Conservative discontinuous Galerkin methods for the kinetic Fokker-Planck equation

Wietse Vaes

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2023

Reading Committee:

Jingwei Hu, Chair

Thomas Trogdon

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Conservative discontinuous Galerkin methods for the kinetic Fokker-Planck equation

Wietse Vaes

Chair of the Supervisory Committee:
Associate Professor Jingwei Hu
Applied Mathematics

We consider the kinetic Fokker-Planck equation, a simplified model of the Vlasov-Landau equation, that describes collisions in plasma. This diffusion-type equation exhibits numerous noteworthy properties. One such property is the conservation of mass, momentum and energy. The numerical methods in this thesis, namely the local and recovery discontinuous Galerkin methods for diffusion-type equations, maintain this over large and truncated domains. Employing these methods results in stability results that fall in line with theoretical expectations. However, the findings also include unexpected convergence and asymptotic behaviors, which prompt further investigation.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS

First, I want to express my sincerest gratitude to Professor Jingwei Hu. Her invaluable guidance and expertise have been instrumental throughout the entire thesis journey. The insightful feedback and commitment to excellence have significantly impacted the quality of my research here and surely in future research endeavors

I want to also extend my heartfelt thanks to Jack Coughlin and professor Tom Trogdon, for their efforts, constructive discussions, and shared commitment to this work. Their contributions have played a key role in shaping this research.

The financial support from the Belgian American Educational Foundation (BAEF) and the Department of Applied Mathematics (AMATH) at the University of Washington has been crucial in facilitating the execution of this research. I am deeply appreciative of their generosity and proud to be associated with them.

Special thanks to Alex Barga for his meticulous nitpicking when reviewing and enhancing the clarity of my writing. Appreciation also goes to my cat, Chicken, whose presence provided moments of rest and (in)sanity during the final phases of this research.

Lastly, I want to express my gratitude to the Numerical Analytic Reading Club (NARC) at the University of Washington for their valuable insights and constructive feedback of the practice defense. Their thoughtful comments and probing questions have been instrumental in refining the presentation of this work.

This thesis is a culmination of the collective support, encouragement, and contributions of these individuals and institutions. I am truly grateful and proud to have the privilege of working with such people.

Chapter 1

INTRODUCTION

Prior to exploring numerical methods, it is crucial to address what we want to numerically approximate. Following a comprehensive discussion of relevant properties of the kinetic Fokker-Planck equation, the analytical solution itself is derived for a specific case. Subsequently, this introduction will elucidate its research objectives and intentions in a more explicit manner.

1.1 The Fokker-Planck equation

1.1.1 The equation

Our discussion starts from the Vlasov-Landau equation for plasmas,

$$\partial_t f_a + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f_a + \frac{\boldsymbol{F}}{m_a} \cdot \nabla_{\boldsymbol{v}} f_a = (\partial_t f_a)_c \,, \tag{1.1}$$

where f_a describes the molecular distribution of species a in the phase space of position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$ with mass m_a and $\boldsymbol{F}$ is an external force field. Denote the spatial and velocity domain, respectively, by $\Omega_{\mathbf{x}}$ and $\Omega_{\mathbf{v}}$. Real life situations typically have $\Omega_{\mathbf{x}} \subset \mathbb{R}^3$ and $\Omega_{\mathbf{v}} = \mathbb{R}^3$. All that remains is to describe $(\partial_t f_a)_c$. This quantity portrays the change of the function produced by collisions of charged particles. These collisions happen between all the species, including its own, with a collision frequency. For the purpose of this thesis, only one species is observed. As this is the case, specification of it is no longer needed. For ease of notation, their mass will also be taken as one. This choice in mass is justified through non-dimensionilization. The case for a general constant mass is easily found and has very similar results.

The Fokker-Planck equation is at its core a conservation equation; therefore, an advection and diffusion term is involved in the collision term. A collision term preferred by physicists is described by Rosenbluth et al. (1957). From a mathematical perspective, the collision

term $(\partial_t f)_c$ is described by the bilinear Landau collision operator, first defined by Landau (1936):

$$Q(g,f) = \nabla_{\boldsymbol{v}} \cdot \int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)[g(\boldsymbol{v}_*)\nabla_{\boldsymbol{v}} f(\boldsymbol{v}) - f(\boldsymbol{v})\nabla_{\boldsymbol{v}_*} g(\boldsymbol{v}_*)]\,\mathrm{d}\boldsymbol{v}_*,$$

where the collision kernel A is a semi-positive-definite matrix corresponding to the Coulomb collision kernel. To describe this kernel, we define the mass density ρ, temperature T and bulk velocity u as

$$\rho = \int_{\Omega_{\boldsymbol{v}}} f\,\mathrm{d}\boldsymbol{v}, \qquad T = \frac{1}{\rho}\int_{\Omega_{\boldsymbol{v}}} |\boldsymbol{v}-\boldsymbol{u}|^2 f\,\mathrm{d}\boldsymbol{v}, \qquad \text{with} \qquad \boldsymbol{u} = \frac{1}{d\rho}\int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v} f\,\mathrm{d}\boldsymbol{v}, \quad (1.2)$$

with d the dimension of the velocity space. Now the collision kernel can be defined as

$$A(\boldsymbol{v}) := c(\rho,T)|\boldsymbol{v}|^{-3}(|\boldsymbol{v}|^2 I - \boldsymbol{v}\otimes\boldsymbol{v}).$$

The charged particle collision term for one species now arises from assuming the distribution function f collides with the Maxwellian M, where the Boltzmann constant k_B is one, for ease of notation, temperature is denoted by T and is assumed to be centered around the bulk velocity $\boldsymbol{u}$,

$$M(v) = \rho\left(\frac{1}{2\pi T}\right)^{\frac{d}{2}} \exp\left(-\frac{|v-u|^2}{2T}\right). \quad (1.3)$$

We thus want to evaluate

$$Q(M,f)(v) = \nabla_{\boldsymbol{v}} \cdot \int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)[M(\boldsymbol{v}_*)\nabla_{\boldsymbol{v}} f(\boldsymbol{v}) - f(\boldsymbol{v})\nabla_{\boldsymbol{v}_*} M(\boldsymbol{v}_*)]\,\mathrm{d}\boldsymbol{v}_*.$$

For this, we look closer at the loss term

$$\int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)\nabla_{\boldsymbol{v}_*} M(\boldsymbol{v}_*)\,\mathrm{d}\boldsymbol{v}_* = -\int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)\frac{\boldsymbol{v}_*-\boldsymbol{u}}{T} M(\boldsymbol{v}_*)\,\mathrm{d}\boldsymbol{v}_*.$$

One can then add and subtract the velocity variable in the fraction and use the definition of $A(\boldsymbol{v}-\boldsymbol{v}_*)$ to see that

$$\int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)\frac{\boldsymbol{v}-\boldsymbol{v}_*}{T} M(\boldsymbol{v}_*)\,\mathrm{d}\boldsymbol{v}_* = 0.$$

The loss term then becomes

$$\int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)\nabla_{\boldsymbol{v}_*} M(\boldsymbol{v}_*)\,\mathrm{d}\boldsymbol{v}_* = -\int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v}-\boldsymbol{v}_*)\frac{\boldsymbol{v}-\boldsymbol{u}}{T} M(\boldsymbol{v}_*)\,\mathrm{d}\boldsymbol{v}_*.$$

Therefore, the Landau operator on the Maxwellian and distribution function turns into

$$Q(M,f)(v) = \nabla_{\boldsymbol{v}} \cdot A_M \left(\nabla_{\boldsymbol{v}} f + \frac{v-u}{T} f \right)$$

where

$$A_M = \int_{\Omega_{\boldsymbol{v}}} A(\boldsymbol{v} - \boldsymbol{v}_*) M(\boldsymbol{v}_*) \, \mathrm{d}\boldsymbol{v}_*.$$

Lastly, approximating $A_M = c(\rho, T)I$ gives us the Fokker-Planck collision term for one species

$$(\partial_t f)_c = \tilde{c}(\rho, T) \nabla_{\boldsymbol{v}} \cdot ((v-u)f + T \nabla_{\boldsymbol{v}} f) \, .$$

The factor $\tilde{c}(\rho, T)$ now describes the strictly positive frequency of collision. Combining this collision term with the Vlasov-Fokker-Planck equation (1.1) gives us the Vlasov-Fokker-Planck equation for one species (Montgomery and Tidman (1964)):

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{F} \cdot \nabla_{\boldsymbol{v}} f = \tilde{c}(\rho, T) \nabla_{\boldsymbol{v}} \cdot ((\boldsymbol{v} - \boldsymbol{u}) f + T \nabla_{\boldsymbol{v}} f) \, . \tag{1.4}$$

This equation does not only show up when describing the behavior of particles and molecules, but can also be found in financial modeling (Sornette (2001)), plasma physics (Lenard and Bernstein (1958)), fusion research, nanotech, machine learning and more (Risken (1992)). For the purpose of this thesis, we simplify it even further by non-dimensionalizing, taking the distribution function to not be dependent on its position nor have an external force field, having a one dimensional velocity domain and taking the collision frequency $\tilde{c}(\rho, T)$ to be one. In conclusion, the kinetic Fokker Planck equation studied in this thesis is given by

$$\partial_t f = \partial_v \left((v-u) f + T \partial_v f \right), \quad v \in \Omega. \tag{1.5}$$

As an aside; this equation is frequently reformulated through the utilization of the Maxwellian (1.3):

$$\begin{aligned} \partial_t f &= T \partial_v \left(\frac{v-u}{T} f + \partial_v f \right) = T \partial_v \left(-\frac{\partial_v M f}{M} + \partial_v f \right) = T \partial_v \left(M \frac{\partial_v f - \partial_v M f}{M^2} \right) \\ &= T \partial_v \left(M \partial_v \left(\frac{f}{M} \right) \right). \end{aligned} \tag{1.6}$$

1.1.2 Conservation & entropy properties over $\mathbb{R}$

Now that the kinetic Fokker-Planck equation (1.5) is clear, properties of the molecular distribution function f can be shown. For the purpose of this thesis we only show relevant properties for the single species one dimensional Fokker-Planck equation; however, they also hold for the general Fokker-Planck equation. First, we can assume that both the function itself as its derivative dissipate strong enough as the velocity tends to infinity as it is a distribution function. The kinetic Fokker-Planck equation over $\mathbb{R}$ is given by the non-constant coefficient partial differential equation:

$$\begin{cases} \partial_t f = \partial_v \left((v-u)f + T\partial_v f \right) & \forall v \in \mathbb{R},\ t > t_0, \\ \lim_{|v|\to\infty} f = 0 \ \& \ \lim_{|v|\to\infty} \partial_v f = 0 & \forall t > t_0, \\ f(t_0, v) = f_0(v) & \forall v \in \mathbb{R}. \end{cases} \tag{1.7}$$

As said before, the Fokker-Planck equation is fundamentally supposed to be a conservation equation, so these conservation properties are important to us. Recall the definition of the mass density ρ, or mass for short, and define the momentum and energy over $\mathbb{R}$ by

$$\rho = \int_{\mathbb{R}} f \,\mathrm{d}v, \qquad \mathcal{M} = \int_{\mathbb{R}} vf \,\mathrm{d}v, \quad \text{and} \quad \mathcal{E} = \int_{\mathbb{R}} \frac{v^2}{2} f \,\mathrm{d}v.$$

We aim to conserve these quantities as time goes on, i.e., we aim to have them stay constant over time. Consequently, we look at the derivative of them with respect to time. First, start with the mass of the distribution function:

$$\partial_t \rho = \partial_t \int_{\mathbb{R}} f \,\mathrm{d}v = \Big[\left((v-u)f + T\partial_v f \right) \Big]_{-\infty}^{\infty} = 0.$$

Clearly, because of the dissipation, the mass will be conserved over time. Now one can also look at the conservation of momentum,

$$\begin{aligned} \partial_t \mathcal{M} = \partial_t \int_{\mathbb{R}} vf \,\mathrm{d}v &= \Big[v \left((v-u)f + T\partial_v f \right) \Big]_{-\infty}^{\infty} - \int_{\mathbb{R}} \left((v-u)f + T\partial_v f \right) \mathrm{d}v \\ &= \left(-\mathcal{M} + u\rho - T[f]_{-\infty}^{\infty} \right) = \left(-\mathcal{M} + u\rho \right). \end{aligned}$$

Note that the bulk velocity from (1.2) can be rewritten as $u = \frac{\mathcal{M}}{\rho}$. Since the molecular distribution function dissipates, it follows that momentum is conserved. The conservation of energy can also be shown,

$$\partial_t \mathcal{E} = \partial_t \int_{\mathbb{R}} \frac{v^2}{2} f \, \mathrm{d}v = \left[\frac{v^2}{2} \left((v-u)f + T\partial_v f \right) \right]_{-\infty}^{\infty} - \int_{\mathbb{R}} v \left((v-u)f + T\partial_v f \right) \, \mathrm{d}v$$
$$= \left(-2\mathcal{E} + u\mathcal{M} - T[vf]_{-\infty}^{\infty} + \rho T \right) = \left(-2\mathcal{E} + u\mathcal{M} + \rho T \right).$$

Realize that $u^2\rho = u\mathcal{M}$, then, using the temperature from (1.2), $\rho T = 2\mathcal{E} - u\mathcal{M}$. Hence, energy is also conserved. It is important to now see that since the momenta are constant and the bulk velocity and temperature can be written in terms of these momenta, the bulk velocity u and temperature T should also be constant. Note that this is not generally true for the Vlasov-Fokker-Planck equation.

Two other key qualities of the kinetic Fokker-Planck equation remain: positiveness and increasing entropy. Given a positive initial condition, assume for now that the solutions to the Fokker-Planck equation are positive. This assumption is not outrageous as the equation aims to describe a distribution function. For the purpose of this thesis, the analytical solution will suffice to show this property. Now define the entropy $S(t)$ of the Fokker-Planck equation by

$$S(t) = -\int_{\mathbb{R}} f \ln(f) \, \mathrm{d}v.$$

This is well defined if we assume it will dissipate fast enough. The evolution of the entropy over time can be described as

$$\partial_t S(t) = -\partial_t \int_{\mathbb{R}} f \ln(f) \, \mathrm{d}v = -\int_{\mathbb{R}} \partial_t f \ln(f) \, \mathrm{d}v - \int_{\mathbb{R}} \partial_t f \, \mathrm{d}v$$
$$= -\int_{\mathbb{R}} \partial_t f \ln(f) \, \mathrm{d}v - \partial_t \rho = -\int_{\mathbb{R}} \partial_t f \ln(f) \, \mathrm{d}v.$$

Now see that the natural log of the Maxwellian is a second-degree polynomial. Therefore due to the conservation properties, the temporal derivative of the integral of our molecular distribution function f times the log of any Maxwellian, will be zero. Choose this Maxwellian to be the one previously described in equation (1.3), the evolution of the entropy over time can then be reformulated to

$$\partial_t S(t) = -\int_{\mathbb{R}} \partial_t f \ln\left(\frac{f}{M}\right) \, \mathrm{d}v.$$

Moreover, we can now use equation (1.6) to see that

$$\partial_t S(t) = -T \int_{\mathbb{R}} \partial_v M \partial_v \left(\frac{f}{M}\right) \ln\left(\frac{f}{M}\right) \, \mathrm{d}v = T \int_{\mathbb{R}} \frac{M^2}{f} \left(\partial_v \frac{f}{M}\right)^2 \, \mathrm{d}v \geq 0.$$

From the positiveness of the distribution function f, the increasing nature of the entropy over time becomes clear. Of particular note is that the change of entropy is only zero, or has maximal entropy, when the distribution function is equal to the Maxwellian. Thus, our distribution function should tend to the Maxwellian (1.3) as time goes to infinity.

1.1.3 Analytical solutions over $\mathbb{R}$

Asides from deriving properties of the solution, an analytical solution can be found for the kinetic Fokker-Planck equation. Recall the partial differential equation arising from the Fokker-Planck equation (1.7):

$$\begin{cases} \partial_t f = \partial_v \left((v-u)f + T\partial_v f\right) & \forall v \in \mathbb{R},\ t > t_0 \\ \lim_{|v|\to\infty} f(t,v) = 0 \ \& \ \lim_{|v|\to\infty} \partial_v f(t,v) = 0 & \forall t > t_0, \\ f(t_0, v) = f_0(v) & \forall v \in \mathbb{R}, \end{cases}$$

where u and T can be found through the initial condition, since they are constant. First take the Fourier transform $\mathcal{F}$ of f denoted by $\hat{f}$. The partial differential equation then turns into

$$\begin{cases} \partial_t \hat{f} = -k\partial_k \hat{f} + \left(iuk - Tk^2\right)\hat{f} & k \in \mathbb{R},\ t > t_0, \\ \hat{f}(t_0, k) = \hat{f}_0(k) & k \in \mathbb{R}. \end{cases}$$

Now take the $-k$ out of the right hand side and define

$$\hat{g} = e^{\frac{T}{2}k^2 - iuk}\hat{f}. \tag{1.8}$$

The Fourier transformed partial differential equation is then equivalent to solving

$$\begin{cases} \partial_t \hat{g} + k\partial_k \hat{g} = 0 & k \in \mathbb{R},\ t > t_0, \\ \hat{g}(t_0, k) = e^{\frac{T}{2}k^2 - iuk}\hat{f}_0 & k \in \mathbb{R}. \end{cases}$$

The solution $\hat{g}$ can be found by using the well-known method of characteristics, note that here $k = k_0 e^{(t-t_0)}$. Therefore,

$$\hat{g}(t,k) = \hat{g}(t_0, ke^{-(t-t_0)}) = \exp\left(\frac{T}{2}k^2 e^{-2(t-t_0)} - iuke^{-(t-t_0)}\right)\hat{f}_0(ke^{-(t-t_0)}).$$

Using the original definition of $\hat{g}$ (1.8), we can get the Fourier transform of the solution is

$$\hat{f} = \exp\left(-\frac{T}{2}k^2\left(1-e^{-2(t-t_0)}\right) + iuk\left(1-e^{-(t-t_0)}\right)\right)\hat{f}_0(ke^{-(t-t_0)}) = h_1 h_2.$$

All that is left to do is take the inverse Fourier transform of this. This can be done using the convolution theorem. First finding the inverse Fourier transform of h_1 by completing the squares gives us

$$\mathcal{F}^{-1}\left[h_1\right](x) = \sqrt{\frac{1}{2\pi T(1-e^{-2(t-t_0)})}}\exp\left(-\frac{(x-u(1-e^{-(t-t_0)}))^2}{2T(1-e^{-2(t-t_0)})}\right).$$

Moreover,

$$\mathcal{F}^{-1}\left[h_2\right](x) = e^{(t-t_0)} f_0\left(xe^{(t-t_0)}\right).$$

Thus, by using the convolution theorem over the velocity, the solution is given by

$$f(t,v) = \left[\sqrt{\frac{1}{T(1-e^{-2(t-t_0)})}}\exp\left(-\frac{(v-u(1-e^{-(t-t_0)}))^2}{2T(1-e^{-2(t-t_0)})}\right)\right] * \left[e^{(t-t_0)} f_0\left(ve^{(t-t_0)}\right)\right]. \tag{1.9}$$

Given a positive initial condition, it is now clear that our assumption of a positive solution is true.

1.1.4 Conservation properties over a bounded domain

For all of the previous results we depended mainly on two assumptions: we have an unbounded domain $\mathbb{R}$ and the solution dissipates as the velocity tends to infinity. One should now ask themselves what happens when these assumptions no longer hold. Define a bounded connected domain $\Omega = [v_{\min}, v_{\max}]$ and impose, instead of dissipation, a no-flux boundary condition. The partial differential equation is then given by

$$\begin{cases} \partial_t f = \partial_v\left((v-u)f + T\partial_v f\right) & \forall v \in \Omega,\ t > 0, \\ (v-u)f + T\partial_v f = 0 & \forall v \in \partial\Omega,\ t > 0, \\ f(0,v) = f_0(v) & \forall v \in \Omega. \end{cases} \tag{1.10}$$

For simplicity, time starts at zero. The previous choices of the bulk velocity and temperature (1.2) were essential to the conservation of all momenta. Our choice in bulk velocity u and temperature T over the bounded domain will be made such that the conservation properties also hold for the bounded domain. Redefine the mass density, momentum and energy over this bounded domain Ω, respectively by

$$\rho = \int_\Omega f \,\mathrm{d}v, \qquad \mathcal{M} = \int_\Omega v f \,\mathrm{d}v, \quad \text{and} \quad \mathcal{E} = \int_\Omega \frac{v^2}{2} f \,\mathrm{d}v.$$

We again aim to conserve these quantities, over time. First, we start with the mass of the distribution function:

$$\partial_t \rho = \partial_t \int_\Omega f \,\mathrm{d}v = \left[(v-u)f + T\partial_v f\right]_{v_{\min}}^{v_{\max}} = 0.$$

This holds as we imposed no-flux boundary conditions. Now one can also look at the derivative of the momentum with respect to time

$$\begin{aligned}\partial_t \mathcal{M} = \partial_t \int_\Omega v f \,\mathrm{d}v &= \left[v\left((v-u)f + T\partial_v f\right)\right]_{v_{\min}}^{v_{\max}} - \int_\Omega \left((v-u)f + T\partial_v f\right) \mathrm{d}v \\ &= \left(-\mathcal{M} + u\rho - T\left[f\right]_{v_{\min}}^{v_{\max}}\right),\end{aligned}$$

and the derivative of the energy with respect to time

$$\begin{aligned}\partial_t \mathcal{E} = \partial_t \int_\Omega \frac{v^2}{2} f \,\mathrm{d}v &= \left[\frac{v^2}{2}\left((v-u)f + T\partial_v f\right)\right]_{v_{\min}}^{v_{\max}} - \int_\Omega v\left((v-u)f + T\partial_v f\right) \mathrm{d}v \\ &= \left(-2\mathcal{E} + u\mathcal{M} - T[vf]_{v_{\min}}^{v_{\max}} + \rho T\right).\end{aligned}$$

From these results, the bulk velocity and temperature are chosen such that momentum and energy are conserved, i.e., system (1.11) holds:

$$\begin{cases} \mathcal{M} - u\rho + T(f(t, v_{\max}) - f(t, v_{\min})) = 0, \\ 2\mathcal{E} - u\mathcal{M} - T\rho + T(v_{\max} f(t, v_{\max}) - v_{\min} f(t, v_{\min})) = 0. \end{cases} \tag{1.11}$$

Unlike the unbounded domain, the bulk velocity and temperature will depend on time and on the value of the function on the boundary. While one might consider imposing Dirichlet boundary conditions, this could result in contradictory boundary conditions as no-flux boundary conditions are essential to mass conservation. Formulating a Fokker-Planck

equation that conserves momentum and energy over a bounded domain proves difficult. Consequently, discussing properties such as entropy or asymptotic behavior becomes intricate as well.

1.2 The objective

The main objective of a partial differential equation is to solve it. There already exists a solution to the Fokker-Planck equation over $\mathbb{R}$, why should we develop numerical methods for it? The initial consideration that arises pertains to the potential non-solvability of the convolution integral given an initial condition. To address this, one might resort to quadrature rules. However, in order to attain a physically meaningful solution, it is imperative that conservation properties are preserved. That being said, it appears challenging to efficiently achieve this using conventional quadrature methods.

Consequently, one turns to other methods. The classic methods would be finite difference, volume, or element methods. These all require truncation, for which further complications arise. Determining bulk velocity and temperature values that satisfy the conservation constraints necessitates knowledge of the solution at the boundaries. Consequently, an additional boundary condition should be imposed, which could potentially conflict with the no-flux boundary condition. Even if they could be found, solving the Fokker-Planck equation might then prove impossible.

Moreover, the previous section of this thesis only discusses the single-species Fokker-Planck equation in one dimension, with velocity as the varying parameter. However, the equation of primary interest, denoted as equation (1.4), describes a phase space encompassing position and velocity in three dimensions. Furthermore, the bulk velocity and temperature are spatially and temporally dependent, rendering the partial differential equation unsolvable at the time of this thesis.

Hence, it is vital to develop a numerical method that upholds the conservation properties of the Fokker-Planck equation within a bounded domain. This thesis focuses on applying finite elements methods on a one-dimensional velocity space. This can then be extended to a multi-dimensional velocity space. From this method class, we focus on Discontinuous

Galerkin (DG) methods. These methods excel in approximating advection type problems. In order to approximate the diffusion term, specific DG methods are created. Numerous of the DG methods are available with various numerical fluxes are available. For our purposes, we explore and compare both the local discontinuous Galerkin method and the recovery discontinuous Galerkin method applied to the diffusion term.

Chapter 2

DISCONTINUOUS GALERKIN METHODS

Since the kinetic Fokker-Planck seems to be impossible to solve in various cases, one reaches for numerical approximations. In the world of partial differential equations, three classical categories of numerical methods immediately spring to mind: finite difference methods, finite volume methods, and finite element methods. Each of these categories possess their own set of advantages and disadvantages. For example, finite volume methods are designed for conservation equations like these, however they are difficult to implement for higher order differential equations. It is due to this motivation (and intellectual curiosity) that this thesis explores and evaluates the Discontinuous Galerkin method, which falls under the finite element method category and is specialized for advection type equations.

In general, finite element methods involve subdividing the domain into elements and solving the weak formulation of the partial differential equation within these elements. Galerkin methods approximate the solution using functions from a finite-dimensional function space, expressed as linear combinations of the bases of said space. Discontinuous Galerkin methods, as described in Hesthaven and Warburton (2007), approximate the solution in a similar manner over all elements but do not impose continuity constraints across element boundaries.

In this thesis, we will specifically focus on the application of local and recovery discontinuous Galerkin methods to the one-dimensional Fokker-Planck equation. After which, the stability of the methods are superficially discussed. However, before delving into the details, we will provide an explanation of the notation.

2.1 Notation

First divide the domain Ω into N equally spaced cells

$$v_{\min} = v_{\frac{1}{2}} < v_{\frac{3}{2}} < \cdots < v_{N+\frac{1}{2}} = v_{\max},$$

and denote the elements and spatial spacing as

$$e_k = (v_{k-\frac{1}{2}}, v_{k+\frac{1}{2}}), \text{ and } h = \frac{v_{\max} - v_{\min}}{N}.$$

For the elements, index k will consistently be used to refer to the k-th element. Then define the notations involving $-$ and $+$ on a general function f. Let $f^-(v_{k+\frac{1}{2}})$ be the value of f coming from the negative direction of $v_{k+\frac{1}{2}}$ and $f^+(v_{k+\frac{1}{2}})$ be the value of f from the positive direction of $v_{k+\frac{1}{2}}$, otherwise denoted as

$$f^-(v_{k+\frac{1}{2}}) = \lim_{v \to v^-_{k+\frac{1}{2}}} f(v) \quad \text{and} \quad f^+(v_{k+\frac{1}{2}}) = \lim_{v \to v^+_{k+\frac{1}{2}}} f(v).$$

A spatially discretized function is denoted by subscript h and temporally discretized function is denoted by superscript n. Therefore, $\boldsymbol{f}_h^{k,n}$ denotes the discretized form $\boldsymbol{f}_h$ over element k at time step n.

As for function spaces, the space of N_p-th order polynomial functions over e_k is given by $\mathbb{P}^{N_p}(e_k)$ and the space of piecewise, over every e_k, N_p-th order polynomial functions is defined by $\tilde{\mathbb{P}}^{N_p}(\Omega) = \bigcup_k \mathbb{P}^{N_p}(e_k)$. Also define the one-dimensional Sobolev space

$$H^p(e_k) = \left\{ f \;\middle|\; \left(\sum_{i=0}^{p} \int_{e_k} \left| f^{(i)} \right|^2 \mathrm{d}v \right)^{\frac{1}{2}} < \infty \right\}.$$

Lastly, define the inner product

$$\langle \psi, \varphi \rangle_k = \int_{e_k} \psi \overline{\varphi} \, \mathrm{d}v,$$

and the corresponding norm, or the L^2-norm,

$$\|\varphi\|^2_{L^2(e_k)} = \langle \varphi, \varphi \rangle_k.$$

2.2 Local Discontinuous Galerkin method

First, the local discontinuous Galerkin (LDG) method is worked out, implemented and discussed. This method is designed to solve the partial differential equations on a local level. Instead of solving over the entire domain, it has been divided into the elements e_k,

which we now intend to solve using local information. Specifically, the LDG method is defined by its choice in flux for the diffusion term seen in (2.4). Since it is a discontinuous Galerkin method, the continuity over elements is not a concern.

2.2.1 Weak formulation

In order to get to the weak formulation used for the method, define an auxiliary function

$$g = \sqrt{T}\partial_v f. \tag{2.1}$$

Then, the kinetic Fokker-Planck equation (1.5) can be written as a system of advection equations

$$\begin{aligned} \partial_t f &= \partial_v \left((v-u)f + \sqrt{T}g \right) \quad v \in \Omega, \\ g &= \sqrt{T}\partial_v f \qquad\qquad\qquad\qquad v \in \Omega, \end{aligned}$$

where the bulk velocity and temperature are yet to be specified, but should be considered dependent on time, as Ω is bounded. Now the appropriate weak formulation must be found. For this, both equations can be multiplied by test functions. For the sake of simplicity, we take these to be the same, namely, $\varphi \in H^1(e_k)$. Integrate over the k^{th} element to get

$$\begin{aligned} \langle \partial_t f, \varphi \rangle_k &= \left\langle \partial_v \left((v-u)f + \sqrt{T}g \right), \varphi \right\rangle_k \\ \langle g, \varphi \rangle_k &= \left\langle \sqrt{T}\partial_v f, \varphi \right\rangle_k \end{aligned}$$

Integrating both equations by parts results in the weak formulation for a fixed time t:

Problem f, g: Find $f, g \in H^1(e_k)$ such that for all $\varphi \in H^1(e_k)$ one has

$$\begin{aligned} &\langle \partial_t f, \varphi \rangle_k + \langle (v-u)f, \partial_v \varphi \rangle_k + \sqrt{T}\langle g, \partial_v \varphi \rangle_k = \left[(v-u)f\varphi + \sqrt{T}g\varphi \right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \\ &\langle g, \varphi \rangle_k + \sqrt{T}\langle f, \partial_v \varphi \rangle_k = \sqrt{T}[f\varphi]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \end{aligned}$$

where we impose the no-flux boundary conditions at the boundary elements,

$$(v-u)f + \sqrt{T}g = 0, \qquad \text{at } v = v_{\min}, v_{\max}.$$

2.2.2 Spatial discretization

In order to numerically compute solutions to this weak formulation we first spatially discretize over the velocity domain using the Galerkin method. Approximate both functions by a piecewise N_p-th-order polynomial over all N elements with $N_p + 1$ grid points at the k^{th} element:

$$f(t,v) \approx f_h(t,v) = \bigoplus_{k=1}^{N} f_h^k(t,v) \qquad \text{and} \qquad g(t,v) \approx g_h(t,v) = \bigoplus_{k=1}^{N} g_h^k(t,v),$$

where $g_h^k(t,v), f_h^k(t,v) \in \mathbb{P}^{N_p}(e_k)$. Now, choose $(\varphi_i)_{i=1,\dots,N_p+1}$ to be a basis of $\mathbb{P}^{N_p}(e_k)$. Then, f and g over e_k is approximated by

$$f|_{e_k} \approx f_h^k(t,v) = \sum_{j=1}^{N_p+1} \hat{f}_j^k(t)\varphi_j^k(v) \qquad \text{and} \qquad g|_{e_k} \approx g_h^k(t,v) = \sum_{j=1}^{N_p+1} \hat{g}_j^k(t)\varphi_j^k(v). \tag{2.2}$$

As we now essentially have two choices for the values of f_h at the boundary of an element, we substitute the function values at the element edges by specific numerical fluxes. When looking at the drift term, one naturally considers the upwind flux for the numerical implementation. Instead of this, and following Hakim et al. (2020), we use a more restraining flux, namely, the Lax-Friederichs flux:

$$\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right), \text{ with } \alpha = \max_{v\in\Omega}(|v-u|). \tag{2.3}$$

Note that α is dependent on u and can thus be dependent on time. On the other hand, alternating fluxes are used for the diffusion term so that the diffusion is updated by alternating directions, therefore we set

$$f_h(v_{k+\frac{1}{2}}) = f_h^-(v_{k+\frac{1}{2}}) \qquad \text{and} \qquad g_h(v_{k+\frac{1}{2}}) = g_h^+(v_{k+\frac{1}{2}})$$

This choice of alternating fluxes defines the LDG method. Both reduce the computational complexity, but are first-order accurate. To recap, this method aims to find f_h and g_h such

that for all test functions φ in the Sobolev space over e_k,

$$\begin{aligned}
&\langle \partial_t f_h, \varphi\rangle_k + \langle (v-u) f_h, \partial_v \varphi\rangle_k + \sqrt{T}\langle g_h, \partial_v \varphi\rangle_k \\
&\qquad = \left[\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right)\varphi + \sqrt{T} g_h^+ \varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \qquad (2.4)\\
&\langle g_h, \varphi\rangle_k + \sqrt{T}\langle f_h, \partial_v \varphi\rangle_k = \sqrt{T}[f_h^- \varphi]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}.
\end{aligned}$$

In order for this to work, $g_h^+(v_{\max})$ and $f_h^-(v_{\min})$ must be computable. This is, however, outside of the domain Ω. Therefore, we choose to only use the value on the inside of the domain. As we only have one choice for this value, we do not denote the superscript for the function at the boundary. Since all polynomials are in the Sobolev space, the test functions can be taken as our basis functions φ_i. This turns system (2.4) into two systems of $N_p + 1$ equations:

$$\begin{aligned}
&\langle \partial_t f_h, \varphi_i\rangle_k + \langle (v-u) f_h, \partial_v \varphi_i\rangle_k + \sqrt{T}\langle g_h, \partial_v \varphi_i\rangle_k \\
&\qquad = \left[\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right)\varphi_i + \sqrt{T} g_h^+ \varphi_i\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \qquad (2.5)\\
&\langle g_h, \varphi_i\rangle_k + \sqrt{T}\langle f_h, \partial_v \varphi_i\rangle_k = \sqrt{T}[f_h^- \varphi_i]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}.
\end{aligned}$$

In order to efficiently compute these, define the following matrices

$$M_{ij}^k = \langle \varphi_i, \varphi_j\rangle_k, \quad V_{ij}^k = \langle \partial_v \varphi_i, v\varphi_j\rangle_k, \quad \text{and} \quad P_{ij}^k = \langle \partial_v \varphi_i, \varphi_j\rangle_k, \quad \text{with } M^k, V^k, P^k \in \mathbb{R}^{N_p+1\times N_p+1}.$$

It is critical for conservation that these integrals are computed exactly. To this end, one can use Gauss quadrature, since the test functions are polynomials. Now also define the vectors

$$(\boldsymbol{b}_f^k)_i = \left[\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right)\varphi_i + \sqrt{T} g_h^+ \varphi_i\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \text{ and } (\boldsymbol{b}_g^k)_i = [f_h^- \varphi_i]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}},$$

with $\boldsymbol{b}_f^k, \boldsymbol{b}_g^k \in \mathbb{R}^{N_p+1}$. As all elements will be treated the same, with certain exceptions at the boundary, the k superscript will only be used when specification is needed. Using both

our Galerkin approximations (2.2) and filling them into (2.5) the LDG method then comes down to solving the system,

$$M\partial_t \hat{\boldsymbol{f}}_h + (V - uP)\hat{\boldsymbol{f}}_h + \sqrt{T}P\hat{\boldsymbol{g}}_h = \boldsymbol{b}_f, \qquad (2.6)$$
$$M\hat{\boldsymbol{g}}_h + \sqrt{T}P\hat{\boldsymbol{f}}_h = \sqrt{T}\boldsymbol{b}_g,$$

over all elements, where $\hat{\boldsymbol{f}}_h$ and $\hat{\boldsymbol{f}}_h$ are collumn vectors comprised of the coefficients $\hat{f}_i^k$ and $\hat{g}_i^k$. Notice that $\hat{\boldsymbol{f}}_h$, $\hat{\boldsymbol{g}}_h$, u, T, $\boldsymbol{b}_f$ and $\boldsymbol{b}_g$ remain dependent on time.

2.2.3 Discretization with respect to time

With an eye on numerically approximating the solution, time must be discretized. Truncate the time from $t = 0$ until $t = t_{\max}$ and divide it into N_t equidistant time steps. Denote the distance between time steps by Δt and denote any time dependent function at time step t_n with the superscript of n. Thereafter, a choice in approximation for the time derivative must be made. This is done in two ways: the well-known forward Euler method and the second-order strong stability-preserving Runga-Kutta method described in Gottlieb et al. (2011). This fully discretizes the Fokker-Planck equation, thus making it possible to computationally solve the problem. Moreover, since these are explicit methods, one can compute the elements in parallel. In the end, an analysis of the conservation properties of the discrete model is conducted.

Forward Euler

Using the forward Euler method to approximate the time derivative, from (2.5), the fully discrete weak formulation is then given by

Problem LDG (Forward Euler): Find $f_h^{n+1}, g_h^n \in \mathbb{P}^{N_p}(e_k)$ such that for all $\varphi \in H^1(e_k)$ one has

$$\begin{aligned}\langle f_h^{n+1}, \varphi\rangle_k + \Delta t\langle (v-u^n) f_h^n, \partial_v\varphi\rangle_k + \Delta t\sqrt{T^n}\langle g_h^n, \partial_v\varphi\rangle_k = \langle f_h^n, \varphi\rangle_k \\ + \Delta t\left[\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right)\varphi + \sqrt{T^n}\,(g_h^n)^+\,\varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}\end{aligned} \qquad (2.7a)$$

$$\langle g_h^n, \varphi\rangle_k + \sqrt{T^n}\langle f_h^n, \partial_v\varphi\rangle_k = \sqrt{T^n}[(f_h^n)^-\,\varphi]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \tag{2.7b}$$

where we impose the no-flux boundary conditions at the boundary elements,

$$(v-u^n)f_h^n + \sqrt{T^n}g_h^n = 0, \quad \text{at } v = v_{\min}, v_{\max}. \tag{2.7}$$

Now, given an initial condition for f, the projection of $f(0,v)$ can be found onto $\tilde{\mathbb{P}}(\Omega)$ to get f_h^0. One can then sequentially solve these equations to get the solution at all time steps.

At last the conservation properties of our fully discrete equation can be analyzed. The discrete momenta are now different from the theoretical momenta. In the continuous case, it was proven that the momenta are conserved. However, this may no longer be the case because of the errors introduced by the spatial and temporal discretizations. They are defined as

$$\rho_h^n = \int_\Omega f_h^n \,\mathrm{d}v, \qquad \mathcal{M}_h^n = \int_\Omega v f_h^n \,\mathrm{d}v, \quad \text{and} \quad \mathcal{E}_h^n = \int_\Omega \frac{v^2}{2} f_h^n \,\mathrm{d}v \tag{2.8}$$

These momenta should stay conserved in order to have an approximation that is consistent with the solution to the Fokker-Planck equation over continuous time and space. Conservation of mass can easily be check by summing (2.7a) over all elements and with test function $\varphi = 1$:

$$\begin{aligned}
\rho_h^{n+1} &= \sum_k \int_{e_k} f_h^{n+1} dv = \sum_k \langle f_h^{n+1}, 1\rangle_k \\
&= \sum_k \int_{e_k} f_h^n + \Delta t \sum_k \left[\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + \sqrt{T^n}\,(g_h^n)^+\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \\
&= \rho_h^n + \Delta t \sum_k \left[\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + \sqrt{T^n}\,(g_h^n)^+\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}
\end{aligned}$$

Take a closer look at the sum over all elements. Expanding this, only the first and last terms remain:

$$
\begin{aligned}
&\sum_k \left[\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + \sqrt{T^n}\,(g_h^n)^+\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \\
&= \frac{1}{2}\left((v_{\frac{3}{2}}-u^n)((f_h^n)^+(v_{\frac{3}{2}}) + (f_h^n)^-(v_{\frac{3}{2}})) - \alpha^n((f_h^n)^-(v_{\frac{3}{2}}) - (f_h^n)^+(v_{\frac{3}{2}}))\right) + \sqrt{T^n}\,(g_h^n)^+(v_{\frac{3}{2}}) \\
&\quad - (v_{\frac{1}{2}}-u^n)f_h^n(v_{\frac{1}{2}}) - \sqrt{T^n}g_h^n(v_{\frac{1}{2}}) \\
&\quad + \frac{1}{2}\left((v_{\frac{5}{2}}-u^n)((f_h^n)^+(v_{\frac{5}{2}}) + (f_h^n)^-(v_{\frac{5}{2}})) - \alpha^n((f_h^n)^-(v_{\frac{5}{2}}) - (f_h^n)^+(v_{\frac{5}{2}}))\right) + \sqrt{T^n}\,(g_h^n)^+(v_{\frac{5}{2}}) \\
&\quad - \frac{1}{2}\left((v_{\frac{3}{2}}-u^n)((f_h^n)^+(v_{\frac{3}{2}}) + (f_h^n)^-(v_{\frac{3}{2}})) - \alpha^n((f_h^n)^-(v_{\frac{3}{2}}) - (f_h^n)^+(v_{\frac{3}{2}}))\right) - \sqrt{T^n}\,(g_h^n)^+(v_{\frac{3}{2}}) \\
&\quad \vdots \\
&\quad + (v_{N+\frac{1}{2}}-u^n)f_h^n(v_{N+\frac{1}{2}}) + \sqrt{T^n}g_h^n(v_{N+\frac{1}{2}}) \\
&\quad - \frac{1}{2}\left((v_{N-\frac{1}{2}}-u^n)((f_h^n)^+(v_{N-\frac{1}{2}}) + (f_h^n)^-(v_{N-\frac{1}{2}})) - \alpha^n((f_h^n)^-(v_{N-\frac{1}{2}}) - (f_h^n)^+(v_{N-\frac{1}{2}}))\right) - \sqrt{T^n}\,(g_h^n)^+(v_{N-\frac{1}{2}}) \\
&= (v_{\max}-u^n)f_h^n(v_{\max}) + \sqrt{T^n}g_h^n(v_{\max}) - (v_{\min}-u^n)f_h^n(v_{\min}) - \sqrt{T^n}g_h^n(v_{\min})
\end{aligned}
$$

This recurrent pattern, often referred to as the "telescoping principle", is featured prominently throughout the conservation analysis. Notably, our analysis now reveals that since the no-flux-like boundary condition (2.7) is imposed, mass is conserved. Now, look at the conservation of momentum. For this purpose, we sum (2.7a) over all elements with $\varphi = v$:

$$
\begin{aligned}
\mathcal{M}_h^{n+1} &= \sum_k \left\langle f_h^{n+1}, v\right\rangle_k \\
&= \mathcal{M}_h^n - \Delta t\left(\sum_k \langle (v-u^n)f_h^n, 1\rangle_k + \sqrt{T^n}\langle g_h^n, 1\rangle_k\right) \\
&\qquad + \Delta t \sum_k \left[\left(\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + \sqrt{T^n}\,(g_h^n)^+\right) v\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \\
&\overset{(2.7b)}{=} \mathcal{M}_h^n - \Delta t\left(\mathcal{M}_h^n - u^n\rho_h^n + T^n\sum_k [(f_h^n)^-]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}\right) \\
&= \mathcal{M}_h^n - \Delta t\left(\mathcal{M}_h^n - u^n\rho_h^n + T^n\left(f_h^n(v_{\max}) - f_h^n(v_{\min})\right)\right).
\end{aligned}
$$

Here the no-flux boundary conditions and telescoping principle are used. Lastly, for the conservation of energy, sum (2.7a) over all elements with $\varphi = \frac{v^2}{2}$.

$$\begin{aligned}
\mathcal{E}_h^{n+1} &= \sum_k \langle f_h^{n+1}, \frac{v^2}{2} \rangle_k \\
&= \mathcal{E}_h^n - \Delta t \left(2\mathcal{E}_h^n - u^n \mathcal{M}_h^n + T^n \left(\sum_k [(f_h^n)^- v]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} - \rho_h^n \right) \right) \\
&\qquad + \Delta t \sum_k \left[\left(\frac{1}{2} \left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n ((f_h^n)^- - (f_h^n)^+) \right) + \sqrt{T^n} \, (g_h^n)^+ \right) \frac{v^2}{2} \right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \\
&= \mathcal{E}_h^n - \Delta t \left(2\mathcal{E}_h^n - u^n \mathcal{M}_h^n - T^n \rho_h^n + T^n \left(f_h^n(v_{\max}) v_{\max} - f_h^n(v_{\min}) v_{\min} \right) \right).
\end{aligned}$$

From these results we see that, in order to conserve momentum and energy, one must have that u^n and T^n are chosen such that

$$\begin{cases} \mathcal{M}_h^n - u^n \rho_h^n + T^n \left(f_h^n(v_{\max}) - f_h^n(v_{\min}) \right) = 0 \\ 2\mathcal{E}_h^n - u^n \mathcal{M}_h^n - T^n \rho_h^n + T^n \left(f_h^n(v_{\max}) v_{\max} - f_h^n(v_{\min}) v_{\min} \right) = 0. \end{cases} \tag{2.9}$$

Since this system only depends on the bulk velocity u^n, temperature T^n and parameters using f_h^n, one can compute the bulk velocity and temperature at time t^n.

This result would not have been directly computable if we did not specify a choice for the values of f_h at the boundary of the domain Ω. Recall that we chose to use the value on the inside of domain at equation (2.4). Say we did not make a choice and reintroduce the notation $(f_h^n)^-(v_{\min})$ and $(g_h^n)^+(v_{\max})$ as variables. We would then have $(f_h^n)^-(v_{\min})$ in both restrictions (2.9). Solving these two nonlinear equations with three variables would be impossible. Naturally, the no-flux boundary conditions (2.7) can be involved, thus adding another unknown $(g_h^n)^+(v_{\max})$. One might think that solving these four restrictions for conservation with four variables will result in a conserving method. Even though the system is non-linear, an analytical solution can easily be found. However, this is not where the problem lies. Suppose we have f_h^n; in order to calculate g^n we need T^n, but to calculate that we need g^n, there's the rub. This is why we do not involve the no-flux boundary conditions in order to calculate the bulk velocity and temperature.

Conservation itself holds as long as $(f_h^n)^-(v_{\min})$ is chosen consistently to be one thing for

both the computations of the bulk velocity and temperature, and the right hand side of g_h^n. The choice of $(f_h^n)^-(v_{\min})$ must then be made when looking at consistency of the solution. We hope to have temporally discrete bulk velocity and temperature that is consistent with the time continuous case (1.11). This is why instead of $(f_h^n)^-(v_{\min})$, we chose $(f_h^n)^+(v_{\min})$. This choice acts as if f_h^n flows continuously outside of the domain. Thus, solving linear system (2.9) results in a u^n and T^n that conserves the momenta.

To get these conservation results, we chose the test functions to be a first and second-order polynomial. Therefore, in order to maintain conservation in this scheme, we need N_p to be at least two. This does not mean that conservation generally does not hold for linear and parabolic test functions, see Hakim et al. (2020).

Now that the choice of bulk velocity and temperature are clear, a concrete algorithm can be created. Using the forward Euler method eventually turns (2.6) into

$$M\hat{\boldsymbol{f}}_h^{n+1} = M\hat{\boldsymbol{f}}_h^n + \Delta t\left(\boldsymbol{b}_f^n - (V - u^n P)\hat{\boldsymbol{f}}_h^n - \sqrt{T^n}P\hat{\boldsymbol{g}}_h^n\right), \tag{2.10a}$$

$$M\hat{\boldsymbol{g}}_h^n = \sqrt{T^n}\boldsymbol{b}_g^n - \sqrt{T^n}P\hat{\boldsymbol{f}}_h^n. \tag{2.10b}$$

These are more easily solveable. The algorithm for the LDG method is given by Algorithm 1.

Algorithm 1 Local Galerkin method: forward Euler

Input: $f(v,0)$

1: Build grids, test functions & matrices.
2: Compute f_h^0.
3: Compute $\rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$ with f_h^0. ▷ *using* (2.8)
4: Compute u^0 & T^0 with $f_h^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$. ▷ *using* (2.9)
5: Compute g_h^0 with f_h^0 & T^0. ▷ *using* (2.10b)
6: **for** $n = 0$ to $N_t - 1$ **do**
7: Compute f_h^{n+1} with $f_h^n,\ g_h^n,\ u^n$ & T^n. ▷ *using* (2.10a)
8: Compute $\rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$ with f_h^{n+1}. ▷ *using* (2.8)
9: Compute u^{n+1} & T^{n+1} with $f_h^{n+1},\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$. ▷ *using* (2.9)
10: Compute g_h^{n+1} with f_h^{n+1} & T^{n+1}. ▷ *using* (2.10b)

Strong Stability-Preserving Runge-Kutta

As the forward Euler method is a first-order method and the DG methods are order N_p+1, a second-order method for the time discritization might be more suitable. Generally, the best idea would be to have a method with order N_p+1 for each order of polynomial. For now, look at the second-order Strong Stability-Preserving Runge-Kutta method (SSP-RK2). This method aims to preserve the stability of the forward Euler method while making it a second-order method. Given a differential equation

$$\partial_t f = F(f),$$

the SSP-RK2 method seeks to approximate f at time t^{n+1}, given f at time t^n, by

$$\begin{aligned} f^{(1)} &= f^n + \Delta t F(f^n), \\ f^{n+1} &= \frac{1}{2}\left(f^n + f^{(1)} + \Delta t F(f^{(1)})\right). \end{aligned} \tag{2.11}$$

Here a temporary step $f^{(1)}$ is created. The notation for all time steps will be the same and different, being superscript n or (1). Functions or parameters will use this notation when it is related to their respective time. In order to apply this method and look at the conservation properties, we look at equation (2.4) and apply it to function $\partial_t\langle f_h, \varphi\rangle$. Note that for both the time step and temporary time step we compute function g_h, bulk velocity and temperature. The bulk velocity and temperature should use the momenta corresponding to the correct step of f_h. The discrete weak formulation is then given by

Problem LDG (SSP-RK2): Find $f_h^{n+1}, g_h^n, f_h^{(1)}, g_h^{(1)} \in \mathbb{P}^{N_p}(e_k)$ such that for all $\varphi \in H^1(e_k)$ one has

$$\begin{aligned} &\langle f_h^{(1)}, \varphi\rangle_k + \Delta t\langle (v-u^n) f_h^n, \partial_v\varphi\rangle_k + \Delta t\sqrt{T^n}\langle g_h^n, \partial_v\varphi\rangle_k = \langle f_h^n, \varphi\rangle_k \\ &\quad + \Delta t\left[\left(\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + \sqrt{T^n}\,(g_h^n)^+\right)\varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} \end{aligned} \tag{2.12a}$$

$$\langle g_h^n, \varphi\rangle_k + \sqrt{T^n}\langle f_h^n, \partial_v\varphi\rangle_k = \sqrt{T^n}[(f_h^n)^-\,\varphi]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}. \tag{2.12b}$$

$$\begin{aligned}
&\langle f_h^{n+1},\varphi\rangle_k + \frac{1}{2}\left(\Delta t\langle (v-u^{(1)})f_h^{(1)},\partial_v\varphi\rangle_k + \Delta t\sqrt{T^{(1)}}\langle g_h^{(1)},\partial_v\varphi\rangle_k\right)\\
&\quad = \frac{1}{2}\left(\langle f_h^n,\varphi\rangle_k + \langle f_h^{(1)},\varphi\rangle_k\right) \qquad (2.12c)\\
&\qquad + \frac{\Delta t}{2}\left[\left(\frac{1}{2}\left((v-u^{(1)})(\left(f_h^{(1)}\right)^+ + \left(f_h^{(1)}\right)^-) - \alpha^{(1)}(\left(f_h^{(1)}\right)^- - \left(f_h^{(1)}\right)^+)\right) + \sqrt{T^{(1)}}\left(g_h^{(1)}\right)^+\right)\varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}
\end{aligned}$$

$$\langle g_h^{(1)},\varphi\rangle_k + \sqrt{T^{(1)}}\langle f_h^{(1)},\partial_v\varphi\rangle_k = \sqrt{T^{(1)}}\Big[\left(f_h^{(1)}\right)^-\varphi\Big]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \qquad (2.12d)$$

where we impose the no-flux boundary conditions at the boundary elements for every step,

$$(v-u^{(1)})f_h^{(1)} + \sqrt{T^{(1)}}g_h^{(1)} = 0, \quad \text{and} \quad (v-u^n)f_h^n + \sqrt{T^n}g_h^n = 0, \quad \text{at } v = v_{\min}v_{\max}. \qquad (2.13)$$

Again, $f_h^{(1)}$ and f_h^n only take one value at the boundary of the domain. Now the conservation qualities are looked at and restrictions on u^n, T^n, $u^{(1)}$ & $T^{(1)}$ will be made. Just as before, sum (2.12c) over all elements with $\varphi = 1$. As writing the numerical fluxes becomes convoluted, the telescoping principal is once again applied:

$$\begin{aligned}
\rho_h^{n+1} &= \sum_k \frac{1}{2}\left(\langle f_h^n,1\rangle_k + \langle f_h^{(1)},1\rangle_k\right) + \frac{\Delta t}{2}\left[(v-u^{(1)})f_h^{(1)} + \sqrt{T^{(1)}}g_h^{(1)}\right]_{v_{\min}}^{v_{\max}}\\
&\overset{(2.12a)}{=} \sum_k \langle f_h^n,1\rangle_k + \frac{\Delta t}{2}\left[(v-u^n)f_h^n + \sqrt{T^n}g_h^n\right]_{v_{\min}}^{v_{\max}}\\
&= \rho_h^n
\end{aligned}$$

It is clear that conservation of mass holds as the no-flux boundary condition is imposed for both the temporary function $f_h^{(1)}$ and the function f_h^n at time step t^n. Conservation for momentum and energy are very tedious, but not complex. This is very much alike the conservation of momentum and energy with the forward Euler method. The result of this dull process is that, in order for momentum and energy to be conserved, u^n, T^n, $u^{(1)}$ & $T^{(1)}$ must satisfy

$$\begin{cases}
\mathcal{M}_h^n - u^n\rho_h^n + T^n\left(f_h^n(v_{\max}) - f_h^n(v_{\min})\right) = 0,\\
2\mathcal{E}_h^n - u^n\mathcal{M}_h^n - T^n\rho_h^n + T^n\left(f_h^n(v_{\max})v_{\max} - f_h^n(v_{\min})v_{\min}\right) = 0,
\end{cases} \qquad (2.14a)$$

$$\begin{cases}
\mathcal{M}_h^{(1)} - u^{(1)}\rho_h^{(1)} + T^{(1)}\left(f_h^{(1)}(v_{\max}) - f_h^{(1)}(v_{\min})\right) = 0,\\
2\mathcal{E}_h^{(1)} - u^{(1)}\mathcal{M}_h^{(1)} - T^{(1)}\rho_h^{(1)} + T^{(1)}\left(f_h^{(1)}(v_{\max})v_{\max} - f_h^{(1)}(v_{\min})v_{\min}\right) = 0.
\end{cases} \qquad (2.14b)$$

Now that the choice of bulk velocity and temperature are clear, a concrete algorithm can be created. Using the second-order strong-stability preserving Runge-Kutta method eventually turns (2.6) into

$$M\hat{\boldsymbol{f}}_h^{(1)} = M\hat{\boldsymbol{f}}_h^n + \Delta t\left(\boldsymbol{b}_f^n - (V - u^n P)\hat{\boldsymbol{f}}_h^n - \sqrt{T^n}P\hat{\boldsymbol{g}}_h^n\right), \tag{2.15a}$$

$$M\hat{\boldsymbol{g}}_h^n = \sqrt{T^n}\boldsymbol{b}_g^n - \sqrt{T^n}P\hat{\boldsymbol{f}}_h^n, \tag{2.15b}$$

$$M\hat{\boldsymbol{f}}_h^{n+1} = \frac{1}{2}M\left(\hat{\boldsymbol{f}}_h^n + \hat{\boldsymbol{f}}_h^{(1)}\right) + \frac{\Delta t}{2}\left(\boldsymbol{b}_f^{(1)} - (V - u^{(1)}P)\hat{\boldsymbol{f}}_h^{(1)} - \sqrt{T^{(1)}}P\hat{\boldsymbol{g}}_h^{(1)}\right), \tag{2.15c}$$

$$M\hat{\boldsymbol{g}}_h^{(1)} = \sqrt{T^n}\boldsymbol{b}_g^{(1)} - \sqrt{T^{(1)}}P\hat{\boldsymbol{g}}_h^{(1)}. \tag{2.15d}$$

The resulting algorithm for the Local Discontinuous Galerkin method using SSP-RK2 is given by Algorithm 2.

Algorithm 2 Local Galerkin method: strong stability-preserving Runge-Kutta

Input: $f(v,0)$

1: Build grids, test functions & matrices.
2: Compute f_h^0.
3: Compute $\rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$ with f_h^0. ▷ *using* (2.8)
4: Compute u^0 & T^0 with $f_h^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$. ▷ *using* (2.14a)
5: Compute g_h^0 with f_h^0 & T^0. ▷ *using* (2.15d)
6: **for** $n = 0$ to $N_t - 1$ **do**
7: Compute $f_h^{(1)}$ with $f_h^n,\ g_h^n,\ u^n$ & T^n. ▷ *using* (2.15a)
8: Compute $\rho_h^{(1)},\ \mathcal{M}_h^{(1)}$ & $\mathcal{E}_h^{(1)}$ with $f_h^{(1)}$. ▷ *using* (2.8)
9: Compute $u^{(1)}$ & $T^{(1)}$ with $f_h^{(1)},\ \rho_h^{(1)},\ \mathcal{M}_h^{(1)}$ & $\mathcal{E}_h^{(1)}$. ▷ *using* (2.14b)
10: Compute $g_h^{(1)}$ with $f_h^{(1)}$ & $T^{(1)}$. ▷ *using* (2.15d)
11: Compute f_h^{n+1} with $f_h^{(1)},\ g_h^{(1)},\ u^{(1)}$ & $T^{(1)}$. ▷ *using* (2.15c)
12: Compute $\rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$ with f_h^{n+1}. ▷ *using* (2.8)
13: Compute u^{n+1} & T^{n+1} with $f_h^{n+1},\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$. ▷ *using* (2.14a)
14: Compute g_h^{n+1} with f_h^{n+1} & T^{n+1}. ▷ *using* (2.15b)

2.3 Recovery Discontinuous Galerkin method

First introduced in Van Leer and Nomura (2005) and van Leer et al. (2007), the Recovery Discontinuous Galerkin (RDG) method represents a refinement of the Local Discontinuous Galerkin method. In contrast to LDG, which relies solely on local information for calculating numerical fluxes, RDG utilizes information from two adjacent elements to compute these fluxes. Specifically, the RDG method involves the recovery of a higher-order polynomial function over the domain of two connected elements, followed by the utilization of the value of this function at the shared boundary point as the numerical flux. This approach is aimed at enhancing the accuracy of the numerical method. For more detailed information, reference Hakim et al. (2020).

2.3.1 Weak formulation

As the LDG method, multiply (1.5) by test function $\varphi \in H^2(e_k)$ and integrate over the k^{th} element. Like last section, k will only be used when clarification is needed. The weak formulation results from integrating the drift term by parts once and the diffusion term twice. The diffusion term needs to be integrated by parts twice for conservation of momentum and energy to later hold for the consistent solution. Integrating by parts once will, regardless of numerical flux, when treating the conservation of discrete momentum, result in a term $\langle \partial_v f_h^n, 1\rangle_k$. As f_h^n is a time-discrete piecewise polynomial function over Ω with possible discontinuities, summing these inproducts will results in

$$\sum_k \langle \partial_v f_h^n, 1\rangle_k = \sum_k [f_h^n]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}} = [f_h^n]_{v_{\min}}^{v_{\max}} + \sum_{k=1}^{N-1} \left(\left(f_h^+\right)^n (v_{k+\frac{1}{2}}) - \left(f_h^-\right)^n (v_{k+\frac{1}{2}}) \right).$$

Similar results hold when treating conservation of energy. These added terms are absorbed into the bulk velocity and temperature causing them to no longer be consistent with the continuous Fokker-Planck equation (1.11). The LDG method also relies on integrating by parts twice, but it is less obvious due to the addition of g. The weak formulation after using integration by parts for a fixed time is then

Problem f: Find $f \in H^2(e_k)$ such that for all $\varphi \in H^2(e_k)$ one has

$$\langle \partial_t f, \varphi\rangle_k + \langle (v-u)f, \partial_v \varphi\rangle_k - T\langle f, \partial_{vv}\varphi\rangle_k = [(v-u)f\varphi + T\partial_v f\varphi - Tf\partial_v\varphi]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \quad (2.16)$$

where we impose the no-flux boundary conditions at the boundary elements,

$$(v-u)f + T\partial_v f = 0, \qquad \text{at } v = v_{\min}, v_{\max}.$$

2.3.2 Spatial discretization

Again approximate the solution to (2.16) by the projection onto the N_p-th-order piecewise polynomial space $\tilde{\mathbb{P}}(\Omega)$,

$$f(t,v) \approx f_h(t,v) = \bigoplus_{k=1}^{N} f_h^k(t,v), \text{ with } f_h^k(t,v) \in \mathbb{P}^{N_p}(e_k).$$

Next, approximate f over e_k as

$$f|_{e_k} \approx f_h^k(t,v) = \tilde{f}^k = \sum_{i=1}^{N_p+1} \hat{f}_i^k(t)\varphi_i^k(v), \quad \varphi_i^k \in \mathbb{P}^{N_p}(e_k) \tag{2.17}$$

Like with the LDG method, the RDG method is based on which numerical flux is used. For the drift term, we again employ the penalty form (2.3) used in the LDG method. On the other hand, the numerical flux resulting from the diffusion term will now be a higher order polynomial $\tilde{f}$ recovered over the elements $\tilde{e}_k$ which contain a grid point of choice, i.e., $v_{k+\frac{1}{2}}$. In this case $\tilde{e}_k = e_k \cup e_{k+1}$. In order to comprise a solvable system of equations later on, we choose the recovered polynomial to be of order $2N_p + 1$,

$$\tilde{f}|_{\tilde{e}_k} = \sum_{j=1}^{2N_p+2} \tilde{f}_j^k \psi_j^k(v), \tag{2.18}$$

where $\mathbb{P}^{2N_p+1}(\tilde{e}_k) = \text{span}_i(\psi_i)$. We recover this function $\tilde{f}$ by projecting f_h over the elements e_k and e_{k+1}. Using the same bases as before, φ_i, we get

$$\langle \tilde{f}^k, \varphi_i^k \rangle_k = \langle f_h^k, \varphi_i^k \rangle_k \quad \text{and} \quad \langle \tilde{f}^k, \varphi_i^{k+1} \rangle_{k+1} = \langle f_h^{k+1}, \varphi_i^{k+1} \rangle_{k+1}.$$

Define the matrices

$$L_{ij}^k = \langle \varphi_i^k, \psi_j^k \rangle_k \text{ and } R_{ij}^k = \langle \varphi_i^{k+1}, \psi_j^k \rangle_{k+1}, \quad \text{with } L^k, R^k \in \mathbb{R}^{N_p+1 \times 2N_p+2}.$$

Next, define

$$\tilde{R}^k = \begin{bmatrix} L^k \\ R^k \end{bmatrix}, \qquad \tilde{M}^k = \begin{bmatrix} M^k & 0 \\ 0 & M^k \end{bmatrix} \text{ and } \hat{\boldsymbol{f}}^{k+\frac{1}{2}} = \begin{bmatrix} \hat{\boldsymbol{f}}^k \\ \hat{\boldsymbol{f}}^{k+1} \end{bmatrix},$$

where $\hat{\boldsymbol{f}}^k$ denotes the column vector comprised of the coefficients of the projection of f onto $\mathbb{P}^{N_p}(e_k)$. Finding the recovered polynomial then comes down to solving linear system:

$$\tilde{R}^k \tilde{\boldsymbol{f}}^k = \tilde{M}^k \hat{\boldsymbol{f}}^{k+\frac{1}{2}}. \tag{2.19}$$

We can use the values of the recovered polynomials at the associated grid point to acquire the numerical fluxes. To avoid confusion, we will denote the value $\tilde{f}^k(v_{k+\frac{1}{2}})$ as $\tilde{f}(v_{k+\frac{1}{2}})$. For example, the functions for $\tilde{f}(v_{k+\frac{1}{2}})$ and $\tilde{f}(v_{k_{k-\frac{1}{2}}})$ are using different functions, but we do

not denote them as such. Also note that it is not possible to recover $\tilde{f}(v_{\min})$ and $\tilde{f}(v_{\max})$, as this is only part of one element. Like the LDG method, we substitute them by the value of f_h at the respective boundary. The weak formulation at a fixed time then turns into

Problem f_h: Find $f_h \in \mathbb{P}^{N_p}(e_k)$ such that for all $\varphi \in H^2(e_k)$ one has

$$\begin{aligned}&\langle \partial_t f_h, \varphi\rangle + \langle (v-u) f_h, \partial_v \varphi\rangle - T\langle f_h, \partial_{vv}\varphi\rangle \\ &\quad = \left[\left(\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right) + T\partial_v \tilde{f}_h\right)\varphi - T\tilde{f}_h \partial_v \varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}, \end{aligned} \tag{2.20}$$

where we impose a no-flux boundary conditions at the boundary elements,

$$(v-u)f_h + T\partial_v f_h = 0, \qquad \text{at } v = v_{\min}, v_{\max}.$$

Since all polynomials are in the Sobolev space, the test functions can be taken as our basis functions φ_i. This turns system (2.20) into a system of $N_p + 1$ equations. First, define the matrix

$$W_{ij}^k = \langle \partial_v \varphi_i, \partial_v \varphi_j\rangle_k, \qquad \text{with } W^k \in \mathbb{R}^{N_p+1 \times N_p+1}$$

and vector

$$\boldsymbol{b}_i = \left[\left(\frac{1}{2}\left((v-u)(f_h^+ + f_h^-) - \alpha(f_h^- - f_h^+)\right) + T\partial_v \tilde{f}_h\right)\varphi_i - T\tilde{f}_h \partial_v \varphi_i\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}},$$

with $\boldsymbol{b} \in \mathbb{R}^{N_p+1}$. Essentially, using the RDG method comes down to solving the linear equation

$$M\partial_t \hat{\boldsymbol{f}}_h + (V - uP - TW)\,\hat{\boldsymbol{f}}_h = \boldsymbol{b} \tag{2.21}$$

over all elements.

2.3.3 Discretization with respect to time

Again time is discretized into N_t equidistant time steps from $t = 0$ until $t = t_{\max}$. Denote the distance between time steps by Δt. Any time dependent function at time step t_n is

denoted with the superscript of n. We can again approximate the time derivative in two ways: the forward Euler method and the strong stability-preserving Runga-Kutta method. This fully discretizes the Fokker-Planck equation, making it possible to computationally solve the problem. Then, an analysis of the conservation properties of the discrete model is done.

Forward Euler

One can use the forward Euler method to approximate the time derivative of (2.20). Again both the bulk velocity and temperate will now change over every time step, so that the discrete weak formulation is given by

Problem RDG (Forward Euler): Find $f_h^{n+1} \in \mathbb{P}^{N_p}(e_k)$ such that for all $\varphi \in H^2(e_k)$ one has

$$\begin{aligned}&\langle f_h^{n+1}, \varphi\rangle + \Delta t\left(\langle (v-u^n) f_h^n, \partial_v \varphi\rangle - T^n \langle f_h^n, \partial_{vv}\varphi\rangle\right) = \langle f_h^n, \varphi\rangle \\ &+ \Delta t\left[\left(\frac{1}{2}\left((v-u^n)((f_h^n)^+ + (f_h^n)^-) - \alpha^n((f_h^n)^- - (f_h^n)^+)\right) + T\partial_v \tilde{f}_h^n\right)\varphi - T\tilde{f}_h^n \partial_v \varphi\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}},\end{aligned} \tag{2.22}$$

where we impose a no-flux boundary conditions at the boundary elements,

$$(v-u^n) f_h^n + T^n \partial_v f_h^n = 0, \quad \text{at } v = v_{\min}, v_{\max}. \tag{2.23}$$

Given an initial condition for f, the projection can be found onto $\tilde{\mathbb{P}}(\Omega)$ to get f_h^0. Then the solution can be found sequentially at all time steps. Note that (2.23) suggests the partial derivative with respect to f_h^n needs to be computed. However, this derivative only shows up at the boundary, for which we know no-flux boundary conditions hold. Therefore, it does not need to be explicitly computed. For this scheme, we once again want to check the conservation qualities of the discretizations. For the conservation of mass, we again sum (2.22) over all elements with $\varphi = 1$ and use the telescoping principle. Since the approximation has only one value at the boundary points, the Lax-Friederichs flux (2.3) takes the form of the no-flux boundary condition (2.23). It is trivial to now show that mass

is conserved:

$$\begin{aligned}\rho_h^{n+1} &= \sum_k \langle f_h^{n+1}, 1\rangle_k \\ &= \sum_k \int_{e_k} f_h^n + \Delta t\Big[(v-u^n)f_h^n + T^n\partial_v f_h^n\Big]_{v_{\min}}^{v_{\max}} \\ &= \rho_h^n.\end{aligned}$$

As for the conservation in momentum and energy, summing (2.22) over all elements with $\varphi = v$ and $\varphi = \frac{v^2}{2}$, gives us restrictions (2.24) on u^n and T^n such that momentum and energy is conserved:

$$\begin{cases}\mathcal{M}_h^n - u^n\rho_h^n + T^n\left(f_h^n(v_{\max}) - f_h^n(v_{\min})\right) = 0, \\ 2\mathcal{E}_h^n - u^n\mathcal{M}_h^n - T^n\rho_h^n + T^n\left(f_h^n(v_{\max})v_{\max} - f_h^n(v_{\min})v_{\min}\right) = 0.\end{cases} \tag{2.24}$$

This is the same conditions we have had up until now. Therefore, the results should stay consistent with the continuous choice for the bulk velocity and temperature (1.11). Now that the choice of bulk velocity and temperature are clear, a concrete algorithm can be created. Using the forward Euler method eventually turns (2.21) into system

$$M\hat{\boldsymbol{f}}_h^{n+1} = M\hat{\boldsymbol{f}}_h^n + \Delta t\left(\boldsymbol{b}^n - \left(V - u^nP - T^nW\right)\hat{\boldsymbol{f}}_h^n\right). \tag{2.25}$$

The resulting algorithm for the RDG method is given by Algorithm 3.

Algorithm 3 Recovery Galerkin method: forward Euler

Input: $f(v,0)$

1: Build grids, test functions & matrices.
2: Compute f_h^0.
3: Compute $\tilde{f}_h^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$ with f_h^0. ▷ *using* (2.19) *&* (2.8)
4: Compute u^0 & T^0 with $f_h^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$. ▷ *using* (2.24)
5: **for** $n = 0$ to $N_t - 1$ **do**
6: Compute f_h^{n+1} with $f_h^{n-1},\ \tilde{f}_h^n,\ u^n$ & T^n. ▷ *using* (2.25)
7: Compute $\tilde{f}_h^n,\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$ with f_h^{n+1}. ▷ *using* (2.19) *&* (2.8)
8: Compute u^{n+1} & T^{n+1} with $f_h^n,\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$. ▷ *using* (2.24)

Strong Stability-Preserving Runge-Kutta

Again, one can look at the SSP-RK2 method described in (2.11). In order to construct the RDG method using the SSP-RK2 method that conserves the three momenta, one could take the exact steps that have been done with the LDG method, except the diffusion term is once again integrated by parts twice. Imposing no-flux boundary conditions on f_h^n and $f_h^{(1)}$ guarantees conservation of mass. As for the conservation of momentum and energy, because of the integration by parts twice, it once again comes down to the same result. Thus, $u^{(1)},\ T^{(1)},\ u^n$ & T^n can be found such that

$$\begin{cases} \mathcal{M}_h^n - u^n \rho_h^n + T^n \left(f_h^n(v_{\max}) - f_h^n(v_{\min}) \right) = 0, \\ 2\mathcal{E}_h^n - u^n \mathcal{M}_h^n - T^n \rho_h^n + T^n \left(f_h^n(v_{\max}) v_{\max} - f_h^n(v_{\min}) v_{\min} \right) = 0, \end{cases} \tag{2.26a}$$

$$\begin{cases} \mathcal{M}_h^{(1)} - u^{(1)} \rho_h^{(1)} + T^{(1)} \left(f_h^{(1)}(v_{\max}) - f_h^{(1)}(v_{\min}) \right) = 0, \\ 2\mathcal{E}_h^{(1)} - u^{(1)} \mathcal{M}_h^{(1)} - T^{(1)} \rho_h^{(1)} + T^{(1)} \left(f_h^{(1)}(v_{\max}) v_{\max} - f_h^{(1)}(v_{\min}) v_{\min} \right) = 0. \end{cases} \tag{2.26b}$$

The linear system that needs to be computed over all elements can also be found to be

$$M \hat{\boldsymbol{f}}_h^{(1)} = M \hat{\boldsymbol{f}}_h^n + \Delta t \left(\boldsymbol{b}^n - \left(V - u^n P - T^n W \right) \hat{\boldsymbol{f}}_h^n \right), \tag{2.28a}$$

$$M \hat{\boldsymbol{f}}_h^{n+1} = \frac{1}{2} M \left(\hat{\boldsymbol{f}}_h^n + \hat{\boldsymbol{f}}_h^{(1)} \right) + \frac{\Delta t}{2} \left(\boldsymbol{b}^{(1)} - \left(V - u^{(1)} P - T^{(1)} W \right) \hat{\boldsymbol{f}}_h^{(1)} \right). \tag{2.28b}$$

The resulting algorithm for the Recovery Discontinuous Galerkin method using SSP-RK2 is given by Algorithm 4.

Algorithm 4 Recovery Galerkin method: strong stability-preserving Runge-Kutta

Input: $f(v,0)$

1: Build grids, test functions & matrices.
2: Compute f_h^0.
3: Compute $\tilde{f}^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$ with f_h^0. ▷ *using* (2.19) *&* (2.8)
4: Compute u^0 & T^0 with $f_h^0,\ \rho_h^0,\ \mathcal{M}_h^0$ & $\mathcal{E}_h^0$. ▷ *using* (2.26a)
5: **for** $n = 0$ to $N_t - 1$ **do**
6: Compute $f_h^{(1)}$ with $f_h^n,\ \tilde{f}^n,\ u^n$ & T^n. ▷ *using* (2.28a)
7: Compute $\tilde{f}_h^{(1)},\ \rho_h^{(1)},\ \mathcal{M}_h^{(1)}$ & $\mathcal{E}_h^{(1)}$ with $f_h^{(1)}$. ▷ *using* (2.19) *&* (2.8)
8: Compute $u^{(1)}$ & $T^{(1)}$ with $f_h^{(1)},\ \rho_h^{(1)},\ \mathcal{M}_h^{(1)}$ & $\mathcal{E}_h^{(1)}$. ▷ *using* (2.26b)
9: Compute f_h^{n+1} with $f_h^{(1)},\ \tilde{f}^{(1)},\ u^{(1)}$ & $T^{(1)}$. ▷ *using* (2.28b)
10: Compute $\tilde{f}_h^{n+1},\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$ with f_h^{n+1}. ▷ *using* (2.19) *&* (2.8)
11: Compute u^{n+1} & T^{n+1} with $f_h^{n+1},\ \rho_h^{n+1},\ \mathcal{M}_h^{n+1}$ & $\mathcal{E}_h^{n+1}$. ▷ *using* (2.26a)

2.4 On stability

While these methods may appear theoretically sound and computationally efficient, the crucial question remains: Do they truly converge to accurate solutions? As the original kinetic Fokker-Planck equation is a non-constant coefficient linear partial differential equation, validating the convergence of numerical methods involves proving their consistency and stability. However, we found that, for a bounded domain, the bulk velocity and temperature depend on the boundary value of our solution. This complicates this matter, yet, consistency and stability remain a crucial part of convergence. A method's consistency ensures that it accurately approximates the solution after a time step, while stability guarantees that this approximation does not explode over time. For consistency, both the LDG and RDG method are $N_p + 1$th-order-accurate methods. Literature on this can be found in Hesthaven and Warburton (2007). This section, on the other hand, only addresses the essential stability aspect of the LDG and RDG method. It is well-known that advection equations are stable when $\Delta t = \mathcal{O}(h)$ and diffusion equations when $\Delta t = \mathcal{O}(h^2)$, where h is the spacing of the cells, or $h = |\Omega|/N$. Therefore, a constant coefficient advection-diffusion equation should be

a combination of the two. In most cases, this means it will be stable when $\Delta t = \mathcal{O}(h^2)$, but when the diffusion has less influence, the advection term dominates stability. On the other hand, stability will also depend on the order of polynomial with which we approximate. In Hesthaven and Warburton (2007), it is found that the LDG method requires $\Delta t = \mathcal{O}\left(\frac{h}{N_p^2}\right)$ on an advection term. Therefore, we suspect the results of stability to be $\Delta t = \mathcal{O}\left(\frac{h^2}{N_p^4}\right)$ on a diffusion term. These results hold for constant coefficient advection equations. However, the Fokker-Planck equation also has a drift term. The Courant-Friederichs-Lewy (CFL) condition then becomes something like $\Delta t \leq \frac{h}{\max_{v\in\Omega}|v-u|}$. Therefore, the stability also depends on the size of the domain. Moreover, our bulk velocity and temperature for bounded domains depend on the value of the approximation at the boundary. This results in a non-linear equation, making it even more complicated. For small domains, the influence of the boundary on the bulk velocity and temperature increases, thus small domains are tricky to deal with. The stability analysis is a complex matter for the LDG method, let alone the RDG method. This is why we only give a way to compute whether the schemes are stable or not.

Since the SSP-RK2 method is designed to preserve the stability of the forward Euler method, we only look at the stability of the latter. In order to prove stability, it will be imperative to specify both the element and time step dependence, recall the notation $\boldsymbol{f}_h^{k,n}$ that denotes the coefficients of the discretized function $\boldsymbol{f}_h$ over element k at time step n. If no element is specified, the whole domain should now be considered. Proving stability will involve transforming relevant equations to the form

$$\hat{\boldsymbol{f}}_h^{n+1} = I + \Delta t \boldsymbol{A}^n \hat{\boldsymbol{f}}_h^n, \tag{2.29}$$

which solves the systems over all N elements at once. Therefore, using the notation of the coefficients to describe the Galerkin method (2.2) and (2.17), the column vector $\hat{\boldsymbol{f}}_h^n \in \mathbb{R}^{(N_p+1)N}$ is described by

$$\hat{\boldsymbol{f}}_h^n = \left[\hat{f}_1^{1,n} \quad \dots \quad \hat{f}_{N_p+1}^{1,n}, \quad \hat{f}_1^{2,n} \quad \dots \quad \hat{f}_{N_p+1}^{k-1,n}, \quad \hat{f}_1^{k,n} \quad \dots \quad \hat{f}_{N_p+1}^{N,n}\right]^T.$$

For stability to now hold, the spectral radius of matrix $I + \Delta t \boldsymbol{A}^n$ should be less than or equal to one for all time steps n. In other words, the eigenvalues of $\Delta t \boldsymbol{A}^n$ should be inside

the circle with center -1 and radius 1 on the complex plane. Ideally, Δt can be taken small enough such that this is achieved. The existence of such Δt then comes down to if the eigenvalues of A^n only have negative real components. Theoretically proving this, is outside the scope of this thesis. Note that any matrix involved in the computation of $\boldsymbol{A}^n$ will be denoted with a bold font.

2.4.1 Stability of the LDG method

Recall equations (2.10a) and (2.10b) associated with the forward Euler method and the LDG method over element e_k,

$$M^k \hat{\boldsymbol{f}}_h^{k,n+1} = M^k \hat{\boldsymbol{f}}_h^{k,n} + \Delta t \left(\boldsymbol{b}_f^{k,n} - (V^k - u^n P^k) \hat{\boldsymbol{f}}_h^{k,n} - \sqrt{T^n} P^k \hat{\boldsymbol{g}}_h^{k,n} \right), \tag{2.30}$$

$$M^k \hat{\boldsymbol{g}}_h^{k,n} = \sqrt{T^n} \boldsymbol{b}_g^{k,n} - \sqrt{T^n} P^k \hat{\boldsymbol{f}}_h^{k,n}. \tag{2.31}$$

Since interaction between elements only happens in $\boldsymbol{b}_f^{k,n}$ and $\boldsymbol{b}_g^{k,n}$, we know that all other associated bold matrices will be block matrices with potential non-zero entries on the diagonal blocks. Consequently, $\boldsymbol{M}$ is defined as

$$\boldsymbol{M} = \begin{bmatrix} M^1 & \mathbf{0} & & & \\ \mathbf{0} & M^2 & \mathbf{0} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{0} & M^{N-1} & \mathbf{0} \\ & & & \mathbf{0} & M^N \end{bmatrix}, \quad \boldsymbol{M} \in \mathbb{R}^{N(N_p+1)\times N(N_p+1)}. \tag{2.32}$$

Matrices $\boldsymbol{P}$ and $\boldsymbol{V}$ are defined in a similar manner. We now seek a way to correlate $\boldsymbol{b}_f^{k,n}$ and $\boldsymbol{b}_g^{k,n}$ to the coefficients. To achieve this, recall

$$(\boldsymbol{b}_f^{k,n})_i = \left[\frac{1}{2} \left((v-u^n)(f_h^+ + f_h^-) - \alpha^n (f_h^- - f_h^+) \right) \varphi_i^k + \sqrt{T^n} \left(g_h^n \right)^+ \varphi_i^k \right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}$$

$$(\boldsymbol{b}_g^{k,n})_i = \left[\left(f_h^n \right)^- \varphi_i^k \right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}.$$

Also, recall that $\alpha^n = \max_{v\in\Omega}(|v-u^n|)$. Instead of the coefficients of the basis, this depends on the value of the functions at the grid points. Therefore, a transformation matrix from the coefficients to functions values is necessary. The Galerkin method (2.2) gives us this

transformation matrix. Suppose we divide an element k into grid $v_{k-\frac{1}{2}} = \tilde{v}_1^k < \tilde{v}_2^k < \cdots < \tilde{v}_{N_p+1}^k = v_{k+\frac{1}{2}}$, which was already necessary for Gauss quadrature. The transition matrix over an element k is then given by

$$\Phi_{ij}^k = \varphi_j(\tilde{v}_i^k).$$

Since there is no interaction between elements, we define $\boldsymbol{\Phi}$ similarly to $\boldsymbol{M}$ (2.32). We now seek to find matrices $\boldsymbol{B}_{\text{flux}}$ and $\boldsymbol{B}_g$ such that

$$\boldsymbol{b}_f^n = \boldsymbol{B}_{\text{flux}}^n \boldsymbol{\Phi} \hat{\boldsymbol{f}}^n, \quad \text{and} \quad \boldsymbol{b}_g^n = \boldsymbol{B}_g \boldsymbol{\Phi} \hat{\boldsymbol{f}}^n.$$

Note that $\boldsymbol{B}_{\text{flux}}$ describes the flux. To grasp how to do this, we look at finding the matrix associated with $\mathbf{b}_g^n$. Observe that following matrix-vector multiplication describes $\mathbf{b}_g^n$ everywhere but the first element

$$\left[\begin{array}{cc|ccc|c} \ddots & \vdots & & & & \iddots \\ 0 & \varphi_{N_p+1}^{k-1}(v_{(k-1)+\frac{1}{2}}) & 0 & 0 & 0 & 0 \\ \hline 0 & -\varphi_1^k(v_{k-\frac{1}{2}}) & 0 & \dots 0 & \varphi_1^k(v_{k+\frac{1}{2}}) & 0 \\ & \vdots & & & \vdots & \\ 0 & -\varphi_{N_p+1}^k(v_{k-\frac{1}{2}}) & 0 & 0 & \varphi_{N_p+1}^k(v_{k+\frac{1}{2}}) & 0 \\ \hline 0 & 0 & 0 & \dots 0 & -\varphi_1^{k+1}(v_{(k+1)-\frac{1}{2}}) & 0 \\ \iddots & & & & \vdots & \ddots \end{array}\right] \left[\begin{array}{c} \vdots \\ (f_h^n)^-(v_{k-\frac{1}{2}}) \\ \hline (f_h^n)^+(v_{k-\frac{1}{2}}) \\ \vdots \\ (f_h^n)^-(v_{k+\frac{1}{2}}) \\ \hline (f_h^n)^-(v_{k+\frac{1}{2}}) \\ \vdots \end{array}\right].$$

To this extent, define the following matrices

$$\left(B_{k\pm\frac{1}{2}}^-\right)_{ij} = \begin{cases} \varphi_i^k\left(v_{k\pm\frac{1}{2}}\right) & j = N_p+1, \\ 0 & \text{otherwise,} \end{cases} \text{ and } \left(B_{k\pm\frac{1}{2}}^+\right)_{ij} = \begin{cases} \varphi_i^k\left(v_{k\pm\frac{1}{2}}\right) & j = 1, \\ 0 & \text{otherwise,} \end{cases}$$

with $B_{k\pm\frac{1}{2}}^-, B_{k\pm\frac{1}{2}}^+ \in \mathbb{R}^{N_p+1\times N_p+1}$. It is imperative that we do not simplify the formulation $k\pm\frac{1}{2}$. The notation indicates both element k and the gridpoint, for example, subscript $k+\frac{1}{2}$ indicates that we approach gridpoint $v_{k+\frac{1}{2}}$ from the kth element, while subscript $k+1-\frac{1}{2}$

indicates that we approach gridpoint $v_{k+\frac{1}{2}}$ from the $k+1$th element. As we choose to have one value at the boundary, we can now define matrix $\boldsymbol{B}_g$:

$$\boldsymbol{B}_g = \begin{bmatrix} -B^+_{1-\frac{1}{2}} + B^-_{1+\frac{1}{2}} & \mathbf{0} & & & & \\ -B^-_{2-\frac{1}{2}} & B^-_{2+\frac{1}{2}} & \mathbf{0} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -B^-_{N-1-\frac{1}{2}} & B^-_{N-1+\frac{1}{2}} & \mathbf{0} & \\ & & & -B^-_{N-\frac{1}{2}} & B^-_{N+\frac{1}{2}} \end{bmatrix}.$$

To find $\boldsymbol{B}_{\text{flux}}$, we first rewrite the penalty form into a more approachable form:

$$\left[\frac{1}{2}\left((v-u^n+\alpha^n)\left(f_h^n\right)^+ + (v-u^n-\alpha^n)\left(f_h^n\right)^-\right)\varphi_i\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}.$$

Now define

$$\left(B^n_{-,k\pm\frac{1}{2}}\right)_{ij} = \begin{cases} \frac{1}{2}\left(v_{k\pm\frac{1}{2}} - u^n - \alpha^n\right)\varphi_i^k(v_{k\pm\frac{1}{2}}) & j = N_p+1, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\left(B^n_{+,k\pm\frac{1}{2}}\right)_{ij} = \begin{cases} \frac{1}{2}\left(v_{k\pm\frac{1}{2}} - u^n + \alpha^n\right)\varphi_i^k(v_{k\pm\frac{1}{2}}) & j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

with $B^n_{-,k\pm\frac{1}{2}}, B^n_{+,k\pm\frac{1}{2}} \in \mathbb{R}^{N_p+1\times N_p+1}$. Using these and putting the matrix at the boundary to zero, to impose the no-flux boundary condition, we can find matrix $\boldsymbol{B}_{\text{LF}}$ as

$$\begin{bmatrix} B^n_{-,1+\frac{1}{2}} & B^n_{+,1+\frac{1}{2}} & & & & \\ -B^n_{-,2-\frac{1}{2}} & -B^n_{+,2-\frac{1}{2}} + B^n_{-,2+\frac{1}{2}} & B^n_{+,2+\frac{1}{2}} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -B^n_{-,N-1-\frac{1}{2}} & -B^n_{+,N-1-\frac{1}{2}} + B^n_{-,N-1+\frac{1}{2}} & B^n_{+,N-1+\frac{1}{2}} \\ & & & -B^n_{-,N-\frac{1}{2}} & -B^n_{+,N-\frac{1}{2}} \end{bmatrix} \tag{2.33}$$

Now, we must realize that (2.31) gives a relation between $\hat{\boldsymbol{g}}_h^n$ and $\hat{\boldsymbol{f}}_h^n$:

$$\hat{\boldsymbol{g}}_h^n = \sqrt{T^n}\boldsymbol{M}^{-1}\left(\boldsymbol{B}_g\boldsymbol{\Phi} - \boldsymbol{P}\right)\hat{\boldsymbol{f}}_h^n. \tag{2.34}$$

Then define another matrix that can take g from the right, while imposing no-flux boundary conditions, as

$$\boldsymbol{B}^{+} = \begin{bmatrix} \mathbf{0} & B^{+}_{1+\frac{1}{2}} & & & \\ \mathbf{0} & -B^{+}_{2-\frac{1}{2}} & B^{+}_{2+\frac{1}{2}} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{0} & -B^{+}_{(N-1)-\frac{1}{2}} & B^{+}_{(N-1)+\frac{1}{2}} \\ & & & \mathbf{0} & -B^{+}_{N-\frac{1}{2}} \end{bmatrix}.$$

This now gives us a way to find $\boldsymbol{B}_{\text{flux}}$:

$$\boldsymbol{B}^{n}_{\text{flux}} = \boldsymbol{B}^{n}_{\text{LF}} + T^{n}\boldsymbol{B}^{+}\boldsymbol{\Phi}\boldsymbol{M}^{-1}\left(\boldsymbol{B}_{g}\boldsymbol{\Phi} - \boldsymbol{P}\right)\boldsymbol{\Phi}^{-1}.$$

Using these matrices and the translation from $\hat{\boldsymbol{g}}^{n}$ to $\hat{\boldsymbol{f}}^{n}$ (2.34) we can transform the LDG method into the form (2.29) from which we can derive stability results. Ultimately, we have that

$$\boldsymbol{A}^{n} = \boldsymbol{M}^{-1}\left(\boldsymbol{B}^{n}_{\text{flux}}\boldsymbol{\Phi} - (\boldsymbol{V} - u^{n}\boldsymbol{P}) - T^{n}\boldsymbol{P}\boldsymbol{M}^{-1}\left(\boldsymbol{B}_{g}\boldsymbol{\Phi} - \boldsymbol{P}\right)\right). \tag{2.35}$$

2.4.2 Stability of the RDG method

Recall equation (2.25) associated with the forward Euler method and the RDG method:

$$M^{k}\hat{\boldsymbol{f}}^{k,n+1}_{h} = M^{k}\hat{\boldsymbol{f}}^{k,n}_{h} + \Delta t\left(\boldsymbol{b}^{k,n} - \left(V^{k} - u^{n}P^{k} - T^{n}W^{k}\right)\hat{\boldsymbol{f}}^{k,n}_{h}\right).$$

Since interaction between elements only happens in $\boldsymbol{b}^{k,n}$, we know that all other associated bold matrices will be block matrices with potential non-zero entries on the diagonal blocks. Therefore, we use the same matrices as the LDG method and define $\boldsymbol{W}$ similarly to $\boldsymbol{M}$ (2.32). As for the vector containing the numerical flux components, recall that,

$$\boldsymbol{b}^{k,n}_{i} = \left[\left(\frac{1}{2}\left((v-u^{n})((f^{n}_{h})^{+} + (f^{n}_{h})^{-}\right) - \alpha^{n}((f^{n}_{h})^{-} - (f^{n}_{h})^{+})\right) + T^{n}\partial_{v}\tilde{f}^{n}_{h}\right)\varphi^{k}_{i} - T^{n}\tilde{f}^{n}_{h}\partial_{v}\varphi^{k}_{i}\right]_{v_{k-\frac{1}{2}}}^{v_{k+\frac{1}{2}}}.$$

Here there are two major components: the Lax-Friederichs method and the recovered polynomial. We aim to find matrices $\boldsymbol{B}^{n}_{\text{flux}}$ and $\boldsymbol{B}^{n}_{\text{rec}}$, such that

$$\boldsymbol{b}^{n} = (\boldsymbol{B}^{n}_{\text{flux}}\boldsymbol{\Phi} - \boldsymbol{B}^{n}_{\text{rec}})\hat{\boldsymbol{f}}^{n}_{h}.$$

Again $\boldsymbol{B}^n_{\text{flux}}$ describes the flux, thus in this case the Lax-Friederichs flux and the derivative of the recovered polynomial. For this Lax-Friederichs flux, we can use the same matrix $\boldsymbol{B}_{\text{LF}}$ (2.33) as the LDG method. In order to compute the matrices related to the recovery polynomial we recall the definition of the recovered polynomial over element $\tilde{e}_k = e_k \cup e_{k+1}$ (2.18):

$$\tilde{f}|_{\tilde{e}_k} = \sum_{i=1}^{2N_p+2} \tilde{f}_i^k \psi_i^k(v),$$

and the way the coefficients $\tilde{f}_i^k$ are computed (2.19):

$$\tilde{R}^k \tilde{\boldsymbol{f}}^k = \tilde{M}^k \hat{\boldsymbol{f}}^{k+\frac{1}{2}}.$$

As we are only interested in the value of $\tilde{f}(v_{k+\frac{1}{2}})$, we can define row vector

$$\boldsymbol{\psi}^{k+\frac{1}{2}} = \left[\psi_1^k((v_{k+\frac{1}{2}})) \dots \psi_{2N_p+2}^k((v_{k+\frac{1}{2}}))\right].$$

From (2.19), we see that $\boldsymbol{\psi}^k \tilde{\boldsymbol{f}}^k = \tilde{f}(v_{k+\frac{1}{2}})$, therefore:

$$\tilde{f}(v_{k+\frac{1}{2}}) = \boldsymbol{\psi}^k \left(\tilde{R}^k\right)^{-1} \tilde{M}^k \hat{\boldsymbol{f}}^{k+\frac{1}{2}}.$$

We can now define

$$\left(\Psi^k_{l\pm\frac{1}{2}}\right)_{ij} = \begin{cases} \left(\boldsymbol{\psi}^k \left(\tilde{R}^k\right)^{-1} \tilde{M}^k \partial_v \varphi_i^k(v_{l\pm\frac{1}{2}})\right)_j & j = 1, \dots, 2N_p + 2 \\ 0 & \text{otherwise,} \end{cases}$$

$$\left(DB^+_{\min}\right)_{ij} = \begin{cases} \partial_v \varphi_i^1(v_{\min}) & j = 1, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \left(DB^-_{\max}\right)_{ij} = \begin{cases} \partial_v \varphi_i^N(v_{\max}) & j = N_p + 1, \\ 0 & \text{otherwise.} \end{cases}$$

It is important to note that $\Psi^k_{l\pm\frac{1}{2}} \in \mathbb{R}^{N_p+1 \times 2N_p+2}$ and $DB^-_{k\pm\frac{1}{2}}, DB^+_{k\pm\frac{1}{2}} \in \mathbb{R}^{N_p+1 \times N_p+1}$. The l in this notation also denotes an element here. As k denotes a pair of elements over which we construct the recovered polynomial, a second notation for elements is necessary. Because of this, the boxes in the following matrix represent where they should be elementwise added or subtracted. Recall that we use the function value instead of the recovery polynomial at the boundary; we then have

$$\boldsymbol{\Psi} = \begin{bmatrix} -DB^{+}_{\min}\Phi^{1} & \Psi^{1}_{1+\frac{1}{2}} & \mathbf{0} & & \\ -\Psi^{1}_{2-\frac{1}{2}} & & \Psi^{2}_{2+\frac{1}{2}} & \mathbf{0} & \\ & \ddots & \ddots & \ddots & \\ & & -\Psi^{N-2}_{N-1-\frac{1}{2}} & & \Psi^{N-1}_{N-1+\frac{1}{2}} \\ & & \mathbf{0} & -\Psi^{N-1}_{N-\frac{1}{2}} & DB^{-}_{\max}\Phi^{N} \end{bmatrix}.$$

Similarly, one can find $\partial_v \tilde{f}(v_{k+\frac{1}{2}})$. Define

$$\boldsymbol{d\psi}^{k+\frac{1}{2}} = \left[\partial_v \psi^k_1((v_{k+\frac{1}{2}})) \dots \partial_v \psi^k_{2N_p+2}((v_{k+\frac{1}{2}}))\right],$$

and

$$\boldsymbol{d\psi}^{k+\frac{1}{2}} = \left[\partial_v \psi^k_1((v_{k+\frac{1}{2}})) \dots \partial_v \psi^k_{2N_p+2}((v_{k+\frac{1}{2}}))\right].$$

From deriving (2.18), we see that

$$\partial_v \tilde{f}(v_{k+\frac{1}{2}}) = \boldsymbol{d\psi}^k \left(\tilde{R}^k\right)^{-1} \tilde{M}^k \hat{\boldsymbol{f}}^{k+\frac{1}{2}}.$$

We can now define

$$\left(D\Psi^k_{l\pm\frac{1}{2}}\right)_{ij} = \begin{cases} \left(\boldsymbol{d\psi}^k \left(\tilde{R}^k\right)^{-1} \tilde{M}^k \varphi^k_i(v_{l+\frac{1}{2}})\right)_j & j = 1, \dots, 2N_p+2, \\ 0 & \text{otherwise.} \end{cases}$$

It is again important to note that $D\Psi^k_{l\pm\frac{1}{2}} \in \mathbb{R}^{N_p+1\times 2N_p+2}$. The l in this notation also denotes an element here. As k denotes a pair of elements over which we construct the recovered polynomial, a second notation for elements is necessary. Therefore, we use the same notation as before. In order to impose the no-flux boundary condition, we again choose the zero matrix for the boundary. We then have

$$\boldsymbol{D\Psi} = \begin{bmatrix} \mathbf{0} & D\Psi^1_{1+\frac{1}{2}} & \mathbf{0} & & & \\ -D\Psi^1_{2-\frac{1}{2}} & & D\Psi^2_{2+\frac{1}{2}} & \mathbf{0} & & \\ & \ddots & \ddots & \ddots & & \\ & & -D\Psi^{N-2}_{N-1-\frac{1}{2}} & & D\Psi^{N-1}_{N-1+\frac{1}{2}} & \\ & & \mathbf{0} & -D\Psi^{N-1}_{N-\frac{1}{2}} & \mathbf{0} & \end{bmatrix}.$$

We can now compute both $\boldsymbol{B}^n_{\text{flux}}$ and $\boldsymbol{B}^n_{\text{rec}}$:

$$\boldsymbol{B}^n_{\text{flux}} = \boldsymbol{B}^n_{\text{LF}} + T^n \boldsymbol{D\Psi\Phi}^{-1} \text{ and } \boldsymbol{B}^n_{\text{rec}} = T^n \boldsymbol{\Psi}.$$

Using these matrices, we can translate the RDG method into the form (2.29) with matrix

$$\boldsymbol{A}^n = \boldsymbol{B}^n_{\text{flux}} \boldsymbol{\Phi} - \boldsymbol{B}^n_{\text{rec}} - \boldsymbol{V} + u^n \boldsymbol{P} + T^n \boldsymbol{W}. \tag{2.36}$$

Chapter 3

RESULTS AND SIMULATIONS

In order to discuss how well the four algorithms perform, they must be implemented. This is done in MATLAB. The utilization of Legendre polynomials as test functions forms the foundation of this chapter. Then the integration processes described in Chapter 2 can be approximated utilizing Gauss-Legendre quadrature in conjunction with the associated Legendre-Gauss-Lobatto grid and weights. With a sufficient number of grid points, it is expected that almost all integrals can be calculated precisely, with negligible rounding errors, as they exclusively involve polynomials. This characteristic is of paramount importance for the preservation of conservation, as it ensures the exactness of the integrals. The only integrals not being computed exactly are those when the projection of the initial condition happens. This has no effect on conservation, as we conserve the discrete momenta which already have the results of this error in them.

The obtained results are organized into three distinct sections: the conservation and entropy properties, stability, and convergence. In order to facilitate a comprehensive comparison, the LDG and RDG methods are applied to both relatively large and relatively small bounded domains. Throughout this chapter, a small bounded domain refers to a domain where the solution is non-zero at the boundary. A large bound then refers to a domain where the solution is zero at the boundary for all times.

3.1 Conservation & entropy properties

First, the primary motivation for this thesis: the conservation properties. Each method was first tested over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$, employing piecewise first, second and third-order polynomial approximations with sufficiently small temporal spacing. In the context of conservation, our goal was to maintain constant momenta throughout all time steps. However, given the nature of numerical computations,

rounding errors inevitably come into play. Therefore, when discussing conservation, we refer to it in terms of conservation up to machine precision.

Initially, both algorithms are applied to a case involving a large domain and the same smooth initial conditions. In addition to assessing the conservation properties, we also examined how the bulk velocity, temperature and Gibbs entropy evolved over time. During most simulations, the approximation was not positive, negative values always occurred when the approximations were close to zero. In order to compute the entropy, we neglect these negative values. The entropy purely serves to show the asymptotic behavior is reached, as our methods do not try to preserve any properties regarding it. First, the pitfall of both the LDG and RDG methods are observed: conservation of energy should not hold for piecewise linear functions. Figure 3.1 presents the results obtained from the LDG method over a large domain using linear approximations and the sum of two Gaussians as an initial condition, while 3.2 showcases the results from the RDG method.

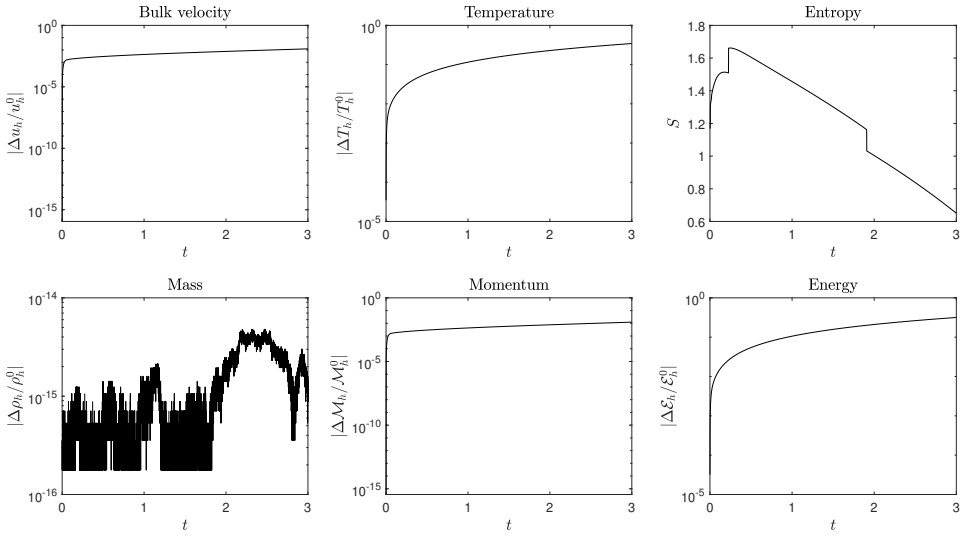

Fig. 3.1: Using the LDG method with piecewise linear approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing, conservation of mass holds, but conservation of momentum and energy do not. This causes the bulk velocity and temperature to change over time, even though the function is zero at the boundary.

The conservation of mass holds, which is to be expected from no-flux boundary conditions.

Surprisingly the conservation of momentum is not preserved. Conservation of momentum should hold since only first-order polynomials are required, while the energy should not be conserved. This lack of conservation causes the bulk velocity and temperature to change over time. Also note that both the entropy and the rising parameters indicate that the solution keeps changing over time, therefore asympotic behavior might not exist for these methods with piecewise first-order polynomials.

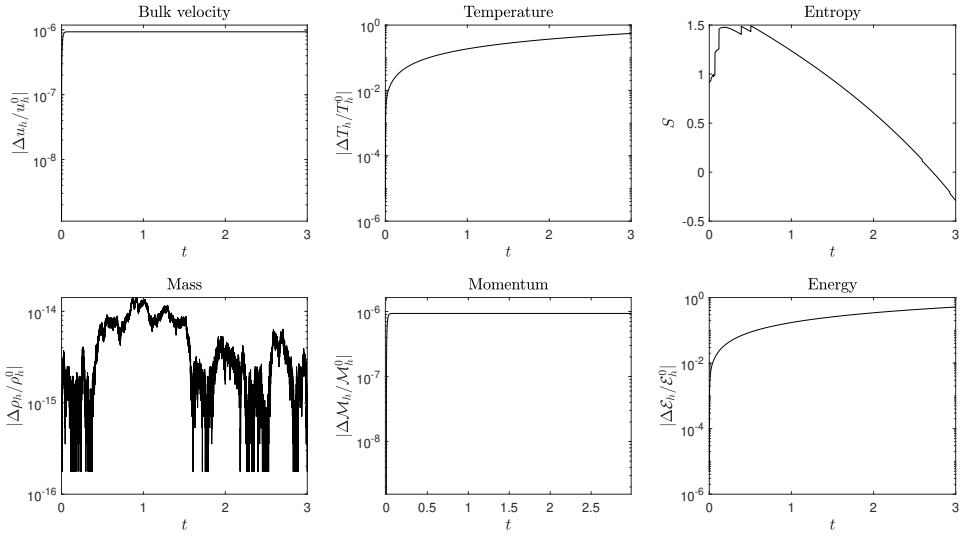

Fig. 3.2: Using the RDG method with piecewise linear approximations over 24 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing and conservation of mass holds but conservation of momentum and energy does not. This causes the bulk velocity and temperature to change over time, even though the function is zero at the boundary.

These results do not fall in line with the theoretical results. However, through experimentation, we found that both methods do conserve momentum once the spatial grid is refined. The new findings are displayed in Figure 3.3 (LDG) and Figure 3.4 (RDG).

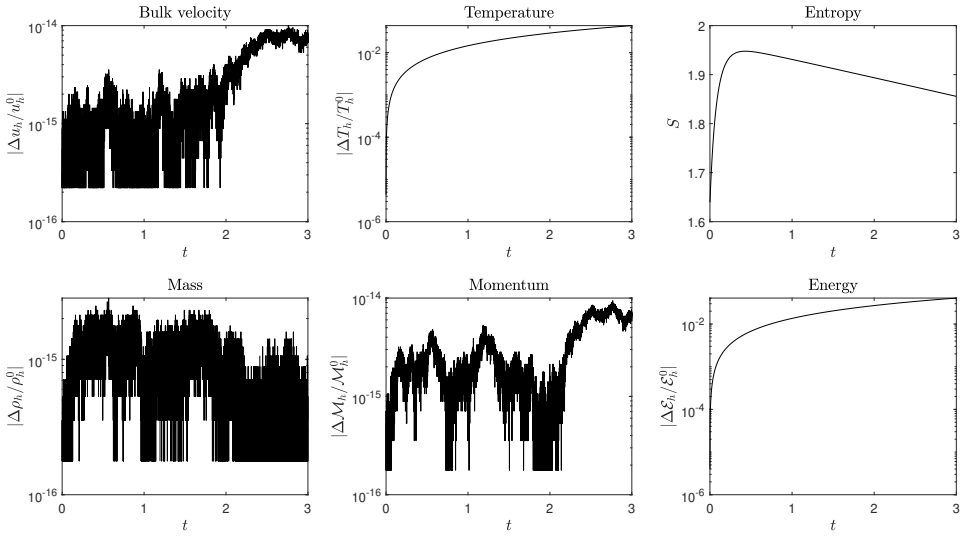

Fig. 3.3: Using the LDG method with piecewise linear approximations over 80 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing and conservation of mass and momentum holds but conservation of energy does not. This sudden change of conservation of momentum is due to the grid refinement.

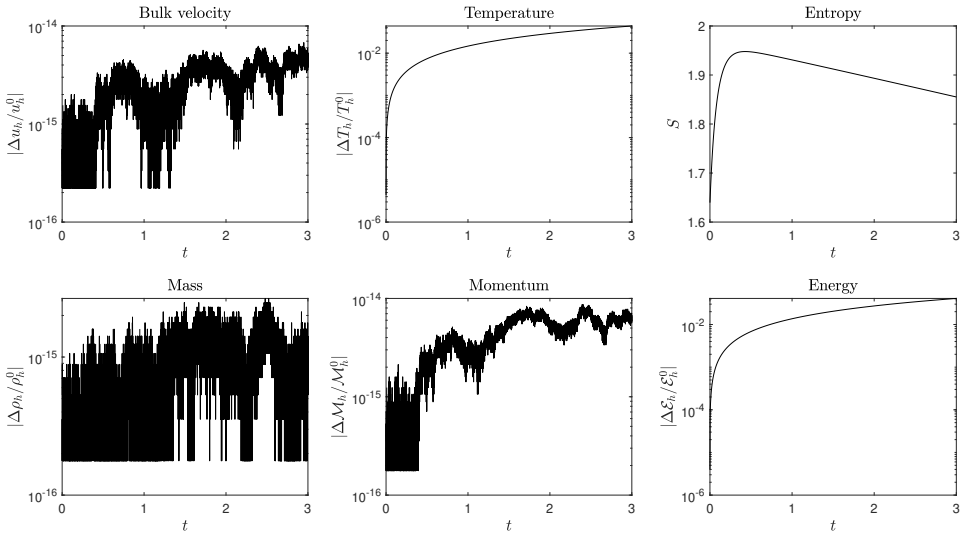

Fig. 3.4: Using the RDG method with piecewise linear approximations over 80 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing and conservation of mass and momentum holds but conservation of energy does not. This sudden change of conservation of momentum is due to the grid refinement.

This raises the question: do piecewise second-order polynomials conserve energy for both methods on a coarse grid? Figure 3.5 and Figure 3.6 show that they do not. Even when the grid is refined, the conservation property does not hold. This result is unexpected, as the proof of conservation only requires the finite-dimensional function space to be at least spanned by second-order polynomials.

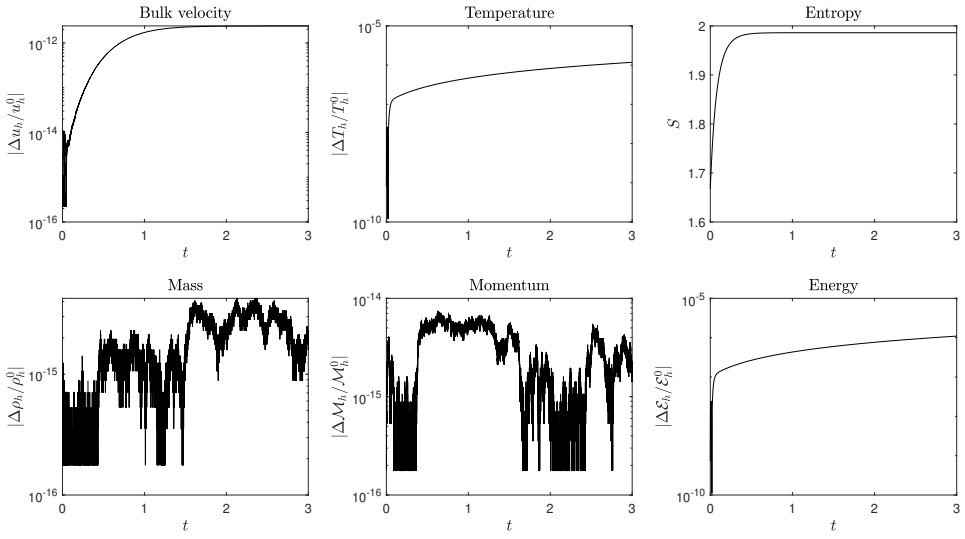

Fig. 3.5: Using the LDG method with piecewise second-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing, conservation of mass and momentum holds, but conservation of energy does not. Refining the grid does not conserve energy, which does not fall in line with theoretical results.

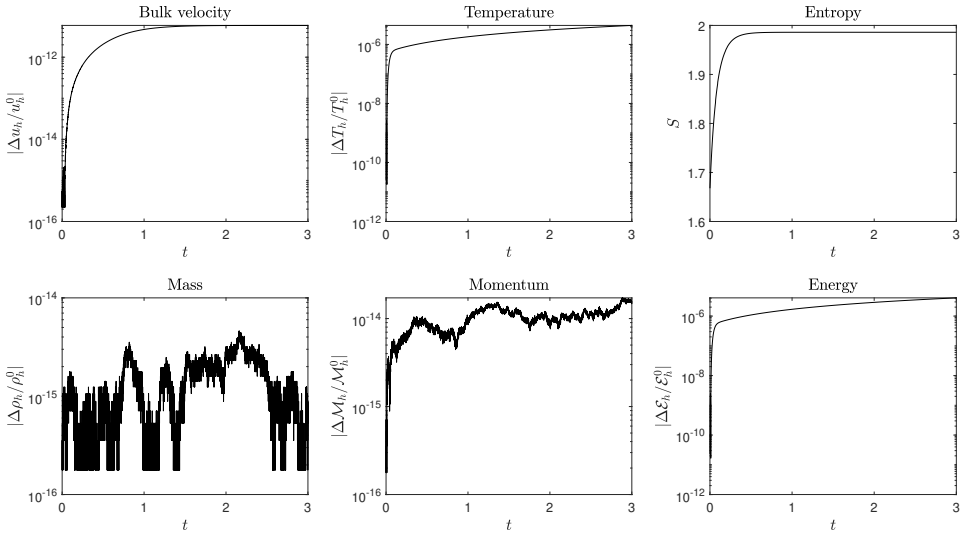

Fig. 3.6: Using the RDG method with piecewise second-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$, the entropy and the changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. Entropy is not strictly increasing, conservation of mass and momentum holds, but conservation of energy does not. Refining the grid does not conserve energy, which does not fall in line with theoretical results.

Now, higher order polynomials are involved. For brevity we only consider third-order polynomials over 24 elements with a small enough time step; nevertheless, the same behaviors are seen for higher order polynomials. Figure 3.7 presents the results obtained from the LDG method for smooth initial conditions and large domains, while Figure 3.8 showcases those of the RDG method.

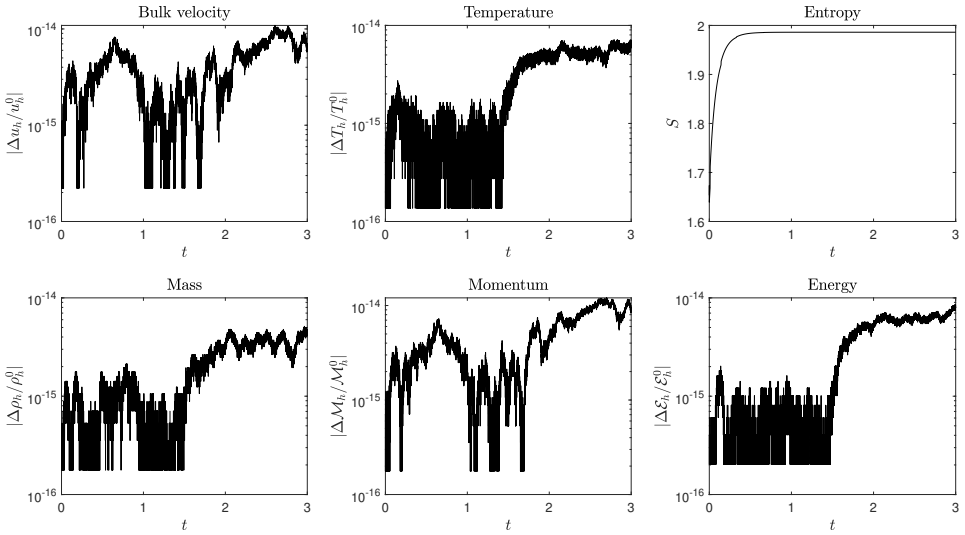

Fig. 3.7: Using the LDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. The bulk velocity and temperature are nearly constant, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision. The entropy also flattens, indicating it reaches its asymptotic behavior.

These results fall exactly in line with the theoretical results: the conservation properties hold. The bulk velocity and temperature are not exactly constant, but also do not change significantly over time. Note that refining the grid results in the expected constant behavior. Entropy seems to flatten, indicating asymptotic behavior is reached. The RDG method gives the same results.

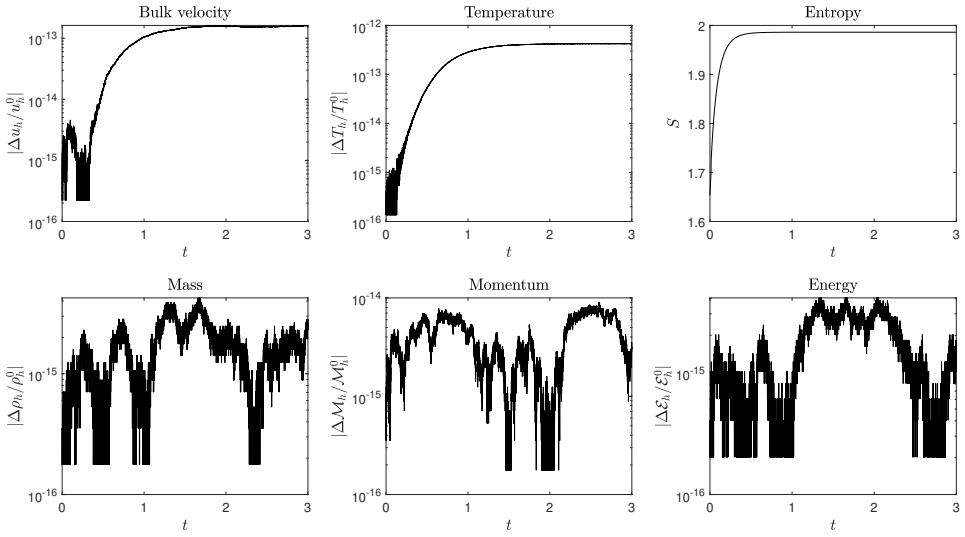

Fig. 3.8: Using the RDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A smooth initial condition was used alongside the bounded domain $[-15, 15]$. The bulk velocity and temperature are nearly constant, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision. The entropy also flattens, indicating it reaches its asymptotic behavior.

Since the empirical conservation properties are proven to align with the theoretical conservation properties over a large domain, we would like to challenge our methods. First, take a discontinuous initial condition, namely, the uniform density function with a large domain. Figure 3.9 show the results from employing the LDG method, while Figure 3.10 shows that of the RDG method.

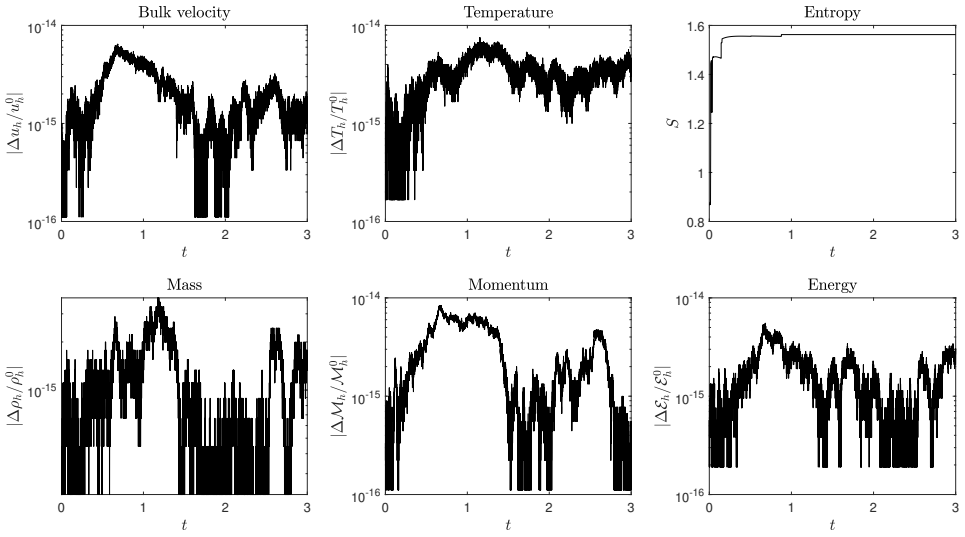

Fig. 3.9: Using the LDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e{-}4$ until time $T = 3$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A discontinuous initial condition was used alongside the bounded domain $[-15, 15]$. The bulk velocity, temperature remain nearly constant over time, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision. The entropy also flattens, indicating it reaches its asymptotic behavior.

Both the LDG and RDG method conserve the momenta for discontinuous initial conditions. The bulk velocity and temperature seem to change more with the RDG method than with the smooth initial condition, but not by a significant amount. The entropy has severe discontinuities, for this we would again like to emphasize that negative values were not counted towards the integral of the entropy. Before the discontinuities, there were severe oscillations, resulting in significant negative values. These discontinuities in entropy thus tell us when these oscillations seemed to smooth out, presumably by the difussion.

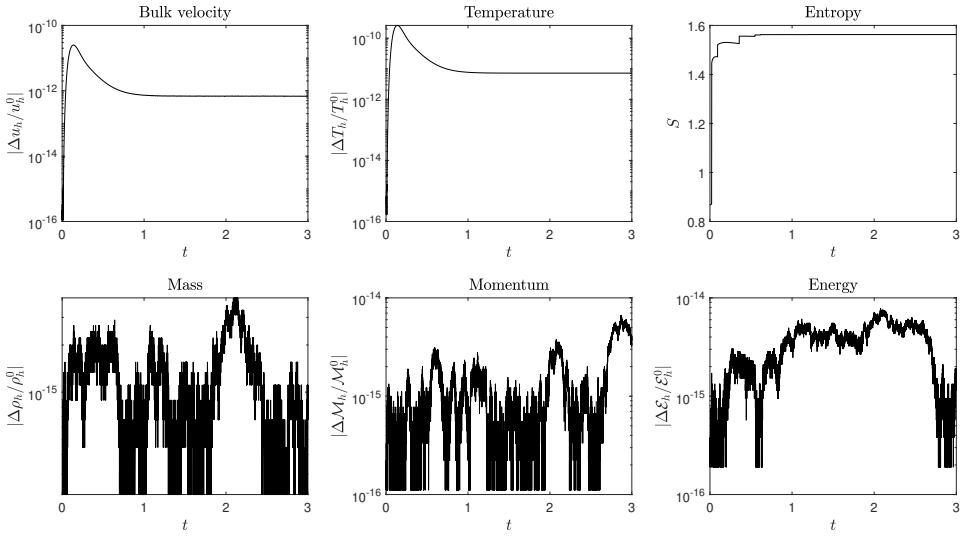

Fig. 3.10: Using the RDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. A discontinuous initial condition was used alongside the bounded domain $[-15, 15]$. The bulk velocity and temperature remain nearly constant over time, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision. The entropy also flattens, indicating it reaches its asymptotic behavior.

Lastly, the primary goal of this thesis is discussed: conservation over a small domain. Figure 3.11 and Figure 3.12 show the conservation results when the domain is shrunken to $\Omega_v = [-4, 4]$.

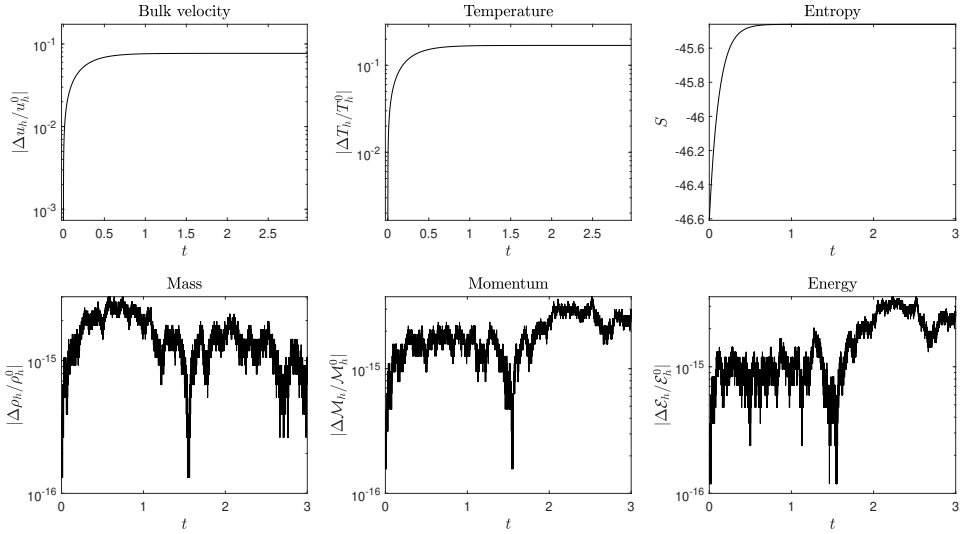

Fig. 3.11: Using the LDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$ on a small domain $\Omega = [-4, 4]$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. The bulk velocity and temperature change over time, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision until. The entropy clearly shows that the asymptotic behavior has been reached.

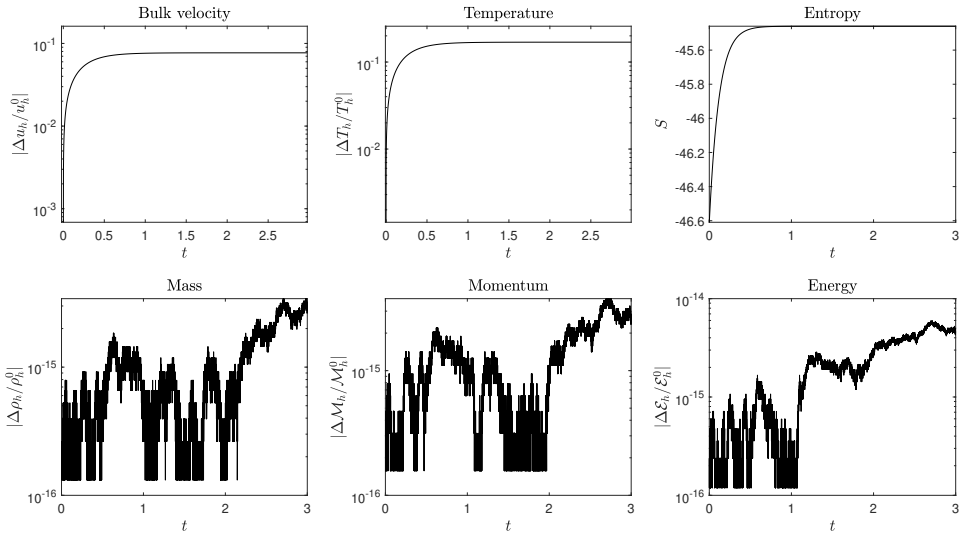

Fig. 3.12: Using the RDG method with piecewise third-order polynomial approximations over 24 elements with time-stepping $\Delta t = 1e-4$ until time $T = 3$ on a small domain $\Omega = [-4, 4]$, the entropy and changes in the bulk velocity, temperature and momenta relative to their respective initial projected values are displayed. The bulk velocity and temperature change over time, while the conservation characteristics of the Fokker-Planck equation are preserved up to machine precision until. The entropy clearly shows that the asymptotic behavior has been reached.

In these figures, all momenta are conserved. Thus, the methods conserve on small domains. Also note that the entropy flattens, so the asymptotic behavior is reached. Figure 3.13 shows us the asymptotic behavior of the approximation and the Maxwellian associated with its initial bulk velocity and temperature over large and small domains. In the original kinetic Fokker-Planck equation, we saw that the solution must go to the Maxwellian using the bulk velocity and temperature from the initial condition. Since now the bulk velocity and temperature can change over time, an initial thought would be that the solution goes to the Maxwellian using the bulk velocity and temperature when the solution stops changing. This, however, is incorrect. This theoretical Maxwellian has a bulk velocity and temperature over an infinite domain, while our solution has this temperature and bulk velocity over a bounded domain. Therefore, we do not expect our approximation to converge to this Maxwellian. Moreover, we do not know if it even goes to a Maxwellian in the first place.

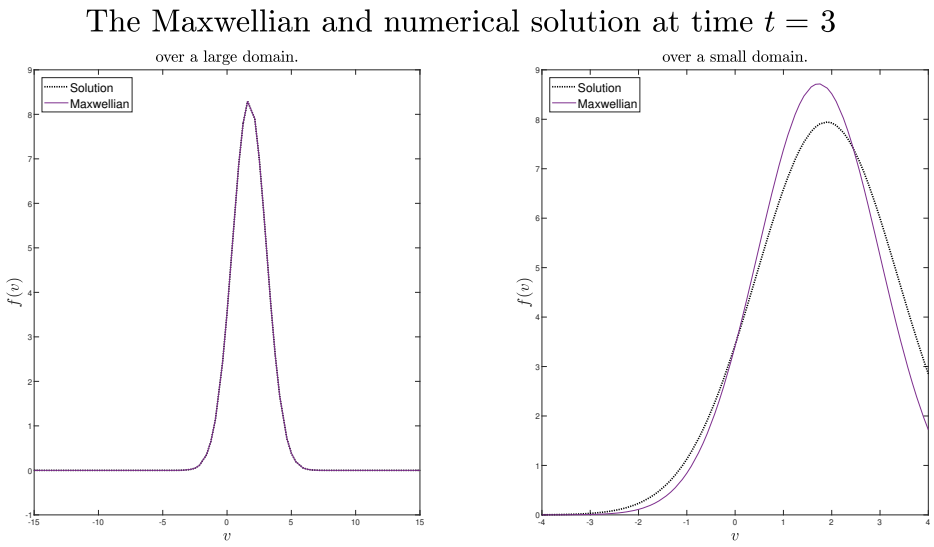

Fig. 3.13: The asymptotic behavior gotten from the methods for both large and small domains versus the Maxwellian distribution with the bulk velocity and temperature associated with the initial condition over an infinite domain. There is a no noticeable difference between the two on the left, but quite a noticeable difference to the two on the right. Therefore, the approximation does not tend to the associated Maxwellian over a small domain. That being said, the approximation does look like a Maxwellian distribution.

3.2 Stability

Now that it is known our methods conserve, we hope they give a stable and accurate solution. It is imperative to look at stability before delving into the empirical convergence results, as they depend on stability itself. For this the eigenvalues of the matrix A^n, found in section 2.4 by equation (2.35), for LDG, and equation (2.36), for RDG, should have strictly negative real components for all time steps. Proving this would indicate that a Δt can be chosen such that the method is stable.

The stability analysis is done using piecewise first, second and third-order polynomials. One might ask: Why use piecewise linear and second-order approximations? First, by changing the definition of our momenta, the method can be conservative for these definitions. Second, and more importantly, since some momenta are not conserved, our bulk velocity and temperature change over time with large bounds. Thus, we can more easily observe the effects of the bulk velocity and temperature on the eigenvalues. Before studying their effects, Figure 3.14 shows the general formations that the eigenvalues make for both the LDG and RDG method for piecewise first, second and third-order polynomials.

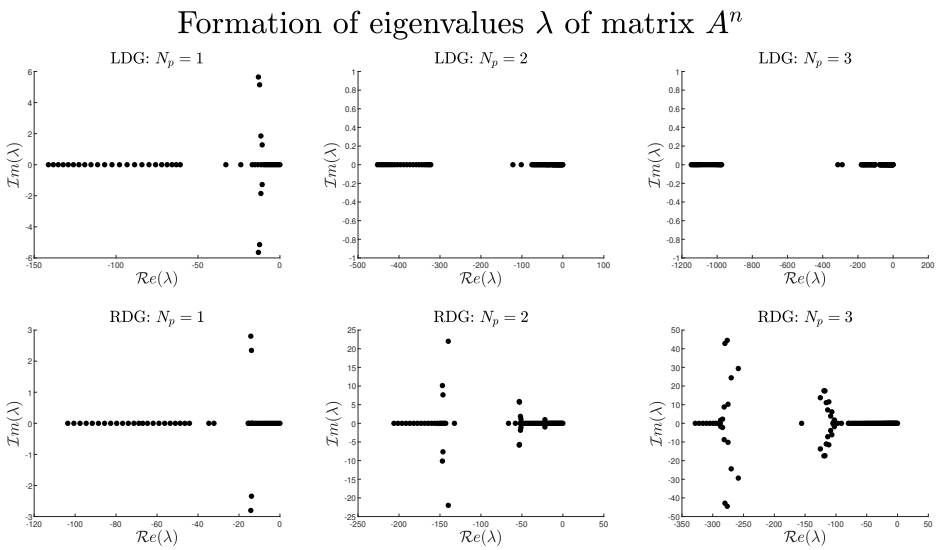

Fig. 3.14: The formation of the eigenvalues for LDG (top row) and RDG (bottom row) for approximation using piecewise first (left), second (central) and third-order polynomials (right) over 24 elements on $\Omega_v = [-15, 15]$. All eigenvalues have negative real components, indicating stability is possible. One eigenvalue is always zero, which does not pose a problem.

From these results, we see that all eigenvalues for both methods have negative real components. However, one eigenvalue is zero. Note in particular that the eigenvalues associated with the LDG method for second-order polynomials or higher are real even for a low amount of grid points. The natural followup is to refine the grid and then see what the eigenvalues look like. This is shown in Figure 3.15:

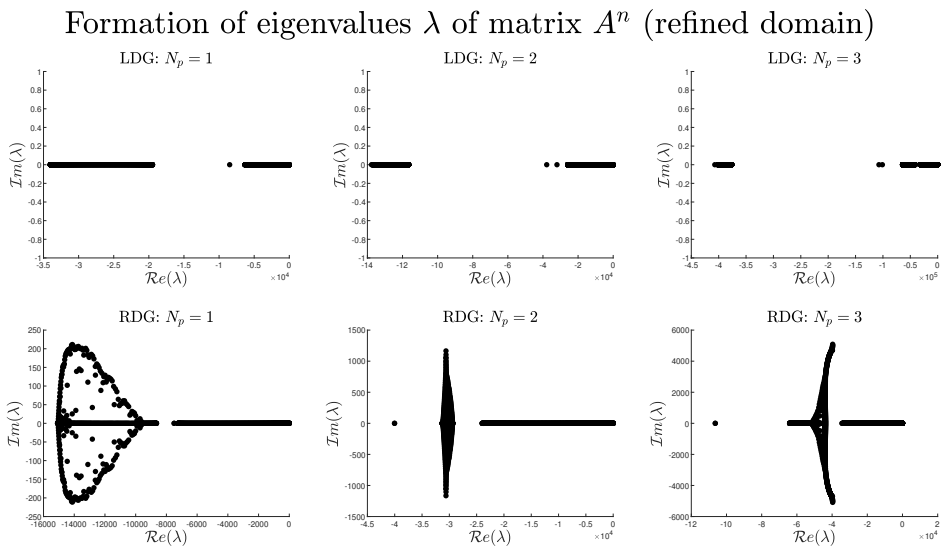

Fig. 3.15: The formation of the eigenvalues for LDG (top row) and RDG (bottom row) for approximation using piecewise first (left), second (central) and third-order polynomials (right) over 500 elements on $\Omega_v = [-15, 15]$. Both methods only have eigenvalues with negative real components, indicating stability is possible. One eigenvalue for both methods is always zero. Also note that the eigenvalues for the LDG method are real as the grid is refined.

These results show that the eigenvalues keep the negative real component even after refining the grid. The eigenvalues expand, which is natural when a grid is refined. Luckily this expansion happens over the negative real domain. Note that the eigenvalues of the LDG method become real. Since the real components stay negative, stability would be possible with a small enough time-stepping if the matrix A^n does not change over time. However, since they depend on u^n and T^n, they will change over time. To analyze the influence of the bulk velocity and temperature we look at the usage of piecewise first and second-order approximations over a large domain and third-order polynomial approximations over a small domain for both methods. The linear approximations' inability to conserve momentum and energy on a coarse grid results in a rapid change in bulk velocity and temperature, even for large domains. Therefore, the stability is only influenced by change in bulk velocity and temperature. The piecewise second-order polynomials allow us to see the influence of the change in temperature. Lastly, for a fuller picture, we also want to see how the bulk velocity and temperature change due to a small domain. Therefore, we look at the usage of piecewise third-order polynomials over a small domain. Figure 3.16 shows the results found

for both methods.

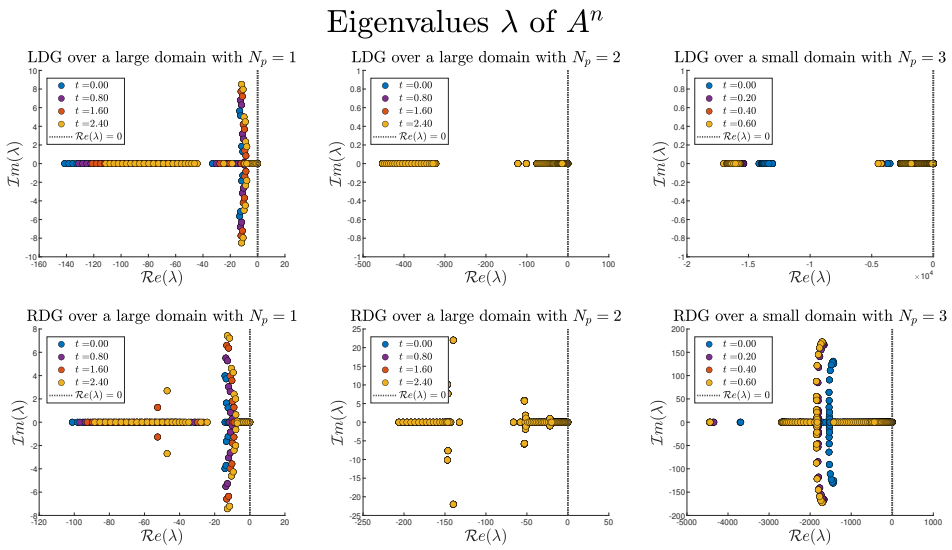

Fig. 3.16: The eigenvalues of matrix A^n using the LDG (top row) and RDG (bottom row) method with piecewise first (left), second (central) and third-order polynomial (right) approximations indicate that the methods exhibit the same changes over time. When zooming in, the piecewise first and second-order approximations slowly approach the instability region, thus there might be a possibility of instability. The opposite effect is shown for piecewise third-order polynomials. Both methods tend away from the instability region as the bulk velocity and temperature change, indicating stability is possible for both methods with a small enough Δt.

The piecewise first-order approximation shows a significant change for both methods towards the instability region. The piecewise second-order approximation does not change a lot over time, since the energy and momentum do not change a lot over time. They do move however. Both methods slowly reach the instability region for piecewise first and second-order approximations. Since the piecewise first-order approximations do not reach a constant solution in time, there will be a time when the method becomes unstable, unless the bulk velocity and temperature turn constant at some point. The piecewise third-degree approximations tend away from the instability region for both methods and eventually become constant. The magnitude of the eigenvalues increases, but the real components stay negative. Stability is then possible, but as there is no apriori result for the bulk velocity and temperature, a choice in Δt is difficult at first. In order to solve this, the method can run on a coarser grid until it reaches its asymptotic behavior and then choosing a Δt based

on the bulk velocity and temperature of that asymptotic behavior. This holds if it is known that the solution converges to a unique asymptotic solution, which is not the case here.

In conclusion, the methods can be stable using polynomials. The crucial part of stability is the bulk velocity and temperature of the solution at all times. We end this section by finding the relation between the time-stepping, spatial discretization and order of polynomial approximation for the both methods over a large domain with third or higher-order polynomial approximations. Figure 3.17 shows our empirical findings:

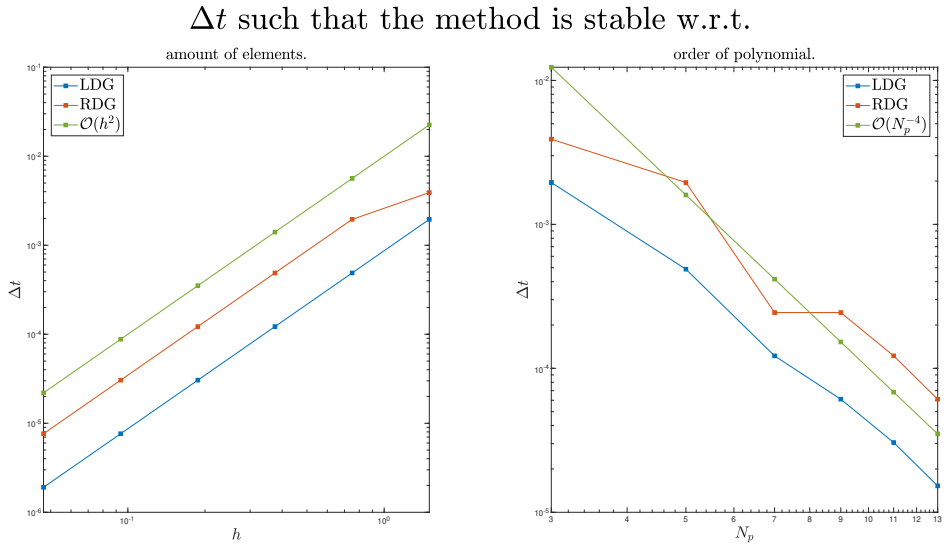

Fig. 3.17: For an initial condition characterized with $u = .5$ and $T = 3.25$ over a large domain such that these are constant over time. The changes in Δt such that the LDG and RDG methods are stable as our grid spacing h changes (left) indicates a relation of h^2. The change with respect to the order of the polynomial N_p (right) indicates a relation of N_p^{-4}. Therefore, stability holds when $\Delta t = \mathcal{O}\left(\frac{h^2}{N_p^4}\right)$. Note that the LDG method requires significantly smaller time-stepping.

Our empirical results show that stability over a large domain holds when $\Delta t = \mathcal{O}\left(\frac{h^2}{N_p^4}\right)$, as previously suspected. The LDG method requires significantly smaller time-stepping.

This rate of Δt is largely due to the influence of the diffusion term. The natural question is then: what if we mitigate this term? By setting the temperature to be small, Figure 3.18 shows the influence on the relation between Δt, h and N_p.

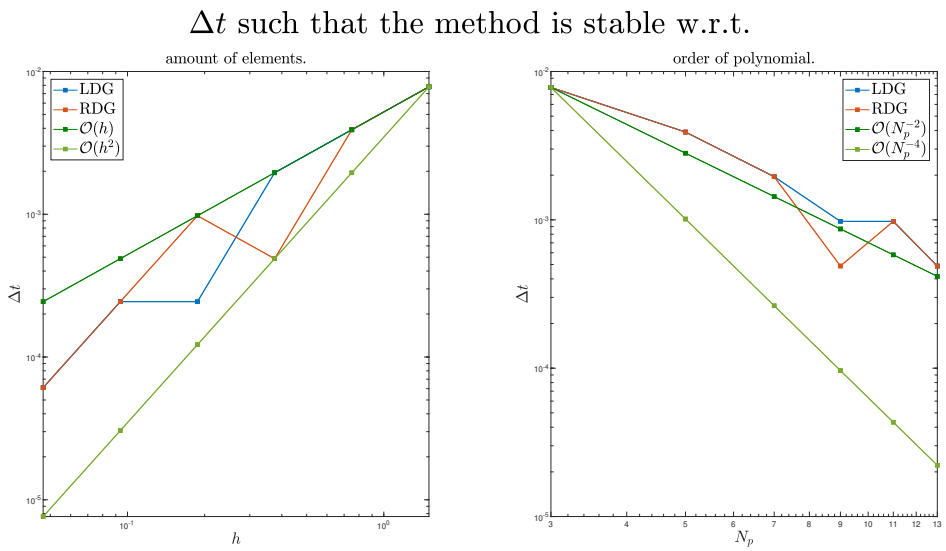

Fig. 3.18: For an initial condition characterized with $u = .5$ and $T = 1.e^{-3}$ over a large domain such that these are constant over time. The changes in Δt such that the LDG and RDG methods are stable as our grid spacing h changes (left) indicates a relation between h and h^2. The change with respect to the order of the polynomial N_p (right) indicates a relation between N_p^{-2} and N_p^{-4}. Therefore, the influence of the diffusion term is large. Note that the LDG and RDG methods require about the same time-stepping. This is the numerical flux applied to the drift term taking over. This numerical flux is the same for both methods.

Our empirical results show that stability over a large domain holds when Δt is between $\mathcal{O}\left(\frac{h}{N_p^2}\right)$ and $\mathcal{O}\left(\frac{h^2}{N_p^4}\right)$. The effects of the diffusion term on stability is thus quite large. Also note that the LDG and RDG methods require about the same time-stepping. This as the numerical flux applied to the drift term takes over. This numerical flux is the same for both methods, thus not much difference should be seen.

3.3 Convergence

Now that because it is empirically proven that stability can hold, convergence can be confirmed. It is worth recalling that theoretical results demonstrate that the spatial convergence order of these discontinuous Galerkin methods are $N_p + 1$ for smooth functions. As for the temporal convergence order, the forward Euler and SSP-RK2 method are first and second-order methods, respectively. Hence, when the velocity discretization is halved, the temporal discretization must be multiplied by $2^{-(N_p+1)}$ and $2^{-(N_p+1)/2}$ for the forward Euler and SSP-RK2 methods, respectively.

We first extend the size of our domain to ensure that the solutions dissipate enough such that the theoretical solution (1.9) serves as a suitable approximation over this expanded bounded region. Figure 3.19 presents the method applied to a smooth initial condition using linear approximations, on the left, and by a piecewise third-order polynomial on the right. In both cases, the initial condition is the sum of two Gaussians.

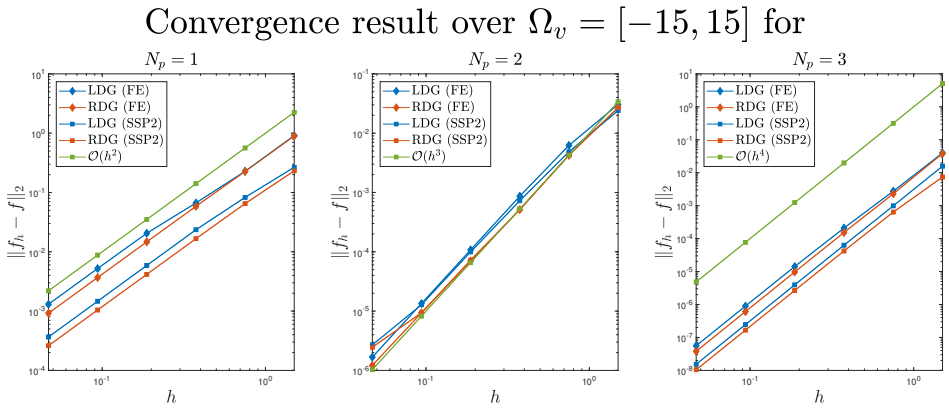

Fig. 3.19: Convergence rate of the LDG (blue) and RDG (red) methods using both forward Euler (FE) and strong stability preserving Runge-Kutta (SSP) with a smooth initial condition. Approximations are computed using a piecewise first (left) , second (central) and third-order polynomial (right) for the Fokker-Planck equation where $\Omega_v = [-15, 15]$. The error is calculated with respect to the L^2 norm, using the analytical solution (1.9). This points to a convergence rate of $\mathcal{O}\left(h^{N_p+1}\right)$ as suspected.

The loglog plots show that the error of the approximation follows the same trend as a reference line of order h^{N_p+1}. Note that the RDG method does provide a more accurate solution, but not by much. Now one can also look at discontinuous initial conditions and see how our methods compare. Take the the uniform density function as the initial condition. Figure 3.20 shows the convergence results.

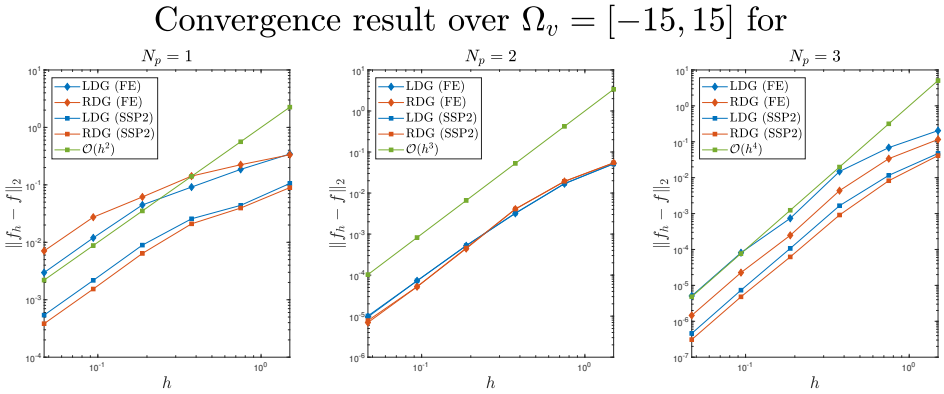

Fig. 3.20: Convergence rate of the LDG (blue) and RDG (red) methods using both forward Euler (FE) and strong stability preserving Runge-Kutta (SSP) with a discontinuous initial condition. Approximations are computed using a piecewise first (left), second (central) and third-order polynomial (right) for the Fokker-Planck equation where $\Omega_v = [-15, 15]$. The error is calculated with respect to the L^2 norm, using the analytical solution (1.9). This points to a convergence rate of $\mathcal{O}\left(h^{N_p+1}\right)$ as suspected.

These figures show that the RDG method might not always be more accurate as the convergence plot for the piecewise polynomials seems to show. The convergence of the method is again of order $\mathcal{O}(h^{N_p+1})$. Note that we do not lose an order of accuracy for discontinuous initial conditions, as a diffusive term is included in the Fokker-Planck equation, thus smoothing the solution.

The true interest of this thesis is applying the method to a more restrained bounded domain. Within these tighter boundaries, the analytical solution is no-longer applicable. Consequently, we employ a highly refined grid to compute a reference solution. Figure 3.21 shows the convergence results for a small domain where the function is never close to zero at the boundary.

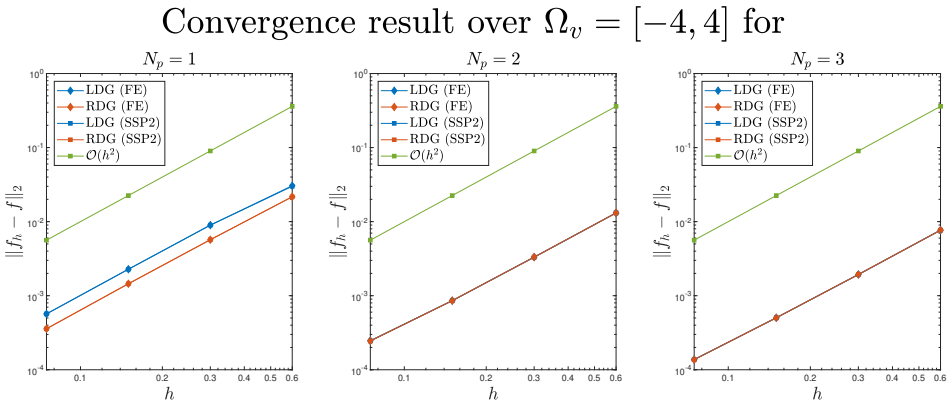

Fig. 3.21: Convergence rate of the LDG (blue) and RDG (red) methods using both forward Euler (FE) and strong stability preserving Runge-Kutta (SSP) on a small domain. Approximations are computed using a piecewise linear (left), second (central) and third-order polynomial (right) for the Fokker-Planck equation over a small domain. The error is calculated using a reference solution on a bounded domain. For linear approximations, the forward Euler and SSP-RK2 method give nearly the same error. The convergence rate is of order $\mathcal{O}(h^2)$. This convergence rate also holds for the other approximations, but now the errors are nearly identical as well. This does not fall in line with the theoretical convergence order of $\mathcal{O}(h^{N_p+1})$.

These results are unexpected. For both methods, there is nearly no difference in approximation between the forward Euler or the SSP-RK2 methods when using piecewise linear appoximations. Then, when we increase the order of polynomial, there is nearly no difference between the LDG or RDG methods. The most unexpected result is that the convergence order is no longer $\mathcal{O}(h^{N_p+1})$, but rather $\mathcal{O}(h^2)$.

Chapter 4

DISCUSSIONS AND CONCLUSIONS

Both the Local Discontinuous Galerkin and Recovery Discontinuous Galerkin methods using explicit time discretizations succeed in conserving the mass, momentum and energy of the solution to the kinetic Fokker-Planck equation, when using piecewise third, or higher-order polynomials, regardless of the size of the domain. The order of piecewise polynomial approximations should be higher than two, even though theoretical results show successful results at lower orders.

The bottleneck of these methods is not conservation, but stability. This largely depends on the bulk velocity and temperature, which are computed using the solution at the boundary. Therefore, this originally-linear problem becomes non-linear and stability becomes complex. When the method conserves the momenta, stability is possible, but a small enough time-stepping should be taken, for which we know no a priori results. The stability study indicates that stability holds when the time step is $\Delta = \mathcal{O}(h^2/N_p^4)$. The study also showed the methods are convergent on large domains with a convergence order of N_p+1. Small domains indicate the convergence order is 2, which is unexpected.

Further research should include a more theoretical approach to stability and convergence. A focus should be put on finding out why the convergence rate changes when the domain size changes. In order to facilitate this research, an in-depth study should be done on the kinetic Fokker-Planck equation, or even the Vlasov-Fokker-Planck equation. Moreover, entropy and the change of the bulk velocity and temperature should be researched. Additionally, it was expected that the bulk velocity and temperature over a small domain should not align with that of the unbounded domain. An interesting study would be to determine if there is a choice in numerical flux at the boundary that might alter this. Lastly, this method should be applied to higher dimensions and possibly the Vlasov-Fokker-Planck equation.

BIBLIOGRAPHY

Gottlieb, S., Ketcheson, D., and Shu, C.-W. (2011). *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific.

Hakim, A., Francisquez, M., Juno, J., and Hammett, G. W. (2020). Conservative discontinuous galerkin schemes for nonlinear dougherty–fokker–planck collision operators. *Journal of Plasma Physics*, 86(4):905860403.

Hesthaven, J. S. and Warburton, T. (2007). *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media.

Landau, L. (1936). Die kinetische gleichung fuer den fall coulombscher wechselwirkung. *Phys. Z. Sowjet. 10 (1936)*, 10:163.

Lenard, A. and Bernstein, I. B. (1958). Plasma oscillations with diffusion in velocity space. *Physical Review*, 112(5):1456.

Montgomery, D. C. and Tidman, D. A. (1964). Plasma kinetic theory. *McGraw-Hill Advanced Physics Monograph Series*.

Risken, H., . F. T. (1992). *The Fokker-Planck equation : Methods of solution and applications.*, volume 18. Springer Berlin / Heidelberg.

Rosenbluth, M. N., MacDonald, W. M., and Judd, D. L. (1957). Fokker-planck equation for an inverse-square force. *Physical Review*, 107(1):1.

Sornette, D. (2001). Fokker–planck equation of distributions of financial returns and power laws. *Physica A: Statistical Mechanics and its Applications*, 290(1-2):211–217.

van Leer, B., Lo, M., and van Raalte, M. (2007). A discontinuous galerkin method for diffusion based on recovery. In *18th AIAA computational fluid dynamics conference*, page 4083.

Van Leer, B. and Nomura, S. (2005). Discontinuous galerkin for diffusion. In *17th AIAA Computational Fluid Dynamics Conference*, page 5108.