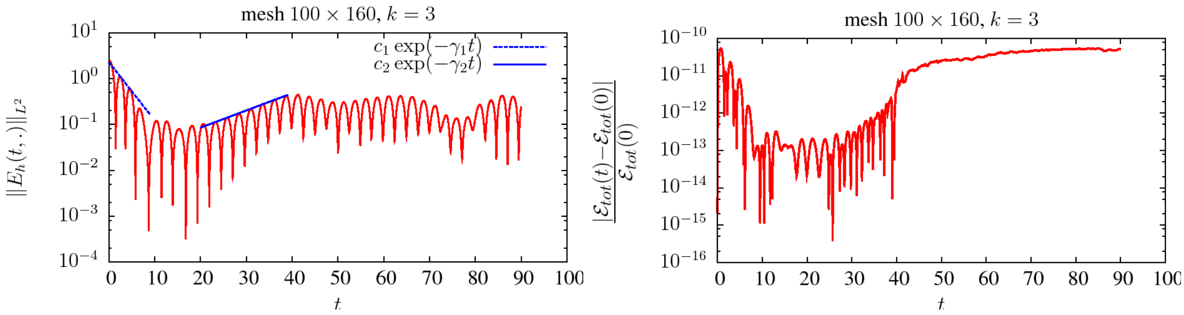


Figure 5.5: (left) The evolution of the electric field L^2-norm in a semi-log scale where $c_1 = 2.279673$, $\gamma_1 = -0.292285$, $c_2 = 0.015228$ and $\gamma_2 = 0.086126$. (right) The evolution of total energy error in a semi-log scale for non-linear Landau damping.

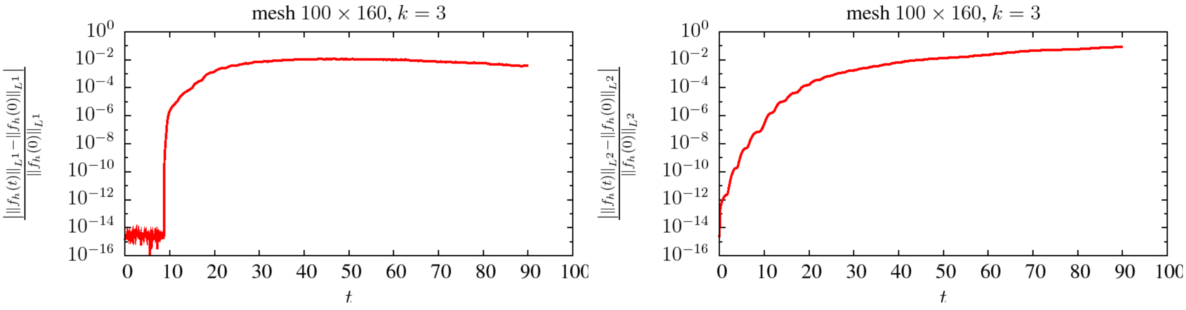


Figure 5.6: The evolution of $\|f_h\|_{L^1}$ and $\|f_h\|_{L^2}$ in a semi-log scale where using a mesh 100×160.

total energy error using

$$\frac{|\mathcal{E}_{tot}(t) - \mathcal{E}_{tot}(0)|}{\mathcal{E}_{tot}(0)} \tag{5.3.4}$$

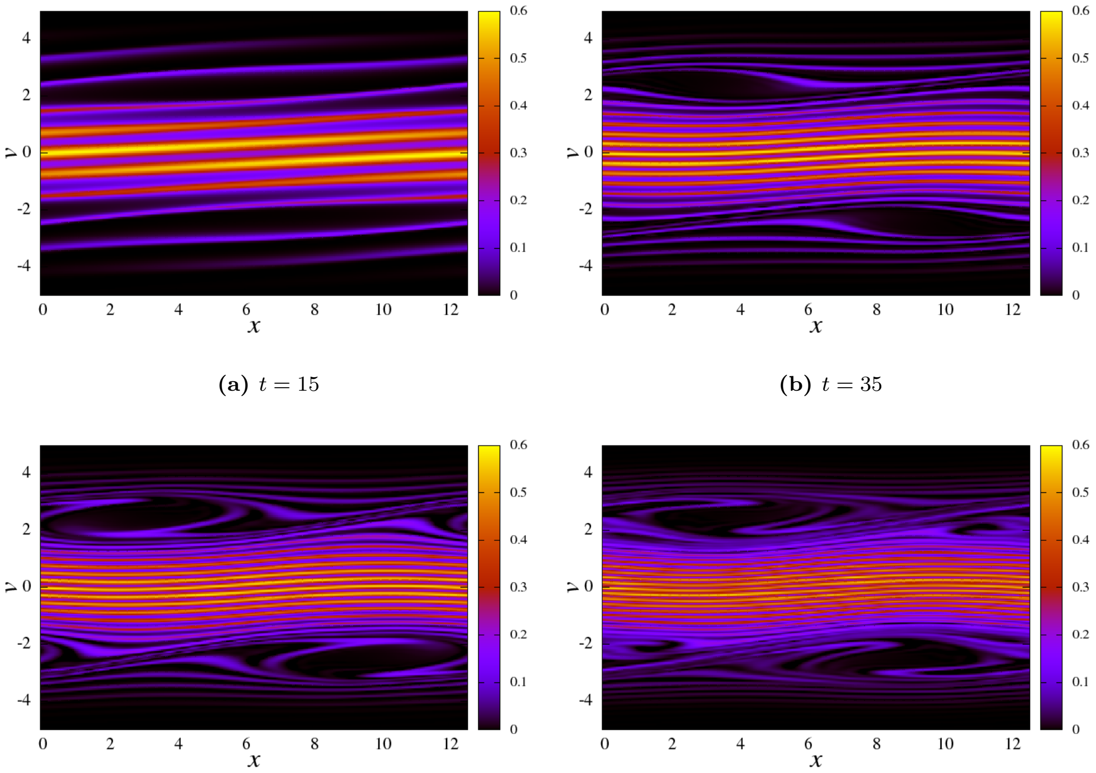
for non-linear Landau damping in Fig. 5.5 (right). Note that upto $t \approx 10$ the error in total energy decreases to 10^{-12} and after that due to the process of filamentation we will have a slowly increasing error until $t \approx 40$ when strong oscillations occur in v-direction. Therefore, the error increases to 10^{-10} which to the knowledge of us was never seen before in literatures. We present the solution of this test case for different times in Fig. 5.7, where one may note the details produced by our scheme. In addition in Fig. 5.8, we present a profile of the solution in v

$$\int_{\Omega_x} f_h(t, x, v) dx$$

to show the ability of scheme to capture the strong oscillation in v-direction.

5.3.1 Effect of polynomial degree (k)

We check the method's response to different polynomials degrees in this section. First we fix the mesh size (100×160) and change the polynomial degree ($k = 1, 2, 3$) then we compare results in Fig. 5.9(top). In order to make a fair comparison between polynomials with different orders we fix the degree of freedoms in the mesh and find the corresponding mesh size, (N_x, N_v). One should mention that the number of degree of freedoms for x and y directions (DG approximation) are $N_x.(k+1)$ and $N_v.(k+1)$ respectively. Hence, for instance we set the degrees of freedoms to 400×640 and we get (N_x, N_v) to be (200×320) for 2nd order, (134×214) 3rd order and (100×160) 4th order. In Fig. 5.9(left-bottom) we show that increasing the degree of polynomials while keeping the degree of freedoms constant will yield a better conservation of total energy and L^2-norm error.



(a) $t = 15$

(b) $t = 35$

(c) $t = 45$

(d) $t = 65$

Figure 5.7: solution of the V-P system for non-linear Landau damping at different times using mesh 100×160 and $k = 3$.

5.3.2 Effect of mesh refinement

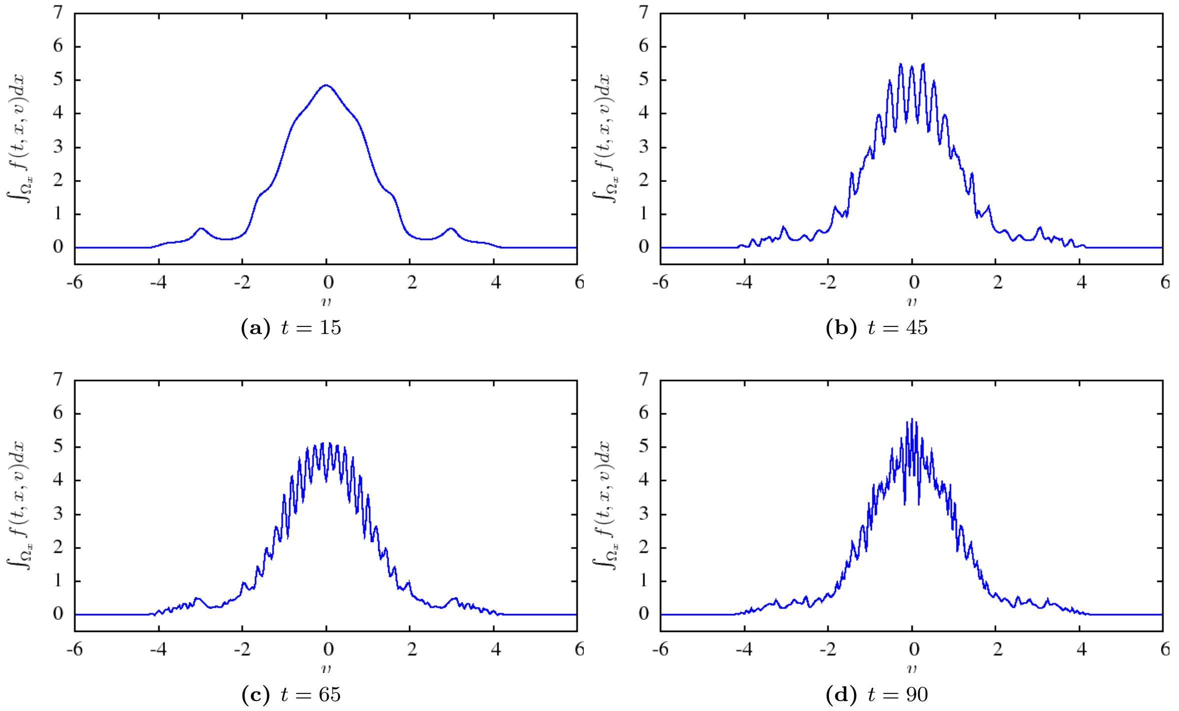
In this part, we show the effect of mesh refinement on the observable quantities of the system for the non-linear Landau damping test case. In order to check the conservation properties of the scheme like total energy, L^1 and L^2-norms we evaluate those values at $t = 0$ using the initial data, $f_0(x, v)$, instead of discrete solution $f_h(t = 0, x, v)$ in double precision and compute the following identities

$$\begin{aligned} &\left|\mathcal{E}_{tot}(t) - \mathcal{E}^0_{tot}\right| && \text{error in total energy} \\ &\left|\|f_h(t, x, v)\|_{L^p} - \|f_0(x, v)\|_{L^p}\right| && \text{error in } L^p \text{ norm} \end{aligned} \tag{5.3.5}$$

where $\mathcal{E}^0_{tot}$ is the exact initial energy

$$\mathcal{E}^0_{tot} \quad := \quad \int_{\Omega_x}\int_{\Omega_v} \frac{|v|^2}{2} f_0(x, v) dx\, dv + \frac{1}{2}\int_{\Omega_x} |E_0(x)|^2 dx \tag{5.3.6}$$

We present the initial decay of the electric field L^2-norm and increments parameters for three different mesh using an exponential fit of the form $c \exp(-\gamma t)$ in Table 5.2. In addition, we present the error illustrated in (5.3.5), in Fig. 5.10 for L^2 and total energy. One may note that by refining the mesh we have less error in those measurements. The estimated $\gamma_{\texttt{decay}}$ is same as the reported one by (-0.292) Rossmanith and Seal in [21] and close to Cheng and Knorr (-0.281) in [9].



(a) $t = 15$

(b) $t = 45$

(c) $t = 65$

(d) $t = 90$

Figure 5.8: x-integrated solution of the non-linear Landau damping for different times where it shows the strong oscillations in v-axis.

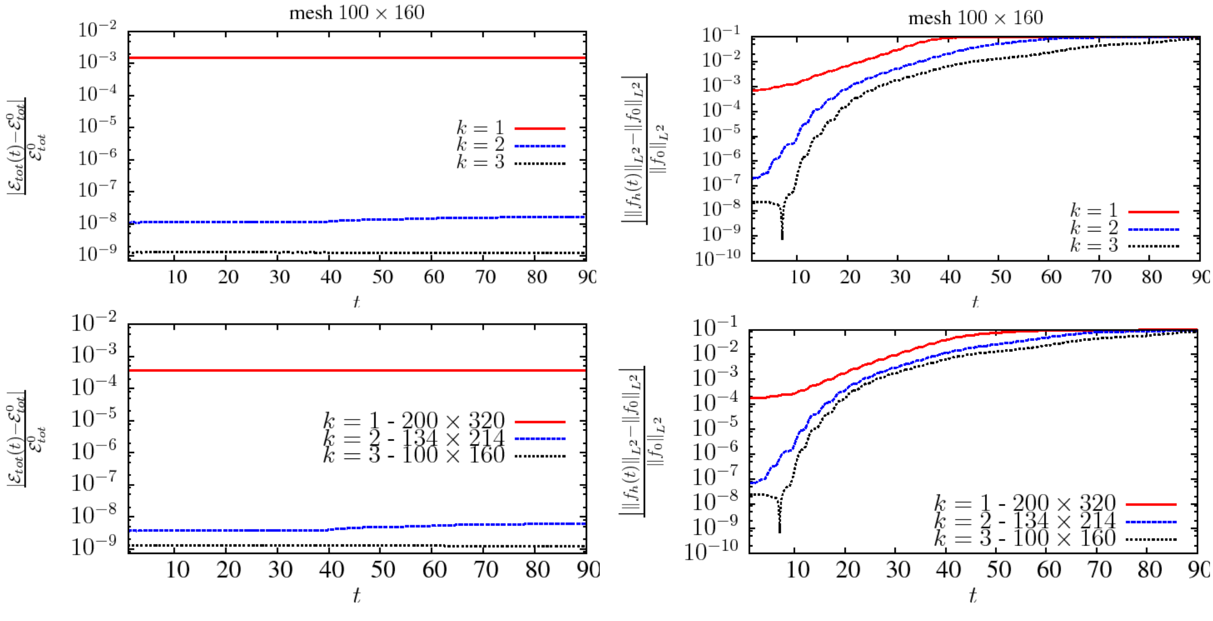


Figure 5.9: evolution of the relative error in L^2-norm and total energy for polynomials of different degree (non-linear Landau damping). (top) comparison with fixed mesh size and different k, (bottom) comparison with fixed degrees of freedom and different k.

5.3.3 Effect of $E(x)$-boundary term approximation

As we introduced in §4.2.4, in the case that electric field has roots in an element I_i, we consider two different approach to evaluate the $E(x)$-boundary term: projecting to $\mathcal{P}_0$ and transforming the integrals.

mesh	$\gamma_{\texttt{decay}}$	$c_{\texttt{decay}}$	$\gamma_{\texttt{increment}}$	$c_{\texttt{increment}}$
50×80	-0.292286	2.279682	0.085114	0.015669
100×160	-0.292285	2.279673	0.086126	0.015228
150×240	-0.292285	2.279673	0.086116	0.015232

Table 5.2: estimated coefficients of initial decay and increment of the electric field L^2-norm where the fitted function is $c\exp(-\gamma t)$ (non-linear Landau damping).

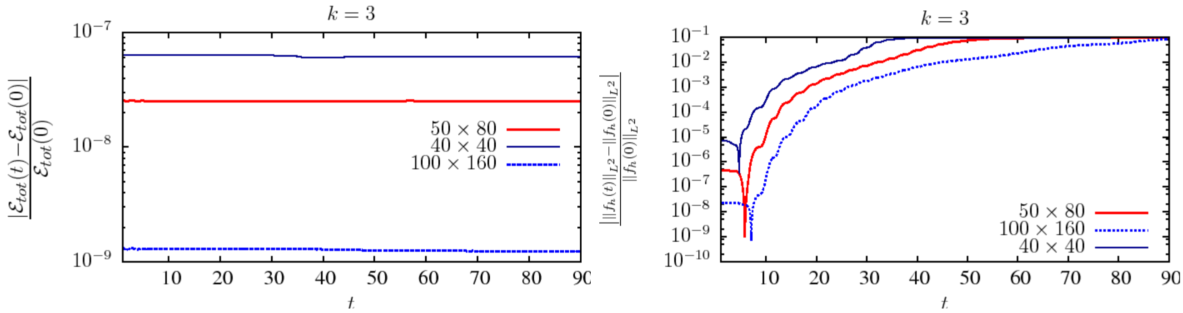


Figure 5.10: evolution of the relative error in L^2, and total energy for different mesh size in a semi-log scale (non-linear Landau damping).

In Fig. 5.11, one may see that using weighted average flux preserves energy better for many order of magnitudes.

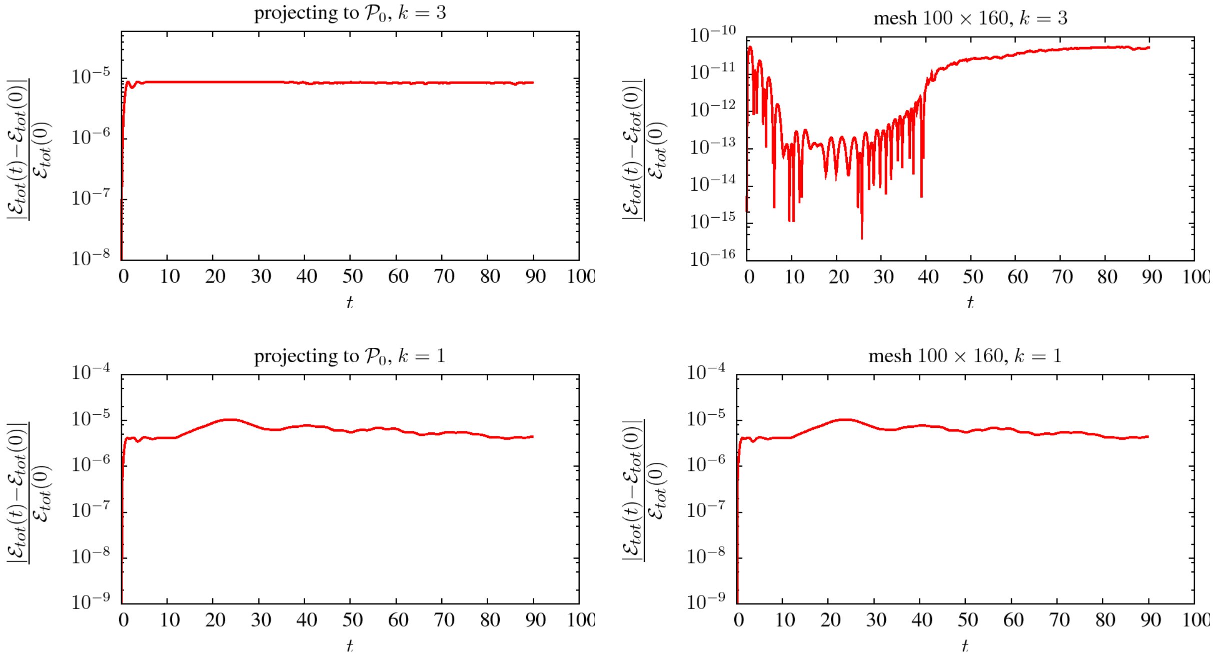


Figure 5.11: The evolution of total energy error in a semi-log scale for non-linear Landau damping using a mesh 100×160 and $k = 1, 3$ with projection to $\mathcal{P}_0$ (left) and weighted average flux (right) for evaluating electric field terms.

5.4 Landau damping (weak case)

The weak Landau damping is also important to the numerical schemes of V-P system because of its similarities to the linear Landau damping problem. Here the initial data is same as nonlinear Landau damping

$$f(t=0,x,v) = \frac{1}{\sqrt{2\pi}}\,(1+\alpha\cos(K\,x))\,e^{-\frac{v^2}{2}} \tag{5.4.1}$$

except that α is chosen very small (small perturbation). Here we choose $\alpha = 0.01$ and same K, Ω_x and Ω_v as before. The mesh is chosen as 60×60 and degree of polynomials $k = 4$. In Fig. 5.12, we show the evolution of total energy error, electric field norm, L^∞ and L^1. Moreover, we fit $c\,\exp(-\gamma t)$ to the “local maximums” of L^2 norm of the electric field where $\gamma = -0.153272$ (compare to -0.1533 in [21]).

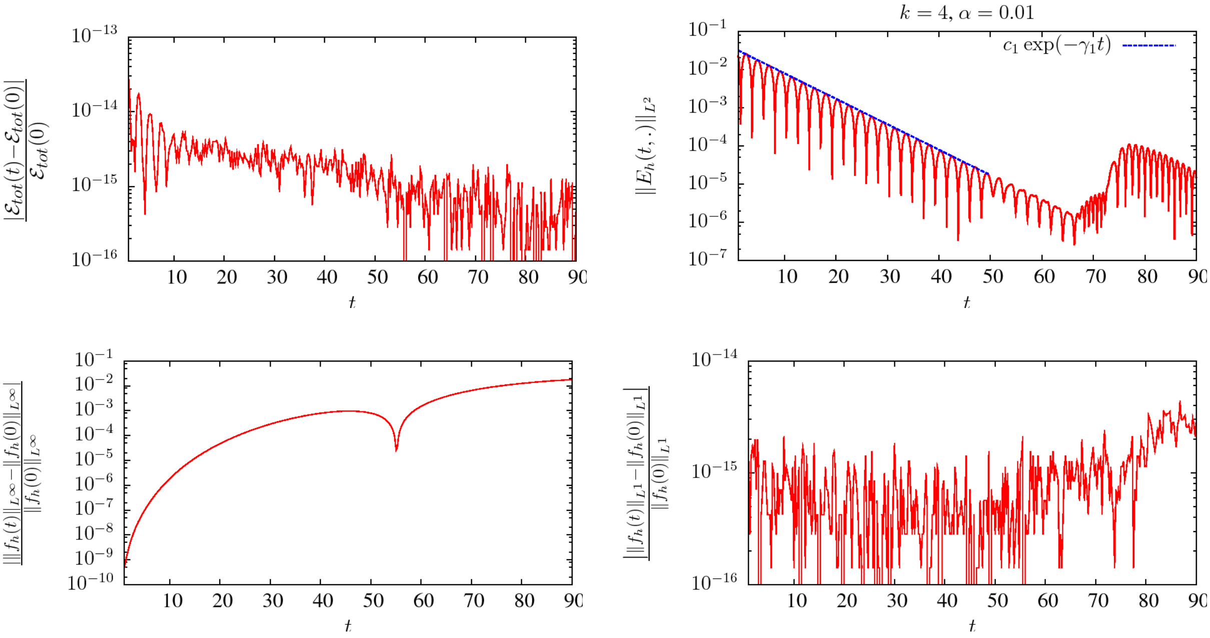


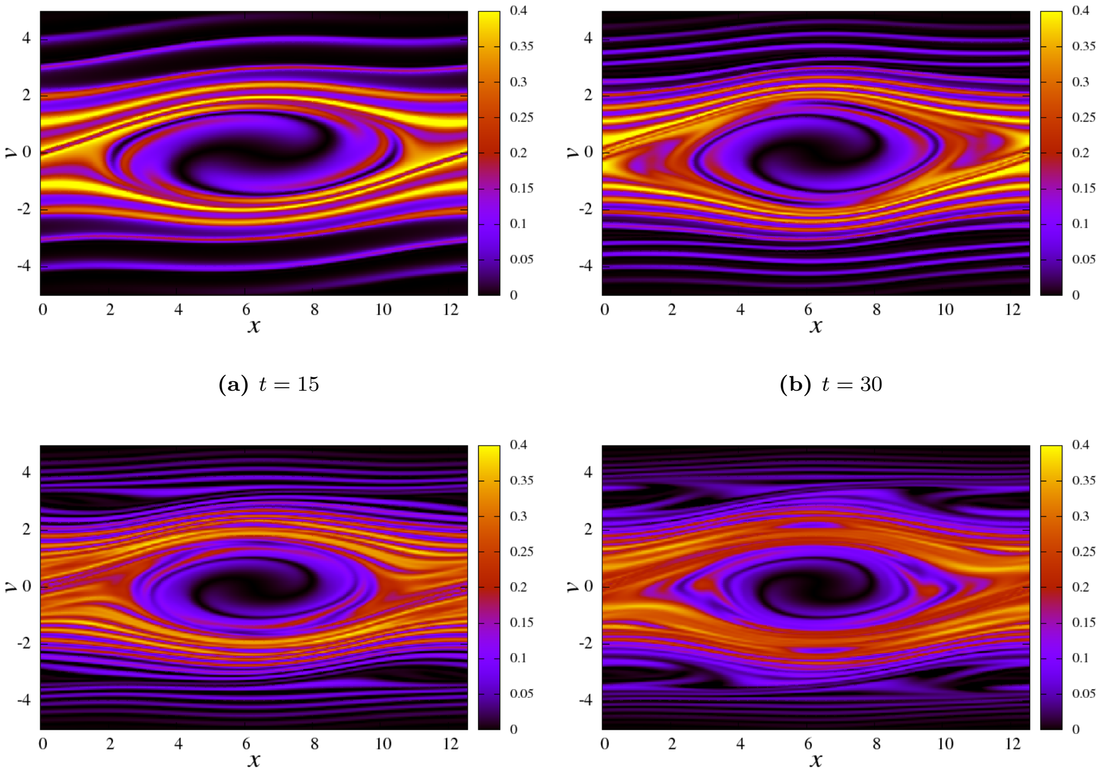
Figure 5.12: evolution of total energy error, electric field norm, L^∞ and L^1 norm for weak nonlinear Landau damping.

5.5 Two stream instability

The two stream instability is a standard benchmark for checking the reliability of the schemes face to strong oscillations. The initial data consists of the two instable flow moving in the opposite direction of each other

$$f(t=0,x,v) = \frac{v^2}{\sqrt{8\pi}}\,\{2-\cos(K(x-2\pi))\}\,e^{-\frac{v^2}{2}} \tag{5.5.1}$$

where $K = 0.5$, $\Omega_x = [0, 4\pi]$ and as discussed in §5.3, $\Omega_v = [-10, 10]$ to insure that $f(t,x,v)|_{v=\partial\Omega_v} \approx 0$. We present the solution, $f_h(t,x,v)$, at different times in Fig. 5.13 for a mesh 150×150. Furthermore, we plot L^p-norms and total energy error upto $t = 90$ in Fig. 5.14. One may note that we again have a good conservation of energy for this test case.



(a) $t = 15$

(b) $t = 30$

(c) $t = 45$

(d) $t = 60$

Figure 5.13: solution of the V-P system for two stream instability at different times using mesh 100×160 and $k = 3$.

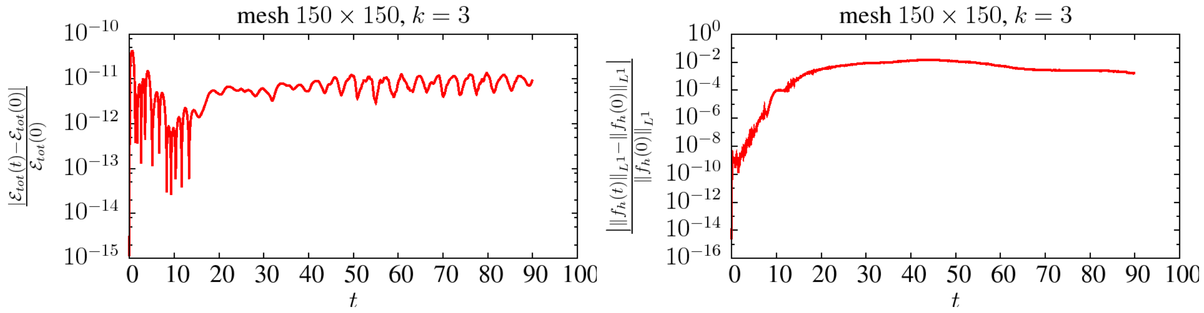


Figure 5.14: evolution of the relative error in L^1 and total energy in a semi-log scale (two stream instability).

5.5.1 Effect of Poisson solvers

Here we check the effect of the different Poisson solver for V-P system on L^2 and total energy. We used the LDG method with and without energy preserving flux described in §3.2.2 and mixed-FEM in §3.2.3. The result for L^2 and total energy preserving is shown in Fig. 5.15. Note that Poisson solver with energy preserving flux (LDG3) has slightly better conservation for total energy compare to others.

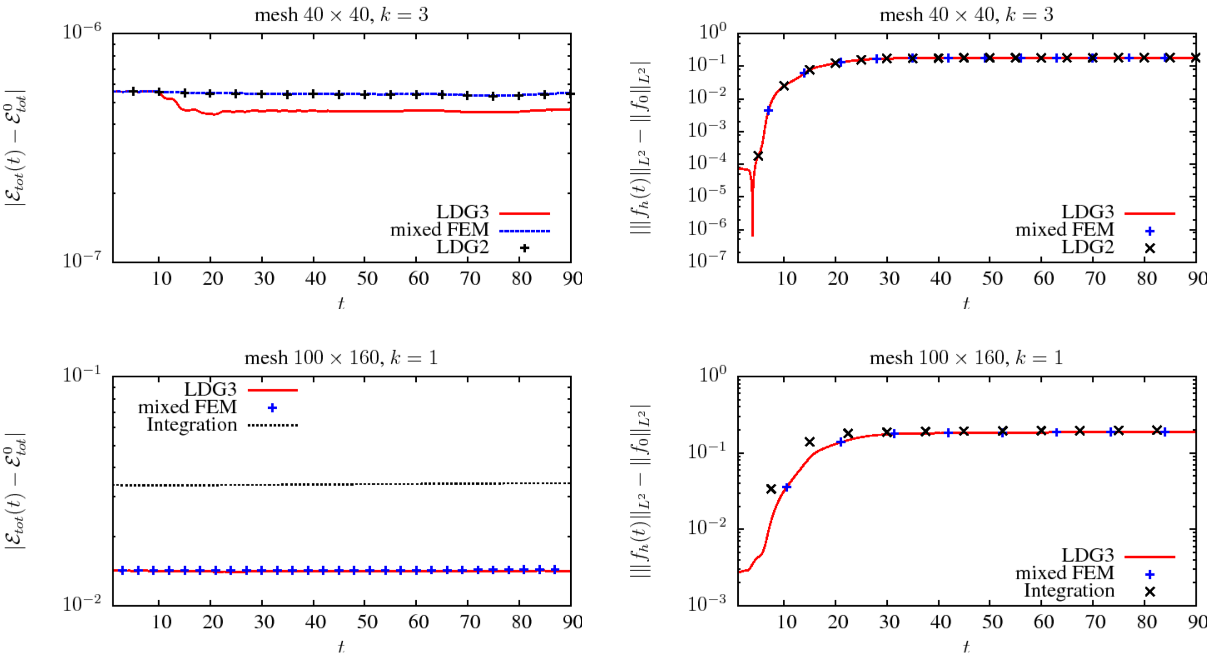


Figure 5.15: evolution of the relative error in L^2-norm and total energy in a semi-log scale (two stream instability).

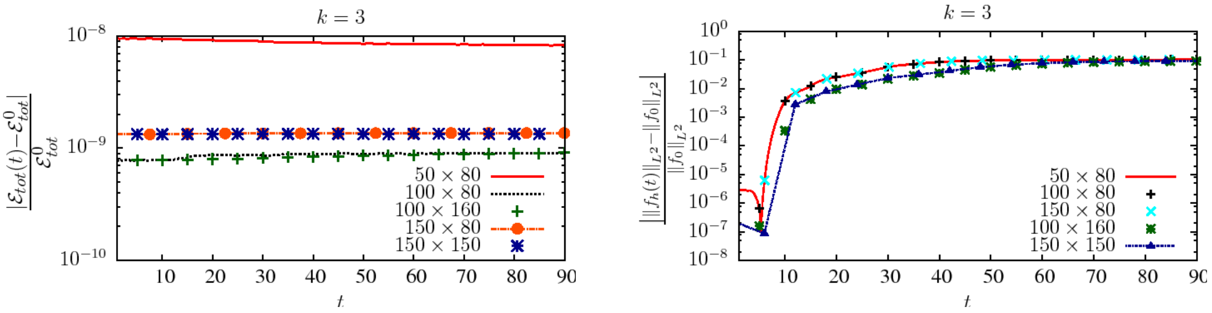


Figure 5.16: evolution of the relative error in L^2-norm and total energy in a semi-log scale (two stream instability).

5.5.2 Effect of mesh refinement

As we mentioned the mesh refinement for non-linear damping case, we also investigate the effect of mesh refinement on the observable quantities for the two stream instability test case. Here, we check the effect of refining the Ω_x and Ω_v separately. Moreover, we will use $\|f_0\|_{L^p}$ as initial L^p-norm of solution instead of $\|f_h(0)\|_{L^p}$ where $\|f_0\|_{L^p}$ is L^p-norm of initial data and similarly for total energy, we choose $\mathcal{E}^0_{tot}$ introduced in (5.0.1). Hence we will compare the errors introduced in (5.3.5) for different meshes.

We found out that the error in L^1, L^2 and L^∞ depends only on refinement of Ω_v and not Ω_x. Moreover we observe a different effect for error in energy, while it depends more on refinement of Ω_x (since it will refine the mesh of the Poisson solver). One may see these effects in Fig. 5.16 for $\mathcal{E}_{tot}$ and L^2.

5.6 Two stream instability II

In this section, we use a different initial data for the two stream instability test case

$$f(t=0,x,v) = \frac{1}{2v_{th}\sqrt{2\pi}}\left\{\exp\left(-\frac{(v-w)^2}{2v_{th}^2}\right)+\exp\left(-\frac{(v+w)^2}{2v_{th}^2}\right)\right\}(1+0.05\cos(K\,x)) \qquad (5.6.1)$$

where $v_{th}=0.3$, $w=0.99$ and $K=\frac{2}{13}$. For this experiment, we set $\Omega_x=[0,13\pi]$ and $\Omega_v=[-8,8]$. We show the relative error in energy of the V-P system using $k=3$ for different mesh in Fig. 5.17(left) and for different k using a fixed mesh in Fig. 5.17(right). Furthermore, in Fig. 5.18, we present the solution of the system at $t=70$ for different $k=$.

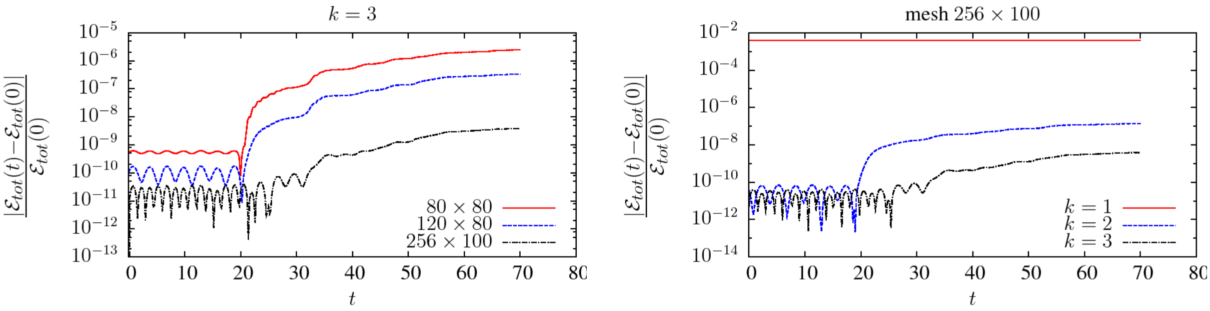
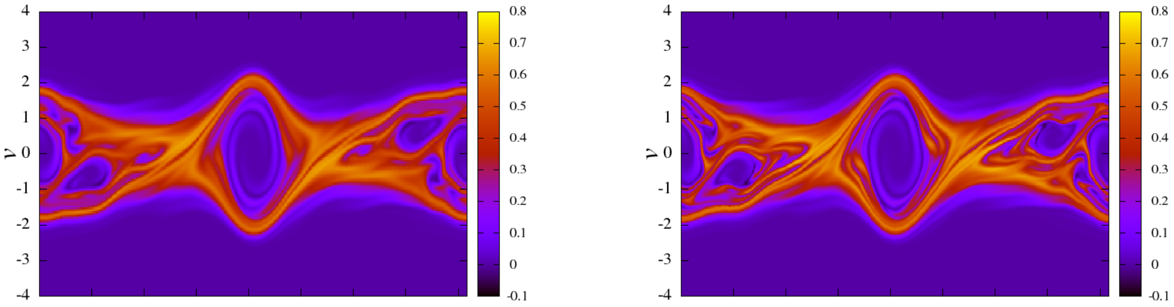


Figure 5.17: evolution of the relative error in total energy in a semi-log scale (two stream instability II).



(a) $k=2$

(b) $k=3$

Figure 5.18: solution of the V-P system for two stream instability II at $t=70$ using mesh 256×100.

5.7 Non-smooth solution

Let's consider the boundary value problem

$$\begin{cases} f_t + vf_x + E(t,x)f_v = 0 \quad (t,x,v)\in\mathbb{R}^+\times[0,1]\times\mathbb{R} \\ f(t,0,v)=g(t,v), \quad v>0; \quad f(t,1,v)=0, \quad v<0 \\ \Phi_{xx}=\partial_x E(t,x)=\rho(t,x) \end{cases} \qquad (5.7.1)$$

with

$$\begin{cases} \{\Phi(t,0)=0, \Phi(t,1)=\lambda_0\} \Leftrightarrow \displaystyle\int_0^1 E(t,x)\,dx = \lambda_0 \\ \lambda_0 > \displaystyle\int_0^1 \int_{\mathbb{R}} (1-x) f_0 \, dv\, dx \end{cases} \tag{5.7.2}$$

where

$$g(t,v) = \frac{1}{\sqrt{2\pi}} v^2 \exp(-v^2/2)$$

using initial data

$$\begin{aligned} f_0(x,v) &= n_0(x) \frac{1}{\sqrt{2\pi}} v^2 \exp(-v^2/2) \\ n_0(x) &= \begin{cases} (1+\gamma x)(1-4x^2)^4 & x \in [0, 0.5] \\ 0 & \text{else} \end{cases} \end{aligned}$$

following [24] and [15], we show that for λ_0 satisfying condition (5.7.2), we will have change in regularity of solution. In practice we use $\lambda_0 = 2.10947$ that satisfies condition and $\lambda_0 = 0$ for normal case. In Fig. 5.19(left), one may note that for $\lambda_0 = 2.10947$, the $\|\nabla f_h\|_{L^2}$ is diverging by refining the mesh while converge for $\lambda_0 = 0$. In Fig. 5.19(right), it is shown that the $\|f_h\|_{L^\infty}$ will converge by refining the mesh. We also presented the solution using $\lambda_0 = 2.10947$ in Fig. 5.20.

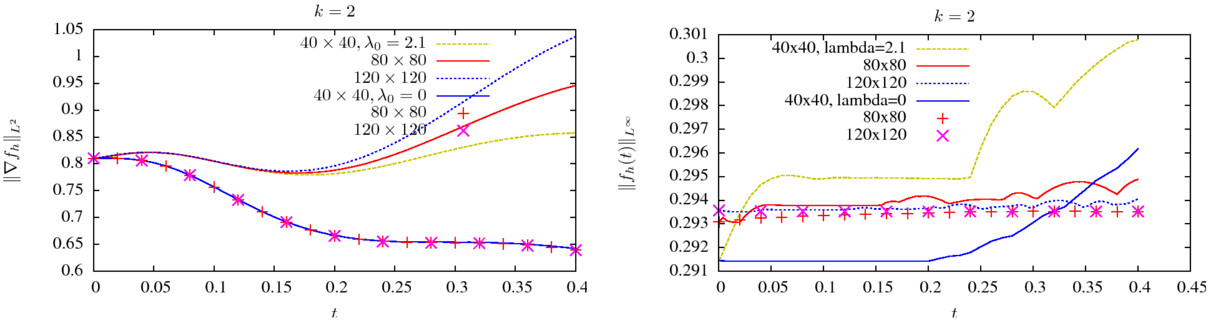


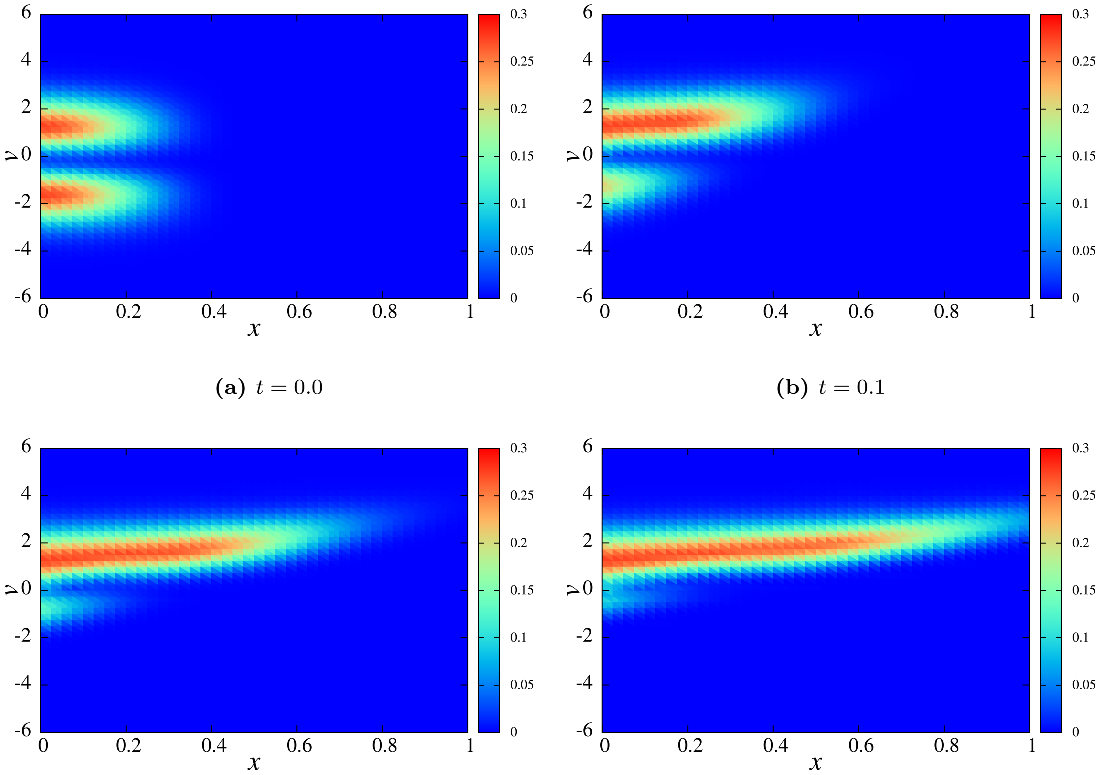
Figure 5.19: Evolution of $\|\nabla f_h\|_{L^2}$ (left) and $\|f_h\|_{L^\infty}$ (right) for different mesh size and different λ_0.

5.7.1 Effect of polynomial degree (k)

In this section, we show the behavior of the system for different degrees of polynomial as well as refining the mesh while keeping $\lambda_0 = 2.10947$. In Fig. 5.21(left), for different $k = 1, 2, 3, 4$ we showed that the gradient of the solution is diverging by increasing k and refining the mesh, while on the (right) we have convergence in $\|f_h\|_{L^\infty}$.

5.7.2 Larger λ_0

We choose a higher value for λ_0 to check its effect on the solution and $\|\nabla f_h\|_{L^2}$. We take $\lambda_0 = 10$ and plot the density solution in Fig. 5.22 and a profile of the solution at $x = 0$ in Fig. 5.23. Finally we present $\|\nabla f_h\|_{L^2}$ and $\|f_h\|_{L^2}$ in Fig. 5.24.



(a) $t = 0.0$

(b) $t = 0.1$

(c) $t = 0.2$

(d) $t = 0.3$

Figure 5.20: solution of the V-P system for non-smooth solution test case at different times using mesh 60×60 and $k = 2$.

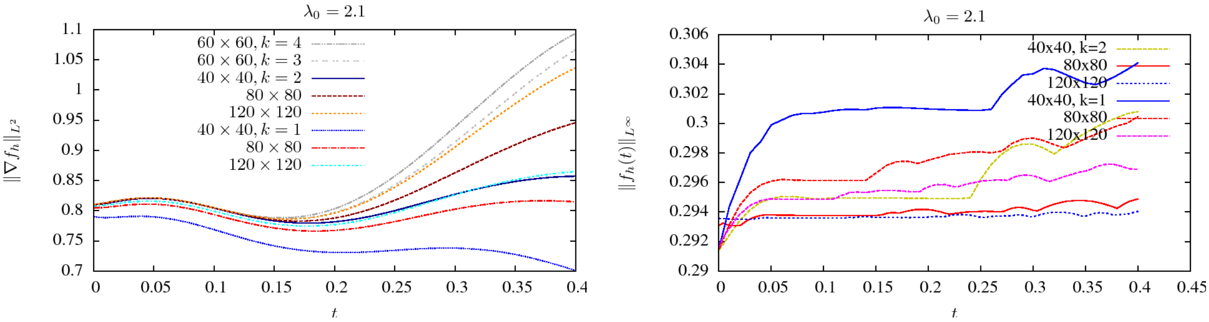
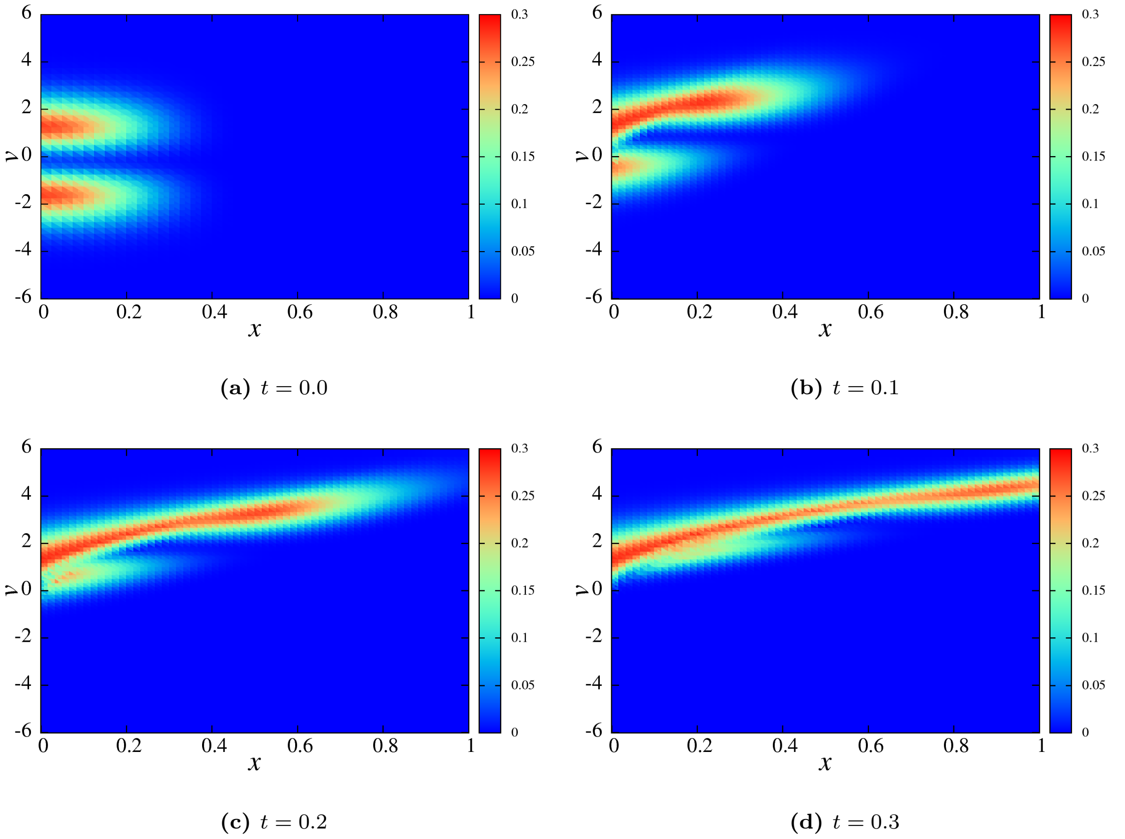


Figure 5.21: Evolution of $\|\nabla f_h\|_{L^2}$ (left) and $\|f_h\|_{L^\infty}$ (right) for different mesh size and different k.



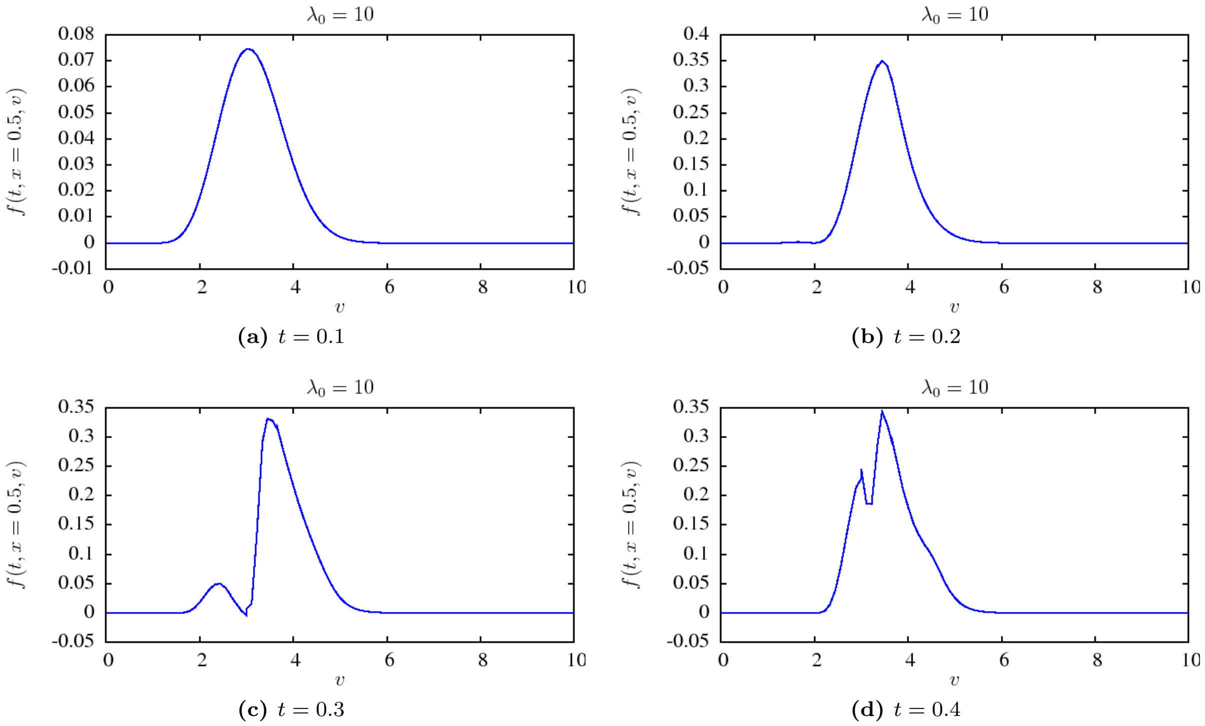
(a) $t = 0.0$

(b) $t = 0.1$

(c) $t = 0.2$

(d) $t = 0.3$

Figure 5.22: solution of the V-P system for non-smooth solution test case at different times using mesh 60×60 and $k = 2$.



(a) $t = 0.1$

(b) $t = 0.2$

(c) $t = 0.3$

(d) $t = 0.4$

Figure 5.23: The profile of the solution at $x = 0.5$ and different times for $\lambda_0 = 10$.

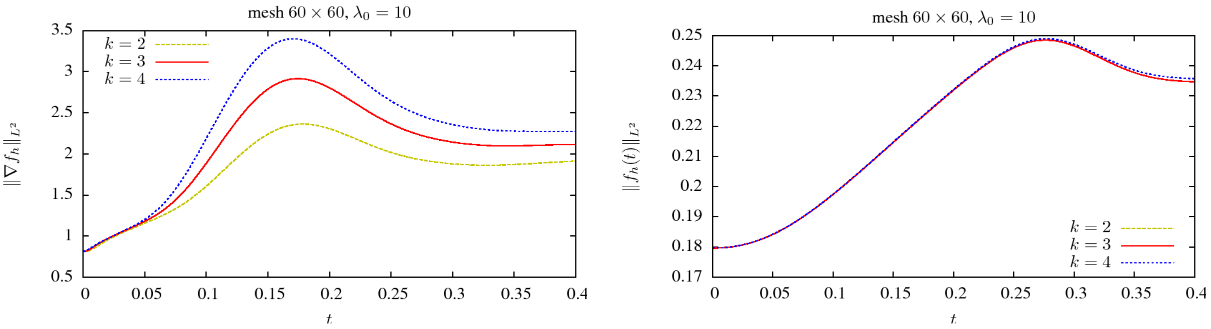


Figure 5.24: Evolution of $\|\nabla f_h\|_{L^2}$ (left) and $\|f_h\|_{L^2}$ (right) for different mesh size and different k.

Appendix A

Weighted average

The idea is to approximate the exact calculation of the integral

$$\int_I \widehat{E}f f\,dx = \int_{I^+} E(x)f^+ f\,dx + \int_{I^-} E(x)f^- f\,dx \tag{A.0.1}$$

to the following approximation

$$\begin{aligned} \int_{I^+} E(x)f^+ f\,dx &\approx w^+ \int_I E(x)f^+ f\,dx && \text{(A.0.2)} \\ \int_{I^-} E(x)f^- f\,dx &\approx w^- \int_I E(x)f^- f\,dx; \quad w^+, w^- \in \mathbb{R} && \text{(A.0.3)} \end{aligned}$$

where f^+, f^- are the solutions that will be used when $x \in I^+$ and $x \in I^-$ respectively. Note that the approximate integrals will be evaluated on I instead of I^+ and I^- and hence are very easy to compute as illustrated in previous section. So the difficulty is to find values w^+ and w^- that minimize the error of approximation. We are going to evaluate the following error

$$\begin{aligned} \left|\int_I \widehat{E}f\, f\,dx - \left(w^+ \int_I E(x)f^+ f\,dx + w^- \int_I E(x)f^+ f\,dx\right)\right| &= \\ \left|\int_{I^+} E(x)f^+ f\,dx - w^+ \int_I E(x)f^+ f\,dx + \int_{I^-} E(x)f^- f\,dx - w^- \int_I E(x)f^+ f\,dx\right| &\leq |\mathcal{A}| + |\mathcal{B}| \end{aligned}$$

where

$$\begin{aligned} \mathcal{A} &:= \int_{I^+} E(x)f^+ f\,dx - w^+ \int_I E(x)f^+ f\,dx, \\ \mathcal{B} &:= \int_{I^-} E(x)f^- f\,dx - w^- \int_I E(x)f^+ f\,dx && \text{(A.0.4)} \end{aligned}$$

then

$$\begin{aligned} \mathcal{A} &= w^+ \int_I E(x)f^+ f\,dx - \int_{I^+} E(x)f^+ f\,dx \\ &= w^+ \int_{I^+} E(x)f^+ f\,dx + w^+ \int_{I^-} E(x)f^+ f\,dx - \int_{I^+} E(x)f^+ f\,dx \\ &= \underbrace{(w^+ - 1)\int_{I^+} E(x)f^+ f\,dx}_{\mathcal{A}_1} + \underbrace{w^+ \int_{I^-} E(x)f^+ f\,dx}_{\mathcal{A}_2} \end{aligned}$$

taking square of both side

$$\mathcal{A}^2 = \mathcal{A}_1^2 + \mathcal{A}_2^2 + 2w^+(w^+ - 1)\left(\int_{I^+} E(x)f^+ f\,dx\right)\left(\int_{I^-} E(x)f^+ f\,dx\right)$$

subtracting $\mathcal{A}_1^2$ and $\mathcal{A}_2^2$ and taking square again

$$\begin{aligned}
\left(\mathcal{A}^2 - \mathcal{A}_1^2 - \mathcal{A}_2^2\right)^2 &= (2w^+(w^+-1))^2 \left(\int_{I^+} E(x) f^+ f\, dx\right)^2 \left(\int_{I^-} E(x) f^+ f\, dx\right)^2 \\
&\leq (2w^+(w^+-1))^2 \left(\int_{I^+} E^2(x)\, dx\right) \left(\int_{I^+} (f^+ f)^2\, dx\right) \cdot \\
&\qquad \left(\int_{I^-} E^2(x)\, dx\right) \left(\int_{I^-} (f^+ f)^2\, dx\right) \\
&\leq (2w^+(w^+-1))^2 \cdot \left[\max_{I^+} E^2\right] \left|I^+\right| \cdot \left[\max_{I^-} E^2\right] \left|I^-\right| \cdot \|f^+ f\|_{L^2(I)}^4
\end{aligned}$$

taking square root of both side

$$\begin{aligned}
\mathcal{A}^2 &\leq \mathcal{A}_1^2 + \mathcal{A}_2^2 \\
&\quad + \left|2w^+(w^+-1)\right| . \left[\max_{I^+} E^2\right]^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}} . \left|I^+\right|^{\frac{1}{2}} \left|I^-\right|^{\frac{1}{2}} . \|f^+ f\|_{L^2(I)}^2
\end{aligned} \tag{A.0.5}$$

moreover

$$\begin{aligned}
\mathcal{A}_1^2 &= (1-w^+)^2 \int_{I^+} E(x) f^+ f\, dx \\
&\leq (1-w^+)^2 \left(\int_{I^+} E^2(x)\, dx\right) \left(\int_{I^+} (f^+ f)^2\, dx\right) \\
&\leq (1-w^+)^2 \left[\max_{I^+} E^2\right] \left|I^+\right| \|f^+ f\|_{L^2(I)}^2
\end{aligned}$$

similarly

$$\mathcal{A}_2^2 \leq (w^+)^2 \left[\max_{I^-} E^2\right] \left|I^-\right| \|f^+ f\|_{L^2(I)}^2$$

substituting to (A.0.5) we will have

$$\begin{aligned}
\mathcal{A}^2 &\leq \Bigg((1-w^+)^2 \left[\max_{I^+} E^2\right] \left|I^+\right| + (w^+)^2 \left[\max_{I^-} E^2\right] \left|I^-\right| \\
&+ \left|2w^+(w^+-1)\right| . \left[\max_{I^+} E^2\right]^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}} . \left|I^+\right|^{\frac{1}{2}} \left|I^-\right|^{\frac{1}{2}}\Bigg) \|f^+ f\|_{L^2(I)}^2 \\
&= \left((1-w^+) \left|I^+\right|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}} - w^+ \left|I^-\right|^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}}\right)^2 \|f^+ f\|_{L^2(I)}^2 \\
&= \left(\left|I^+\right|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}} - w^+ \left\{\left|I^+\right|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}} + \left|I^-\right|^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}}\right\}\right)^2 \|f^+ f\|_{L^2(I)}^2
\end{aligned}$$

now we can minimize the right-hand side (and hence the error) by choosing w^+ as

$$w^+ = \frac{|I^+|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}}}{|I^+|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}} + |I^-|^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}}} \tag{A.0.6}$$

similar calculation for $\mathcal{B}$ in (A.0.4) will yield

$$w^- = \frac{|I^-|^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}}}{|I^+|^{\frac{1}{2}} \left[\max_{I^+} E^2\right]^{\frac{1}{2}} + |I^-|^{\frac{1}{2}} \left[\max_{I^-} E^2\right]^{\frac{1}{2}}} \tag{A.0.7}$$

Note that the values obtained for w^+ and w^- are consistent in the sense that

$$w^+ + w^- = 1$$

moreover

$$\begin{aligned} &(w^+ \to 1)\&(w^- \to 0) \qquad \text{when} \quad I^+ \to I \\ &(w^+ \to 0)\&(w^- \to 1) \qquad \text{when} \quad I^- \to I \end{aligned}$$

although we found a nice approximation to our integral, we are still unable to find $|I^+|$ and $|I^-|$ (but it is still better than calculating the roots and partitioning), hence we make a further simplification and we assume

$$\frac{|I^+|}{|I|} = \frac{|I^-|}{|I|} = 1/2$$

and the final expression for $w^\pm$ reads

$$\begin{aligned} w^+ &= \frac{|\max_{I_i} E_h|}{|\max_{I_i} E_h| + |\min_{I_i} E_h|} \\ w^- &= \frac{|\min_{I_i} E_h|}{|\max_{I_i} E_h| + |\min_{I_i} E_h|} \end{aligned} \tag{A.0.8}$$

Bibliography

[1] R. A. Adams, *Sobolev Spaces*, Pure and Applied Mathematics, vol. 65, Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1975.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39(5) (2001/02), pp. 1749–1779 (electronic).

[3] B. Ayuso, J. A. Carrillo, and C.-W. Shu, *Discontinuous Galerkin methods for the one-dimensional Vlasov-Poisson system*, to appear in Kinetic and Related Models. (See also Brown University, technical report 2009-41 (2009).)

[4] B. Ayuso de Dios, J. A. Carrillo, and C.-W. Shu, *Discontinuous Galerkin methods for the multi-dimensional Vlasov-Poisson problem*, Newton Institute Preprint Series 2010 NI10038-KIT. http://www.newton.ac.uk/preprints/NI10038.pdf

[5] C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation*, McGraw-Hill, New York, 1985.

[6] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag, New York, 1991.

[7] J. A. Carrillo and F. Vecil, *Nonoscillatory interpolation methods applied to Vlasov-based models*, SIAM J. Sci. Comput., 29(3) (2007), pp. 1179–1206.

[8] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau, *An a priori error analysis of the local discontinuous Galerkin method for elliptic problems*, SIAM J. Numer. Anal., 38(5) (2000), pp. 1676–1706 (electronic).

[9] C. Z. Cheng and G. Knorr, *The integration of the Vlasov equation in configuration space*, J. Comp. Phys., 22 (1976), pp. 330–348.

[10] G.-H. Cottet and P.-A. Raviart *Particle methods for the one-dimensional Vlasov-Poisson equations*, SIAM J. Numer. Anal., 21(1) (1984), pp. 52–76.

[11] N. Crouseilles, M. Mehrenberger, and F. Vecil, *Discontinuous Galerkin semi-Lagrangian method for Vlasov-Poisson*, preprint hal-00544677.

[12] N. Crouseilles, G. Latu, and E. Sonnendrücker, *A parallel Vlasov solver based on local cubic spline interpolation on patches*, J. Comput. Phys., 228(5) (2009), pp. 1429–1446.

[13] F. Filbet and E. Sonnendrücker, *Comparison of Eulerian Vlasov solvers*, Comput. Phys. Comm., 150(3) (2003), pp. 247–266.

[14] F. Filbet, E. Sonnendrücker, and P. Bertrand, *Conservative numerical schemes for the Vlasov equation*, J. Comput. Phys., 172(1) (2001), pp. 166–187.

[15] Filbet, Francis; Guo, Yan; Shu, Chi-Wang *Analysis of the relativistic Vlasov-Maxwell model in an interval*, Quart. Appl. Math. 63 (2005), no. 4, 691714.

[16] I. Gamba, R. E. Heath, P. Morrison, and C. Michler, *A discontinuous Galerkin method for the Vlasov-Poisson system*, to appear in Jour. Comp. Phys., 2011.

[17] R. T. Glassey, *The Cauchy Problem in Kinetic Theory*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[18] R. E. Heath, *Analysis of the discontinuous galerkin method applied to collisionless plasma physics*, PhD thesis, University of Texas at Austin, Austin TX (2007).

[19] A. J. Klimas and W. M. Farrell, *A splitting algorithm for Vlasov simulation with filamentation filtration*, J. Comput. Phys., 110(1) (1994), pp. 150–163.

[20] P. Lasaint and P.-A. Raviart, *On a finite element method for solving the neutron transport equation*, in Mathematical Aspects of Finite Elements in Partial Differential Equations (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1974), publication no. 33, Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, 1974, pp. 89–123.

[21] J. A. Rossmanitha, and D. C. Sea, *A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations*, to appear in Journal of Computational Physics (2011), arXiv:1012.2494v2

[22] S. I. Zaki and L. R. T. Gardner, *A finite element code for the simulation of one-dimensional vlasov plasmas. i. theory*, J. Comput. Phys., 79(1) (1988), pp. 184–199.

[23] S. I. Zaki, B. T.J.M., and L. R. T. Gardner, *A finite element code for the simulation of one-dimensional vlasov plasmas. ii. applications*, J. Comput. Phys., 79(1) (1988), pp. 200–208.

[24] T. Zhou, Y. Guo, and C.-W. Shu, *Numerical study on landau damping*, Physica D, 157 (2001), pp. 322–333.