Sparse-grid Discontinuous Galerkin Methods for the Vlasov–Poisson–Lenard–Bernstein Model ⋆,⋆⋆

Stefan Schnake[a,*], Coleman Kendrick[c], Eirik Endeve[a,d], Miroslav Stoyanov[a], Steven Hahn[a], Cory D. Hauck[a,e], David L. Green[g], Phil Snyder[b], John Canik[f]

[a] *Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA*
[b] *Fusion Energy Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA*
[c] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA*
[d] *Department of Physics and Astronomy, University of Tennessee Knoxville, TN 37996 USA*
[e] *Department of Mathematics, University of Tennessee Knoxville, TN 37996 USA*
[f] *Type One Energy Group, Oak Ridge, TN 37830 USA*
[g] *Commonwealth Scientific and Industrial Research Organisation, Mayfield West, NSW, 2304, Australia*

Abstract

Sparse-grid methods have recently gained interest in reducing the computational cost of solving high-dimensional kinetic equations. In this paper, we construct adaptive and hybrid sparse-grid methods for the Vlasov–Poisson–Lenard–Bernstein (VPLB) model. This model has applications to plasma physics and is simulated in two reduced geometries: a $0x3v$ space homogeneous geometry and a $1x3v$ slab geometry. We use the discontinuous Galerkin (DG) method as a base discretization due to its high-order accuracy and ability to preserve important structural properties of partial differential equations. We utilize a multiwavelet basis expansion to determine the sparse-grid basis and the adaptive mesh criteria. We analyze the proposed sparse-grid methods on a suite of three test problems by computing the savings afforded by sparse-grids in comparison to standard solutions of the DG method. The results are obtained using the adaptive sparse-grid discretization library ASGarD.

Keywords: Kinetic equation, Discontinuous Galerkin, Implicit-Explicit, Sparse Grids, Vlasov–Poisson, Lenard–Bernstein

⋆⋆This material is based upon work partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, as part of their Applied Mathematics Research Program; the U.S. Department of Energy, Office of Science, Office of Fusion Energy Science as part of their Fusion Research Energy Program; and the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. De-AC05-00OR22725. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

⋆Corresponding author. Address: PO Box 2008, MS6013, Oak Ridge, TN 37831-6013

Email addresses: schnakesr@ornl.gov (Stefan Schnake), kendrick6@llnl.gov (Coleman Kendrick), endevee@ornl.gov (Eirik Endeve), stoyanovmk@ornl.gov (Miroslav Stoyanov), hahnse@ornl.gov (Steven Hahn), hauckc@ornl.gov (Cory D. Hauck), d.green@csiro.au (David L. Green), snyderpb@ornl.gov (Phil Snyder), john.canik@typeoneenergy.com (John Canik)

1. Introduction

In this paper, we investigate sparse-grid solutions to kinetic equations with applications in plasma physics. In a general setting, equations of this type are defined in terms of a kinetic distribution f that evolves over a six-dimensional phase space (three position and three velocity variables). To discretize f in phase-space, we work with sparse-grid approximations based on the discontinuous Galerkin (DG) method. First introduced for kinetic models of radiation transport [70], the DG method is a finite element method that offers high-order accurate solutions to elliptic [71] and hyperbolic partial differential equations (PDEs) [24] with compact stencils. In addition to being locally conservative, DG methods are efficient at preserving important physical constraints and structural properties inherited from the underlying PDEs that they are used to simulate. Such properties include positivity [89], energy conservation [34, 88, 85], asymptotic limits [55, 2, 38, 76, 86, 82], entropy stability [18, 33, 16, 87, 8], and invariant domains [65, 23, 63]. For these reasons, the DG method has become a popular tool in the simulation of kinetic equations [43, 29, 3, 19, 20, 26, 30, 35, 46, 69, 1].

When applied to high-dimensional PDEs, Eulerian grid-based methods, including DG, suffer from the *curse of dimensionality* [9], where the cost to approximate a general measurable function scales like $\mathcal{O}(N^d)$, with d the dimension of the domain and N the degrees of freedom in a single dimension. Such a scaling in six dimensions makes the standard DG method intractable for approximating general kinetic equations, even on leadership class computing facilities [39].

Particle-based methods, e.g. particle in cell, attempt to mitigate the curse of dimensionality using a Lagrangian approach [11, 17, 53, 47, 45, 15]. However, there has been recent interest in reducing the computational and memory footprint of Eulerian methods by compressing the full-resolution distribution. One popular technique is low-rank approximations where the discretized kinetic distribution is treated as a d-mode tensor and compressed using a low-rank factorization. The low-rank decomposition is evolved through time using methods such as step-truncation [72, 41] or dynamical low-rank approximation [28, 27]. Another popular avenue is the sparse-grid method [40, 83, 49] which is the focus of this paper.

The sparse-grid method [12] is a general technique used for the approximation of high-dimensional functions. These methods replace the $\mathcal{O}(N^d)$ scaling of tensor-based discretizations to $\mathcal{O}(N(\log N)^{d-1})$. First developed for the integration of high-dimensional functions [77, 36], current flavors of the sparse-grid method are far reaching. Sparse-grid interpolation has been successfully employed in the construction of surrogate models [13, 21] including addressing challenges of adaptivity for basis with local support [68, 66, 67, 58, 52, 79], global support [81, 61, 62, 60], and even discontinuous response surfaces [50, 51, 80].

Additionally, sparse-grids have gained favor in the approximation of high-dimensional PDEs with examples in finite differences [56], finite volumes [44, 59], conforming finite element methods [73, 74, 75, 7] as well as the DG method [84, 39]. In the DG context, functions are decomposed in a multiwavelet basis [4] with specific basis functions discarded via a sparse-grid selection rule. This multiwavelet decomposition induces a decay in the coefficient's magnitude over finer levels. This decay is utilized to build model-independent criteria for adaptively choosing whether to keep or discard basis functions. This is referred to as the adaptive sparse-grid DG method and has shown promise in the modelling of kinetic equations [54, 40, 83].

The main goal of this work is to study the computational savings provided by sparse-grids on the Vlasov–Poisson–Lenard–Bernstein (VPLB) model. We measure the savings by the reduction of the total degrees of freedom required to accurately represent the solution. The grids of choice are the adaptive sparse-grid DG method and a hybrid sparse-grid method, called the mixed-grid method, which is a standard DG grid in position space tensored with a sparse-grid in velocity space. Similar hybrid splittings have been studied in the context of collisionless kinetic problems [54]. The methods are tested on the following three problems: a simple relaxation to a Maxwellian equilibrium, the Sod shock tube problem [78], and an example of collisional Landau damping [25, 43, 32]. In each problem, we present the computational savings achieved as well as general qualitative performance, such as capturing desired physical features, of the methods presented. In general, the adaptive sparse-grid method significantly reduces the storage cost of the distribution while the mixed-grid method only provides favorable savings in determining lower-order moments of the distribution.

We work with the VPLB model on a slab geometry which reduces the problem to a four-dimensional $1x3v$ problem (one position dimension, three velocity dimensions). This is done so that the problem size is sufficiently small to be run on a single node machine; the Chu reduction method [22] can be utilized to

further reduce the problem to a moment model in $1x1v$ which then allows the creation of fine-resolution reference solutions; and, in a slab geometry, the VPLB model can be written as a sum of terms which are each tensor products of one-dimensional PDE operators. The latter property, commonly referred to as separability, allows efficient evaluations of the model on sparse-grids. We note that recent endeavors, e.g. interpolatory wavelets [49], are being researched to bypass the separability condition. Nonetheless, DG approximations to the slab problem with sufficient resolution to accurately capture fine-scale features will still tax a single node machine and thus will require some sort of compression. Additionally, we are keeping three velocity dimensions which we expect to provide the main source of savings captured by the adaptive sparse-grid DG method; this is because, locally in space, the collision operator mollifies the distribution in the velocity domain, which will in turn cause a rapid decay in the coefficient size when the distribution is represented in the multiwavelet DG basis. Moreover, in regimes of high collisionality, where the distribution typically approaches a local thermal equilibrium that is very smooth in the velocity variable, the distribution in physical space is close to a fluid model which can form shocks and other non-smooth features. Therefore, we believe the $1x3v$ geometry provides an indication of the savings expected in the full $3x3v$ model, as well as clues for constructing hybrid approaches.

Complementing this work is the development of the adaptive sparse-grid DG codebase ASGarD (Adaptive Sparse-Grid Discretization) [37]. The goal of this open-source project is to facilitate and promote the use of adaptive sparse-grid methods for the approximation of kinetic models by providing a robust yet flexible adaptive sparse-grid library. All sparse-grid results of this work were computed using ASGarD. The algorithmic specifics of how ASGarD evaluates PDE operators will be delayed for a future work. This manuscript only focuses on the mathematics of the adaptive sparse-grid method and the memory reduction realized via its utilization.

The rest of the paper is organized as follows. In Section 2, we present the VPLB model, as well as the Chu reduction method for generating reference solutions and the geometric reductions used to formulate the aforementioned test problems. In Section 3, we present the DG method for the VPLB model which we refer to as the full-grid method. Section 4 provides an overview to the standard and adaptive sparse-grid methods and details the specifics implemented in ASGarD. In Section 5, we analyze the results of the adaptive sparse-grid and mixed-grid methods, compared against the full-grid method, for the chosen suite of test problems. Finally, Section 6 gives our conclusions and future plans.

2. The Vlasov–Poisson–Lenard–Bernstein Model

The Vlasov–Poisson–Lenard–Bernstein (VPLB) model describes the dynamics of charged particles influenced by a self-consistent electric field and collisional dynamics. It couples a kinetic equation for the phase-space distribution function of charged particles with a Poisson equation for the electrostatic potential.

Assuming ions of unit mass and charge, the governing kinetic equation is

$$\partial_t f(\mathbf{x},\mathbf{v},t) + \mathbf{v}\cdot\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{v},t) + \mathbf{E}(\mathbf{x},t)\cdot\nabla_{\mathbf{v}} f(\mathbf{x},\mathbf{v},t) = \mathcal{C}_{\text{LB}}(f)(\mathbf{x},\mathbf{v},t), \tag{2.1}$$

where the phase-space distribution function f depends on position $\mathbf{x} = (x_1, x_2, x_3)^\top \in \Omega_{\mathbf{x}} \subseteq \mathbb{R}^3$, velocity $\mathbf{v} = (v_1, v_2, v_3)^\top \in \mathbb{R}^3$, and time $t \geq 0$. The electric field $\mathbf{E} = -\nabla_{\mathbf{x}}\Phi$ is obtained from the electrostatic potential Φ by solving the Poisson equation

$$-\nabla_{\mathbf{x}}\cdot\nabla_{\mathbf{x}}\Phi(\mathbf{x},t) = n_f(\mathbf{x},t) - n_{\text{e}}, \tag{2.2}$$

where $n_f = \langle f\rangle_v \equiv \int_{\mathbb{R}^3} f\,\mathrm{d}\mathbf{v}$ is the ion density, and n_{e} is a constant background electron density chosen to enforce global charge neutrality: $n_e = \int_{\Omega_{\mathbf{x}}} n_f(\mathbf{x},t)\,\mathrm{d}\mathbf{x}$ for all $t \geq 0$. We assume the physical domain Ω_x is periodic. The collision operator $\mathcal{C}_{\text{LB}}$ on the right-hand side of (2.1) is the Lenard–Bernstein (LB) operator [57]. It takes the form [43, 32]

$$\mathcal{C}_{\text{LB}}[\boldsymbol{\rho}_f](f)(\mathbf{x},\mathbf{v},t) = \nu\nabla_{\mathbf{v}}\cdot\big((\mathbf{v}-\mathbf{u}_f)\,f + \theta_f\nabla_{\mathbf{v}} f\big), \tag{2.3}$$

where $\nu \geq 0$ is a collision frequency that is assumed to be a constant independent of $\mathbf{v}$, $\mathbf{x}$, and t. The moments of f,

$$\boldsymbol{\rho}_f = \langle \mathbf{e} f \rangle_v, \quad \text{where} \quad \mathbf{e} = (e_0, \mathbf{e}_1, e_2)^\top \equiv (1, \mathbf{v}, \frac{1}{2}|\mathbf{v}|^2)^\top, \tag{2.4}$$

represent the number, momentum, and energy densities, respectively, and the bulk velocity and temperature are defined from $\boldsymbol{\rho}_f$ by

$$\mathbf{u}_f = \frac{1}{n_f}\langle f\mathbf{v}\rangle_v \quad \text{and} \quad \theta_f = \frac{1}{3n_f}\langle f|\mathbf{v}-\mathbf{u}_f|^2\rangle_v. \tag{2.5}$$

Direct calculations show that

$$\boldsymbol{\rho}_f = \big(\, n_f,\, n_f\mathbf{u}_f,\, n_f(\frac{1}{2}|\mathbf{u}_f|^2 + \frac{3}{2}\theta_f)\,\big)^\top. \tag{2.6}$$

Proposition 1 ([43, 29]). *The LB operator satisfies the following properties*

1. *Conservation of number, momentum, and energy:*

$$\langle \mathcal{C}_{\text{LB}}[\boldsymbol{\rho}_w](w)\mathbf{e}\rangle_v = 0, \qquad \forall w \in \text{Dom}(\mathcal{C}_{\text{LB}}). \tag{2.7}$$

2. *Dissipation of entropy:*

$$\langle \mathcal{C}_{\text{LB}}[\boldsymbol{\rho}_w](w)\log w\rangle_v \leq 0, \qquad \forall w \in \text{Dom}(\mathcal{C}_{\text{LB}}). \tag{2.8}$$

3. *Characterization of equilibria: For any $w \in \text{Dom}(\mathcal{C}_{\text{LB}})$,*

$$\langle \mathcal{C}_{\text{LB}}[\boldsymbol{\rho}_w](w)\log w\rangle_v = 0 \tag{2.9}$$

if and only if w is a Maxwellian distribution, i.e.,

$$w = M_w := \frac{n_w}{(2\pi\theta_w)^{3/2}}\exp\Big\{\frac{-|\mathbf{v}-\mathbf{u}_w|^2}{2\theta_w}\Big\}. \tag{2.10}$$

Proposition 2 ([43, 29]). *On a periodic spatial domain $\Omega_{\mathbf{x}} \subseteq \mathbb{R}^3$, the VPLB model satisfies the following global conservation laws:*

1. *Conservation of number:*

$$\partial_t \int_{\Omega_{\mathbf{x}}} \langle f\rangle_v \,\mathrm{d}\mathbf{x} = 0. \tag{2.11}$$

2. *Conservation of momentum:*

$$\partial_t \int_{\Omega_{\mathbf{x}}} \langle \mathbf{e}_1 f\rangle_v \,\mathrm{d}\mathbf{x} = 0. \tag{2.12}$$

3. *Conservation of energy:*

$$\partial_t \int_{\Omega_{\mathbf{x}}} (\langle e_2 f\rangle_v + \frac{1}{2}|\mathbf{E}|^2)\,\mathrm{d}\mathbf{x} = 0. \tag{2.13}$$

2.1. Geometric reductions

2.1.1. Space homogeneous problem

To investigate the relaxation induced by the LB collision operator of a velocity distribution to a Maxwellian, we consider the equation (2.3) under the assumption that f does not depend on $\mathbf{x}$. In this case, the PDE is given by

$$\partial_t f(\mathbf{v},t) = \nu\,\mathcal{C}_{\text{LB}}(f)(\mathbf{v},t). \tag{2.14}$$

2.1.2. Reduction to slab geometry

Under the assumption that $\partial_y f = \partial_z f = 0$, the VPLB model (2.1) reduces to

$$\partial_t f(x,\mathbf{v},t) + v_x \partial_x f(x,\mathbf{v},t) + E\partial_{v_x} f(x,\mathbf{v},t) = \mathcal{C}_{\text{LB}}(f)(x,\mathbf{v},t), \tag{2.15}$$

where $E := E_x = -\partial_x \Phi$ and Φ satisfies

$$-\partial_{xx}\Phi(x) = n_f(x,t) - n_e. \tag{2.16}$$

Let $(v_r, \vartheta, \varphi)$ be a spherical-polar coordinate system in which the x-axis is aligned with the polar direction, so that

$$v_x = v_r \cos\vartheta, \quad v_y = v_r \sin\vartheta\cos\varphi, \quad \text{and} \quad v_z = v_r \sin\vartheta \sin\varphi, \tag{2.17}$$

where $v_r = |\mathbf{v}|$, ϑ is the polar angle, and φ is the azimuthal angle. We assume further that f is independent of φ; as a result $(u_f)_y = (u_f)_z = 0$, and by abuse of notation we set

$$(u_f)_x := u_f = \frac{\langle f v_x \rangle_v}{n_f} \tag{2.18}$$

so that $\mathbf{u}_f = [u_f, 0, 0]^\top$. The equation (2.15) has a phase space with four total dimensions: one for physical space and three for velocity space, i.e., $1x3v$.

2.2. Reduction to $1x1v$

The Chu reduction method is a tool for further reducing the slab geometry problem to $1x1v$, at the cost of solving an additional equation. It was first developed in [22] for the Bhatnagar–Gross–Krook (BGK) equation and is used here to provide reference solutions in Section 5 for sparse-grid simulations when exact solutions are not known and full-grid reference calculations are prohibitively expensive.

To derive the Chu reduction of (2.15), let

$$g_1(x,v_x) = \int_{\mathbb{R}^2} f(x,\mathbf{v})\,\mathrm{d}v_y\,\mathrm{d}v_z \quad \text{and} \quad g_2(x,v_x) = \int_{\mathbb{R}^2} (v_y^2 + v_z^2) f(x,\mathbf{v})\,\mathrm{d}v_y\,\mathrm{d}v_z, \tag{2.19}$$

Testing (2.15) by 1 and by $v_y^2 + v_z^2$, respectively and integrating over $\mathrm{d}v_y\,\mathrm{d}v_z$ yields the following coupled system in (x, v_x):

$$\partial_t g_1 + v_x \partial_x g_1 + E\partial_{v_x} g_1 = \nu\,\mathcal{C}_1(g_1; u_f, \theta_f), \tag{2.20a}$$

$$\partial_t g_2 + v_x \partial_x g_2 + E\partial_{v_x} g_2 = \nu\,\mathcal{C}_1(g_2; u_f, \theta_f) + \nu\,(\,4\theta_f g_1 - 2g_2\,), \tag{2.20b}$$

where

$$\mathcal{C}_1(g; u, \theta) = \partial_{v_x}((v_x - u)g + \theta\partial_{v_x} g), \quad E = -\partial_x\Phi, \quad \text{and} \quad -\partial_{xx}\Phi = n_f - n_e, \tag{2.21}$$

and, importantly, the velocity moments of f can be expressed in terms of g_1 and g_2:

$$n_f = \int_{\mathbb{R}} g_1\,\mathrm{d}v_x, \quad u_f = \frac{\int_{\mathbb{R}} g_1 v_x\,\mathrm{d}v_x}{n_f}, \quad \text{and} \quad \theta_f = \frac{1}{3n_f}\int_{\mathbb{R}} \left[\, g_1(v_x - u_f)^2 + g_2 \,\right] \mathrm{d}v_x. \tag{2.22}$$

The conservation properties of (2.7) are preserved; namely,

$$\int_{\mathbb{R}} \mathcal{C}_1(\tilde{g}_1; u_f, \theta_f)\,\mathrm{d}v_x = \int_{\mathbb{R}} \mathcal{C}_1(\tilde{g}_1; u_f, \theta_f) v_x\,\mathrm{d}v_x = 0, \tag{2.23a}$$

$$\frac{1}{2}\int_{\mathbb{R}} \mathcal{C}_1(\tilde{g}_1; u_f, \theta_f) v_x^2 + \mathcal{C}_1(\tilde{g}_2; u_f, \theta_f) + (4\theta_f\tilde{g}_1 - 2\tilde{g}_2)\,\mathrm{d}v_x = 0, \tag{2.23b}$$

for any $\tilde{g}_1$ and $\tilde{g}_2$ such that the fluid variables n_f, u_f, and θ_f are built via (2.22) using $\tilde{g}_1$ and $\tilde{g}_2$.

Though not required for (2.20), we will, for diagnostic purposes in Section 5, also consider the function

$$g_3(x,v_x) = \int_{\mathbb{R}^2} (v_y^4 + v_z^4) f(x,\mathbf{v})\,\mathrm{d}v_y\,\mathrm{d}v_z, \tag{2.24}$$

which satisfies

$$\partial_t g_3 + v_x \partial_x g_3 + E\partial_{v_x} g_3 = \nu\,\mathcal{C}_1(g_3; u_f, \theta_f) + \nu\,(\,12\theta_f g_2 - 4g_3\,). \tag{2.25}$$

3. Notation and the Discontinuous Galerkin Method

3.1. Notation

Let $\ell_x \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$, $\Omega_x = (-L_x, L_x)$ be an interval in physical space, and $\mathcal{T}_{x,\ell_x}$ be a uniform mesh on Ω_x with 2^{ℓ_x} elements. Let $\mathcal{E}_{x,\ell_x}$ be the skeleton of $\mathcal{T}_{x,\ell_x}$.

Similarly, let $\ell_v \in \mathbb{N}_0$, $\Omega_v = (-L_v, L_v)^3 \subset \mathbb{R}^3$, and $\mathcal{T}_{v,\ell_v}$ be a uniform cubic mesh on Ω_v with 2^{ℓ_v} elements in each dimension. Let $\mathcal{E}^{\mathrm{I}}_{v,\ell_v}$ be the interior (i.e., not including boundaries) skeleton on this mesh. We will often use $\langle \cdot \rangle_v$ and $\langle \cdot \rangle_{v_y,v_z}$ to denote integration in $\mathrm{d}\mathbf{v}$ and $\mathrm{d}v_y\,\mathrm{d}v_z$, respectively.

We let $\Omega = \Omega_x \times \Omega_v \subset \mathbb{R}^4$, and denote $L^2(\Omega)$ and $H^s(\Omega)$ to be the standard Lebesgue and Sobolev spaces on Ω. Let $(\cdot\,,\cdot)$ be the $L^2(\Omega)$-inner product with norm $\|\cdot\|_{L^2(\Omega)}$ and let $\|\cdot\|_{H^s(\Omega)}$ be the norm on $H^s(\Omega)$. We denote by $L^2(D)$ and $(\cdot\,,\cdot)_D$ the L^2 space with standard inner product on some domain D which is typically Ω_x or Ω_v. Any of the inner products mentioned above can be trivially extended to vector-valued functions with the standard Euclidean inner product.

Denote the discontinuous Galerkin finite element spaces $V_{x,\ell_x} \subset L^2(\Omega_x)$ and $V_{v,\ell_v} \subset L^2(\Omega_v)$ by

$$\begin{aligned} V_{x,\ell_x} &= \{g \in L^2(\Omega_x) : g\big|_K = \mathbb{Q}_k(K)\ \forall K \in \mathcal{T}_{x,\ell_x}\} \\ V_{v,\ell_v} &= \{g \in L^2(\Omega_v) : g\big|_K = \mathbb{Q}_k(K)\ \forall K \in \mathcal{T}_{v,\ell_v}\} \end{aligned} \tag{3.1}$$

where $\mathbb{Q}_k(K)$ is the set of all polynomials of maximum degree k in any direction on K. We assume $k = 2$ unless written otherwise. Let $\mathcal{V}_\ell = V_{x,\ell_x} \otimes V_{v,\ell_v}$.

Given $x_* \in \mathcal{E}_{x,\ell_x}$, let g be a function with traces $g^\pm(x_*) := \lim_{x \to x_*^\pm} g(x)$ well defined. Define the average and jump of g in x, respectively, by

$$\{\!\{g\}\!\} = \tfrac{1}{2}(g^+ + g^-) \qquad \text{and} \qquad [\![g]\!] = g^- - g^+. \tag{3.2}$$

We account for the periodic boundary in $\mathcal{E}_{x,\ell_x}$ by defining the jumps and averages on the boundary using (3.2) with $g^+ = g(L_x)$ and $g^- = g(-L_x)$. We denote by S_{x,ℓ_x} the intersection of V_{x,ℓ_x} with $k = 1$ and all continuous and periodic functions on the closure of Ω_x, i.e. $\overline{\Omega_x}$. The space S_{x,ℓ_x} is used for the discretization of (2.16) and uses linear functions so that the electric field E is constant on each element.

Similarly, consider the edge $e \in \mathcal{E}^{\mathrm{I}}_{v,\ell_v}$, where $e = \partial K^+ \cap \partial K^-$ and $K^\pm \in \mathcal{T}_{v,\ell_v}$ with normal outward vector $\mathbf{n}_v^\pm$. Given a scalar and vector valued function g and $\boldsymbol{\sigma}$ respectively with well defined traces on $\partial K^\pm$, define the average and jump of g and $\boldsymbol{\sigma}$ in $\mathbf{v}$, respectively, by

$$\begin{aligned} \{\!\{g\}\!\} &= \tfrac{1}{2}(g^+ + g^-) \quad \text{and} \quad [\![g]\!] = g^-\mathbf{n}_v^- + g^+\mathbf{n}_v^+, \\ \{\!\{\boldsymbol{\sigma}\}\!\} &= \tfrac{1}{2}(\boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^-) \quad \text{and} \quad [\![\boldsymbol{\sigma}]\!] = \boldsymbol{\sigma}^- \cdot \mathbf{n}_v^- + \boldsymbol{\sigma}^+ \cdot \mathbf{n}_v^+, \end{aligned} \tag{3.3}$$

where for any $\mathbf{v}_* \in e$,

$$g^\pm(\mathbf{v}_*) = \lim_{\substack{\mathbf{v} \to \mathbf{v}_* \\ \mathbf{v} \in K^\pm}} g(\mathbf{v}) \tag{3.4}$$

with analogous definition for $\boldsymbol{\sigma}^\pm$. While the same notation for average and jumps is used in the physical and velocity domains, the domain of integration of the DG formulation provides context to which case is used (see (3.6)). Let $\langle\!\langle \cdot\,,\cdot \rangle\!\rangle_e$ be the L^2 inner product over an edge e and denote $\langle\!\langle \cdot\,,\cdot \rangle\!\rangle_{\mathcal{E}_{x,\ell_x}} = \sum_{e \in \mathcal{E}_{x,\ell_x}} \langle\!\langle \cdot,\cdot \rangle\!\rangle_e$ with an analogous definition for $\langle\!\langle \cdot \rangle\!\rangle_{\mathcal{E}^{\mathrm{I}}_{v,\ell_v}}$. For functions g in $\mathcal{V}_\ell$, let ∂_x and $\nabla_{\mathbf{v}}$ represent the piece-wise spatial derivative and velocity gradient g.

Finally, for time integration, let $\Delta t > 0$ be the timestep, assumed for our purposes to be uniform. For $\mathfrak{n} \in \mathbb{N}_0$ define $t^{\mathfrak{n}} = \mathfrak{n}\Delta t$ and denote $f^{\mathfrak{n}}$ to be an approximation to $f(t^{\mathfrak{n}})$.

3.2. Discontinuous Galerkin Method

We first discretize (2.1) in phase space on $\mathcal{V}_\ell$ by the following semi-discrete problem: Find $f_h \in C([0,\infty]; \mathcal{V}_\ell)$ such that

$$(\partial_t f_h, g_h) + \mathcal{A}_{\mathrm{VP}}(f_h, g_h) = \nu \mathcal{A}_{\mathrm{LB}}(f_h, \boldsymbol{\rho}_{f_h}, g_h) \tag{3.5}$$

holds for all $g_h \in \mathcal{V}_\ell$. The Vlasov–Poisson portion, $\mathcal{A}_{\mathrm{VP}}$, is discretized with upwind fluxes; specifically,

$$\begin{aligned} \mathcal{A}_{\mathrm{VP}}(w_h, g_h) = &-(v_x w_h, \partial_x g_h) + \left\langle\!\!\left\langle v_x \{\!\{ w_h \}\!\} + \tfrac{|v_x|}{2} [\![w_h]\!], [\![g_h]\!] \right\rangle\!\!\right\rangle_{\mathcal{E}_{x,\ell_x} \times \Omega_v} \\ &- (\tilde{\boldsymbol{E}}_h w_h, \nabla_{\mathbf{v}} g_h) + \left\langle\!\!\left\langle \{\!\{ \tilde{\boldsymbol{E}}_h w_h \}\!\} + \tfrac{|\tilde{\boldsymbol{E}}_h \cdot \mathbf{n}_v|}{2} [\![w_h]\!], [\![g_h]\!] \right\rangle\!\!\right\rangle_{\Omega_x \times \mathcal{E}^{\mathrm{I}}_{v,\ell_v}} \end{aligned} \tag{3.6}$$

for all $w_h, g_h \in \mathcal{V}_\ell$ where $\tilde{\boldsymbol{E}}_h := (E_h, 0, 0)^\top$ and E_h is given by $-\partial_x \Phi_h$ where $\Phi_h \in S_{x,\ell_x}$ satisfies

$$(\partial_x \Phi_h, \partial_x q_h)_{\Omega_x} = \left(\langle w_h \rangle_v - n_e, q_h \right)_{\Omega_x} \tag{3.7}$$

for every $q_h \in S_{x,\ell_x}$. The boundary conditions are periodic in x and we impose zero fluxes on the velocity boundaries.

The Lenard–Bernstein portion, $\mathcal{A}_{\mathrm{LB}}$, of Equation (3.5) is discretized with the LDG method (e.g., [24]), with central fluxes for the diffusion term and a local Lax–Friedrichs flux for the advection term; namely,

$$\begin{aligned} \mathcal{A}_{\mathrm{LB}}(w_h, \boldsymbol{\rho}_h, g_h) = &-((\mathbf{v} - \mathbf{u}) w_h, \partial_v g_h) + \left\langle\!\!\left\langle \{\!\{ \mathbf{v} w_h \}\!\} - \tfrac{|\mathbf{v} \cdot \mathbf{n}_v|}{2} [\![w_h]\!], [\![g_h]\!] \right\rangle\!\!\right\rangle_{\Omega_x \times \mathcal{E}^{\mathrm{I}}_{v,\ell_v}} \\ &- (\boldsymbol{\sigma}_h, \nabla_{\mathbf{v}} g_h) + \left\langle\!\!\left\langle \{\!\{ \boldsymbol{\sigma}_h \}\!\}, [\![g_h]\!] \right\rangle\!\!\right\rangle_{\Omega_x \times \mathcal{E}^{\mathrm{I}}_{v,\ell_v}} \end{aligned} \tag{3.8}$$

for every $w_h, g_h \in \mathcal{V}_\ell$, where $\mathbf{u}$ is determined from $\boldsymbol{\rho}_h \in [V_{x,\ell_x}]^3$ via formulas in (2.5). Here $\boldsymbol{\sigma}_h \in [\mathcal{V}_\ell]^3$ is the approximation to the velocity gradient of w_h and is defined by

$$(\boldsymbol{\sigma}_h, \tau_h) = (\theta \nabla_{\mathbf{v}} w_h, \tau_h) - \left\langle\!\!\left\langle \theta [\![w_h]\!], \{\!\{ \tau_h \}\!\} \right\rangle\!\!\right\rangle_{\Omega_x \times \mathcal{E}^{\mathrm{I}}_{v,\ell_v}} \tag{3.9}$$

for every $\tau_h \in [\mathcal{V}_\ell]^3$, where θ is determined by $\boldsymbol{\rho}_h$ the relevant formula in (2.5).

If $w_h = 0$ on $\partial\Omega_v$, then it can be shown that $\mathcal{A}_{\mathrm{LB}}(w_h, \boldsymbol{\rho}_{w_h}, \mathbf{e} q_h) = 0$ for all $q_h \in V_{x,\ell_x}$, which implies that the conservation properties in Equation (2.7) hold.

For brevity, we do not provide the discretization for the Chu reduction (2.20), but we note it is similar to the discretizations given above for the slab problem.

3.3. Time Stepping Method

We discretize (3.5) in time via Implicit-Explicit (IMEX) Runge–Kutta (RK) methods [6]. Such methods are popular time steppers for evolving kinetic models that feature multiple time scales [64, 23, 29]. In our case, the Vlasov–Poisson portion $\mathcal{A}_{\mathrm{VP}}$ will be evolved explicitly and the collision operator $\mathcal{A}_{\mathrm{LB}}$ will be evolved implicitly. We will use IMEX-RK method of [23] which is given by:

$$(f_h^{(1,*)}, g_h) = (f_h^{\mathfrak{n}}, g_h) - \Delta t \mathcal{A}_{\mathrm{VP}}(f_h^{\mathfrak{n}}, g_h), \tag{3.10a}$$

$$(f_h^{(1)}, g_h) = (f_h^{(1,*)}, g_h) + \Delta t \nu \mathcal{A}_{\mathrm{LB}}(f_h^{(1)}, \boldsymbol{\rho}_{f_h^{(1*)}}, g_h), \tag{3.10b}$$

$$(f_h^{(2,*)}, g_h) = \tfrac{1}{2}(f_h^{\mathfrak{n}}, g_h) + \tfrac{1}{2}\left((f_h^{(1)}, g_h) - \Delta t \mathcal{A}_{\mathrm{VP}}(f_h^{(1)}, g_h)\right), \tag{3.10c}$$

$$(f_h^{(2)}, g_h) = (f_h^{(2,*)}, g_h) + \tfrac{1}{2}\Delta t \nu \mathcal{A}_{\mathrm{LB}}(f_h^{(2)}, \boldsymbol{\rho}_{f_h^{(2*)}}, g_h), \tag{3.10d}$$

and $f_h^{\mathfrak{n}+1} := f_h^{(2)}$

Assuming zero velocity-boundary data, the invariance of the discrete collision operator implies $\boldsymbol{\rho}_{f_h^{(s,*)}} = \boldsymbol{\rho}_{f_h^{(s)}}$, for $s \in \{1, 2\}$. Therefore we plug the moments $f_h^{(s,*)}$ into the collision operator $\mathcal{A}_{\mathrm{LB}}$ in (3.10b) and (3.10d) for $s = 1$ and $s = 2$ respectively. This decouples the moments from the distribution and provides a linear solve for $f_h^{(s)}$. Both (3.10b) and (3.10d) are solved iteratively using GMRES with the possible inclusion of a block-Jacobi preconditioner.

4. Sparse-grid Method

In this section, we describe the sparse-grid DG method and adaptivity procedure used in ASGarD. The method, first introduced in [84] (and from which some of the presentation of this section is based), is provided here for completeness. We first construct the multiwavelet basis in one dimension, then extend to multiple dimensions and introduce the sparse-grid selection rule, and finally discuss the adaptivity procedure.

4.1. Single Dimension Multiwavelet Basis

The one-dimensional multiwavelet basis is a hierarchical basis in which additional basis functions for resolving fine scale features are introduced using orthogonal complements to current functions in the basis. To simplify the presentation, we assume a domain $\Omega = [0, 1]$. Given a level $\ell \in \{0, \dots, N\}$, let $\mathcal{T}_\ell$ be a uniform mesh of Ω with mesh size $h_\ell = 2^{-\ell}$. The partition of $\mathcal{T}_\ell$ is characterized by the union of disjoint intervals $I_{\ell,j} := (2^{-\ell} j, 2^{-\ell}(j+1))$ for $j = 0, \dots, 2^\ell - 1$. Given this mesh, define the corresponding DG finite element space V_ℓ by[1]

$$V_\ell := V_\ell^k = \left\{ g \in L^2(\Omega) : g\big|_{I_{\ell,j}} \in \mathbb{P}_k(I_{\ell,j}) \ \forall j = 0, \dots, 2^\ell - 1 \right\}, \tag{4.1}$$

where $\mathbb{P}_k$ is the space of polynomials of degree up to k. This space has dimension $\dim(V_\ell) = 2^\ell(k+1)$. Additionally, due to the uniform partitioning,

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_N. \tag{4.2}$$

Let W_ℓ to be the orthogonal complement of $V_{\ell-1}$ in V_ℓ with respect to the $L^2(\Omega)$ inner product; that is, $W_0 = V_0$, while for $\ell \geq 1$,

$$V_\ell = V_{\ell-1} \oplus W_\ell \qquad \text{and} \qquad W_\ell \perp V_{\ell-1}, \tag{4.3}$$

where $\oplus$ is the direct sum and $\dim(W_\ell) = \max\{0, 2^{\ell-1}(k+1)\}$. Then

$$V_N = \bigoplus_{\ell=0}^{N} W_\ell. \tag{4.4}$$

The hierarchical decomposition in (4.4) induces a natural decay in the coefficients for the approximation of smooth functions. Specifically, let $Q_\ell : L^2(\Omega) \to W_\ell$ be the orthogonal L^2 projection onto W_ℓ. Then by standard polynomial approximation theory (see, e.g., [14, Section 5.4.2] or [71, Theorem 2.6]), there exists a constant $C > 0$, independent of ℓ, such that for any $g \in H^s(\Omega)$,

$$\|Q_\ell g\|_{L^2(\Omega)} \leq C h_\ell^{\min\{s, k+1\}} \|g\|_{H^s(\Omega)}. \tag{4.5}$$

This decay property motivates the adaptive strategy described in Section 4.4.

A standard choice for the basis of W_ℓ for $\ell \geq 1$ are *wavelets* – functions that are scaled and shifted to capture finer-scale features. The prototype wavelet is the piece-wise constant Haar basis [42]. Here we use Alpert wavelets [4].

Definition 1. *The Alpert wavelets are a set of a functions $\{\phi_i(y) : i = 1, \dots, k+1\} \subset L^2(\mathbb{R})$ with support in $[-1, 1]$ and defined such that*

1. $\phi_i\big|_{(0,1)} \in \mathbb{P}_k(0, 1)$.
2. $\phi_i(y) = (-1)^{i+k} \phi_i(-y)$.
3. $\int_{-1}^{1} \phi_i(y) y^j \, \mathrm{d}y = 0$ *for all* $j = 0, 1, \dots, i + k - 1$.
4. $\int_{-1}^{1} \phi_i(y) \phi_j(y) \, \mathrm{d}y = \delta_{ij}$ *for all* $i, j = 1, \dots, k$ *where* δ_{ij} *is the Kronecker delta.*

[1] We will often drop the polynomial degree superscript on V_ℓ^k for brevity.

For a given polynomial degree k, the Alpert wavelets satisfying Definition 1 are unique up to a sign. The Alpert basis is not hierarchical in the polynomial degree; thus each wavelet must be reconstructed when k is changed. For $k = 0$, Alpert's wavelets correspond to the Haar basis. For $k = 2$, the wavelets are given on the interval $(0, 1)$ by

$$\phi_1(y) = \tfrac{1}{3}\sqrt{\tfrac{1}{2}}(1 - 24y + 30y^2), \quad \phi_2(y) = \tfrac{1}{2}\sqrt{\tfrac{3}{2}}(3 - 16y + 15y^2), \quad \phi_3(y) = \tfrac{1}{3}\sqrt{\tfrac{5}{2}}(4 - 15y + 12y^2) \tag{4.6}$$

Construction of the wavelets and examples for other polynomial degrees can be found in [4, Page 5].

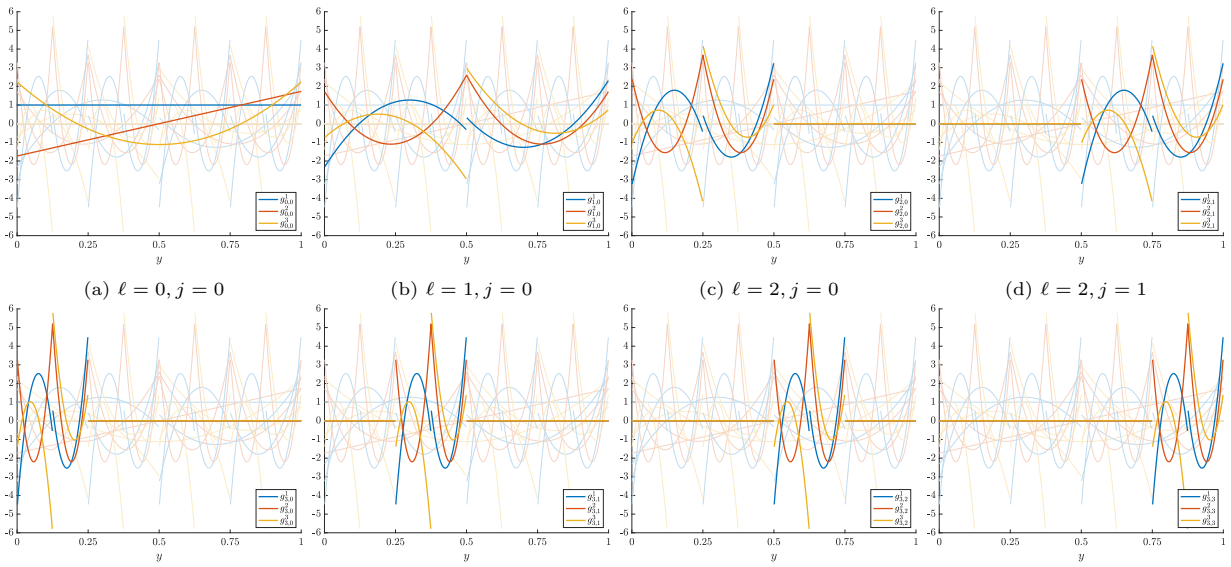
For each $\ell \geq 0$, we use the Alpert wavelets to define a basis set $\{g^i_{\ell,j}\}$ of W_ℓ. For $\ell = 0$, we choose $g^i_{0,0}$ to be the shifted Legendre polynomials normalized on $L^2(\Omega)$. For $\ell \geq 1$, we shift and rescale the Alpert wavelets so that for each $x \in (0, 1)$,

$$g^i_{\ell,j}(y) = 2^{(\ell-1)/2}\gamma_i(2^{\ell-1}y - j), \quad \text{where} \quad \gamma_i(y) := \sqrt{2}\phi_i(2y - 1). \tag{4.7}$$

Here ℓ is the level, $j = 0, \ldots, 2^{\ell-1} - 1$ is the level index, and $i = 1, \ldots, k + 1$ is the polynomial index. The support of $g^i_{\ell,j}$ is precisely $I_{\ell-1,\lfloor j/2 \rfloor}$, where $\lfloor \cdot \rfloor$ is the floor function. Additionally, since every wavelet $g^{i'}_{\ell',j'}$ for any i', j', and $\ell' < \ell$ is a polynomial on $I_{\ell-1,\lfloor j/2 \rfloor}$, Item 3 of Definition 1 ensures that the wavelet bases are all orthonormal; that is,

$$\int_0^1 g^i_{\ell,j}(y) g^{i'}_{\ell',j'}(y)\, \mathrm{d}y = \delta_{ii'}\delta_{\ell\ell'}\delta_{jj'}. \tag{4.8}$$

Plots of the wavelets $g^i_{\ell,j}$ for $\ell = 0, 1, 2, 3$ and $k = 2$ are given in Figure 4.1.1.



(a) $\ell = 0, j = 0$ (b) $\ell = 1, j = 0$ (c) $\ell = 2, j = 0$ (d) $\ell = 2, j = 1$

(e) $\ell = 3, j = 0$ (f) $\ell = 3, j = 1$ (g) $\ell = 3, j = 2$ (h) $\ell = 3, j = 3$

Figure 4.1.1: Plots of the wavelet basis $g^i_{\ell,j}$, given by (4.7), for $k = 2$. In each plot, the entire set of wavelet basis functions for level $\ell = 3$ and lower are shown in each plot and are translucent.

4.2. Multiwavelet Basis in Higher Dimensions

A d-dimensional basis is achieved through a tensor product extension. Let $\Omega^d = (0, 1)^d$ with $\boldsymbol{y} = (y_1, \ldots, y_d) \in \Omega^d$. Given a multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$, define the norms

$$|\boldsymbol{\alpha}|_1 = \sum_{m=1}^{d} \alpha_m \qquad \text{and} \qquad |\boldsymbol{\alpha}|_\infty = \max_{1 \leq m \leq d} \alpha_m. \tag{4.9}$$

Let $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_d)$ be a multi-index level set, where ℓ_d defines the level for dimension d, and let $\mathcal{T}_{\boldsymbol{\ell}}$ be a tensor product mesh with multi-dimensional mesh parameter $\boldsymbol{h} := (2^{-\ell_1}, \ldots, 2^{-\ell_d})$. We label all elements in $\mathcal{T}_{\boldsymbol{\ell}}$ by $I_{\boldsymbol{\ell},\boldsymbol{j}} = \{\boldsymbol{y} : y_m \in (2^{-\ell_m} j_m, 2^{-\ell_m}(j_m + 1)\}$ and define the tensor product finite element space by

$$\boldsymbol{V}_{\boldsymbol{\ell}} := \boldsymbol{V}_{\boldsymbol{\ell}}^k = \{g \in L^2(\Omega) : g\big|_{I_{\boldsymbol{\ell},\boldsymbol{j}}} \in \mathbb{Q}_k(I_{\boldsymbol{\ell},\boldsymbol{j}}),\ \forall\ 0 \le j_m \le 2^{\ell_m} - 1, m = 1, \ldots, d\}, \tag{4.10}$$

where $\mathbb{Q}_k(I_{\boldsymbol{\ell},\boldsymbol{j}})$ represents the set of polynomials of degree up to k in each dimension on $I_{\boldsymbol{\ell},\boldsymbol{j}}$. If $\boldsymbol{\ell} = (N, \ldots, N)$, then we abbreviate $\boldsymbol{V}_{\boldsymbol{\ell}}$ by $\boldsymbol{V}_N$.

Recall the one-dimensional hierarchical decomposition in Section 4.1. Given the complementary sets W_{ℓ_m} defined in (4.3), let

$$\boldsymbol{W}_{\boldsymbol{\ell}} = W_{\ell_1} \otimes W_{\ell_2} \otimes \cdots \otimes W_{\ell_d}. \tag{4.11}$$

Then (4.4) extends to the multidimensional setting:

$$\boldsymbol{V}_{\boldsymbol{\ell}} = V_{\ell_1} \otimes \cdots \otimes V_{\ell_d} = \bigoplus_{\boldsymbol{0} \le \boldsymbol{\ell}' \le \boldsymbol{\ell}} \boldsymbol{W}_{\boldsymbol{\ell}'}. \tag{4.12}$$

An extension of the coefficient decay result (4.5) also holds. Let $\boldsymbol{Q}_{\boldsymbol{\ell}} : L^2(\Omega) \to \boldsymbol{W}_{\boldsymbol{\ell}}$ be the orthogonal L^2 projection onto $\boldsymbol{W}_{\boldsymbol{\ell}}$, then

$$\|\boldsymbol{Q}_{\boldsymbol{\ell}} g\|_{L^2(\Omega)} = \mathcal{O}\left(\prod_{m=1}^{d} h_m^{\min\{s,k+1\}}\right), \tag{4.13}$$

where s is a regularity parameter tied to a Sobolev-like space including high-order mixed derivative control. We refer the reader to [39, (A.8)] and [73, Proposition 5.1] for specifics on (4.13).

The basis we choose for $\boldsymbol{W}_{\boldsymbol{\ell}}$ are the *multiwavelets* which are products of the 1D wavelets in (4.7):

$$g_{\boldsymbol{\ell},\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{y}) := \prod_{m=1}^{d} g_{\ell_m, j_m}^{i_m}(y_m), \text{ where } j_m = 0, \ldots, \max\{0, 2^{\ell_m - 1} - 1\}, i_m = 1, \ldots, k+1. \tag{4.14}$$

It follows from repeated application of (4.8) in each dimension that these multiwavelets are orthonormal in $L^2(\Omega)$.

4.3. The Sparse-grid Selection Rule

The spaces $\boldsymbol{W}_{\ell}$ are used to define the sparse grid. From (4.12) we can rewrite the full-grid as

$$\boldsymbol{V}_N = \bigoplus_{|\boldsymbol{\ell}|_\infty \le N} \boldsymbol{W}_{\boldsymbol{\ell}}. \tag{4.15}$$

This space has dimension $\dim(\boldsymbol{V}_N) = (k+1)^d 2^{Nd}$. The sparse grid is defined via a selection rule that relaxes the index norm in (4.15).

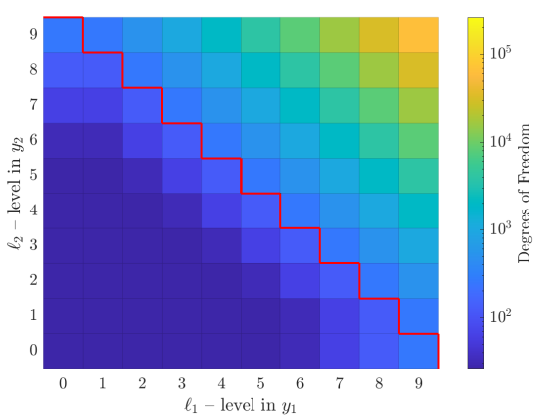
Definition 2 ([84, 13])**.** *The level N sparse grid, $\hat{\boldsymbol{V}}_N \subseteq \boldsymbol{V}_N$, is defined by*

$$\hat{\boldsymbol{V}}_N = \bigoplus_{|\boldsymbol{\ell}|_1 \le N} \boldsymbol{W}_{\boldsymbol{\ell}}. \tag{4.16}$$

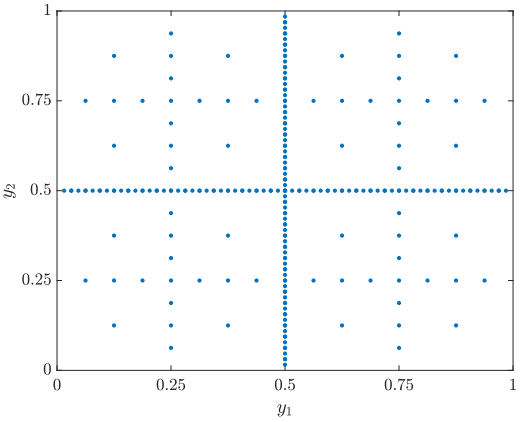
By definition, the sparse-grid only includes components $\boldsymbol{W}_{\boldsymbol{\ell}}$ whose level indices $\boldsymbol{\ell}$ sum up to N, and throws away basis functions deemed too fine to include in multiple dimensions. It was shown in [84, Lemma 2.3] that

$$\dim(\hat{\boldsymbol{V}}_N) = \Theta((k+1)^d 2^N N^{d-1}), \tag{4.17}$$

which avoids the costly $\mathcal{O}(2^{Nd})$ scaling of the full-grid in (4.15) but still maintains exponential dependence on k and on $\log(N)$.



(a) A heat map for the degrees of freedom of $\boldsymbol{W}_{\boldsymbol{\ell}}$ for a 2 dimensional problem. The whole rectangle corresponds to all degrees of freedom for the full-grid $\boldsymbol{V}_9$ while the sparse-grid $\boldsymbol{V}_9$ only contains the spaces on the lower-left portion divided by the red line.



(b) Plot showing the coverage of sparse-grids in two dimensions. Each point represents the barycenter of the support of a wavelet that is in the level 7 sparse-grid.

Figure 4.3.1: Sparse-grid illustrations.

Figure 4.3.1 illustrates which basis functions are kept in the sparse-grid and the reduction in degrees of freedom that sparse-grids provide for the case with $d = 2$. Figure 4.3.1a shows that the dimension of the spaces $\boldsymbol{W}_{\boldsymbol{\ell}}$ being thrown away in the sparse-grid truncation are significantly larger on average than the dimension of the spaces that are kept. As a result, there is a reduction in degrees of freedom from the full-grid space $\boldsymbol{V}_9$ of size $2^{18} \approx 2.62 \times 10^5$ to the sparse grid space $\hat{\boldsymbol{V}}_9$ of size 2816.[2] Figure 4.3.1b shows that the basis functions kept in the sparse-grid allow accurate approximations of derivatives in coordinate directions while throwing away mixed-derivative data which is assumed to be smaller than the components kept by the sparse-grid. It has been shown that $\hat{\boldsymbol{V}}_N$ shares similar approximation properties to $\boldsymbol{V}_N$ in L^2, which is $\mathcal{O}(h^{k+1})$, up to a poly-logarithmic factor of $|\log_2 h|^{d-1}$ (see [84, Theorem 2.4]). This result holds for functions with bounded mixed derivatives of sufficient order.

4.4. Adaptive Sparse-grids

The adaptive sparse-grid method uses an adaptive algorithm based on the hierarchical framework of the sparse-grid method [40]. The first step is to further decompose the orthogonal complements $\boldsymbol{W}_{\boldsymbol{\ell}}$ by their level $\boldsymbol{\ell}$ and position $\boldsymbol{j}$ within the level. This position $\boldsymbol{j}$ in the level is based on the multiwavelet basis. Given the basis in (4.14), we define the space $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{W}_{\boldsymbol{\ell}}$, called a *hierarchical element*, by

$$\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} = \operatorname*{span}_{\substack{1 \le i_m \le k+1 \\ 1 \le m \le d}} \{g^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}}\}. \tag{4.18}$$

This space has dimension $\dim(\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}) = (k+1)^d$ and

$$\boldsymbol{W}_{\boldsymbol{\ell}} = \bigoplus_{\boldsymbol{j} \in \mathcal{B}_{\boldsymbol{\ell}}} \boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \tag{4.19}$$

where

$$\mathcal{B}_{\boldsymbol{\ell}} := \{\boldsymbol{j} = (j_1, \ldots, j_d) : j_m = 0, \ldots, \max\{0, 2^{\ell_m - 1} - 1\}, \forall m = 1, \ldots, d\}. \tag{4.20}$$

[2] Here we use $k = 0$ to calculate $\dim(\boldsymbol{V}_9)$ and $\dim(\hat{\boldsymbol{V}}_9)$.

The spaces $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ are deemed hierarchical because they carry a natural parent-child relationship of which the details will be postponed (see Definition 4). The full- and sparse-grid spaces, (4.15) and (4.16) respectively, can be written as

$$\boldsymbol{V}_N = \bigoplus_{\substack{|\boldsymbol{\ell}|_\infty \le N \\ \boldsymbol{j}\in\mathcal{B}_{\boldsymbol{\ell}}}} \boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \qquad \text{and} \qquad \hat{\boldsymbol{V}}_N = \bigoplus_{\substack{|\boldsymbol{\ell}|_1 \le N \\ \boldsymbol{j}\in\mathcal{B}_{\boldsymbol{\ell}}}} \boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}. \tag{4.21}$$

For the adaptive sparse-grid algorithm, it is helpful to view the full- and sparse-grid spaces as direct sums of the hierarchical elements $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$. We can now define an adaptive sparse-grid which is an arbitrary collection of hierarchical elements.

Definition 3. *Given a max level $N_{max} \in \mathbb{N}_0$ and a level index set $\{(\boldsymbol{\ell}^\iota, \boldsymbol{j}^\iota)\}_{\iota=1}^M$ such that for all $\iota = 1, \dots, M$, $|\boldsymbol{\ell}^\iota|_\infty \le N_{max}$ and $\boldsymbol{j}^\iota \in \mathcal{B}_{\boldsymbol{\ell}}$, the adaptive sparse-grid $\boldsymbol{V} \subseteq \boldsymbol{V}_{N_{max}}$ is defined as*

$$\boldsymbol{V} = \bigoplus_{\iota} \boldsymbol{W}_{\boldsymbol{\ell}^\iota,\boldsymbol{j}^\iota}. \tag{4.22}$$

Here M is said to be the number of active elements of the adaptive sparse-grid $\boldsymbol{V}$.

We will often drop the ι superscript in (4.22) and refer to the level index set as $\{(\boldsymbol{\ell},\boldsymbol{j})\}$. From (4.21), the standard sparse-grid is a specific adaptive sparse-grid where we include all hierarchical elements $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ such that $|\boldsymbol{\ell}|_1 \le N_{\max}$ and $\boldsymbol{j} \in \mathcal{B}_{\boldsymbol{\ell}}$.

4.4.1. Adaptive Approximation of Initial Data

Let $\mathcal{P}_{\boldsymbol{V}}$ be the L^2 projection from $L^2(\Omega)$ onto $\boldsymbol{V}$. The main idea of the adaptive sparse-grid is to choose a grid $\boldsymbol{V} \subseteq \boldsymbol{V}_{N_{\max}}$, depending on the distribution w, such that

1. The relative projection error $\|w - \mathcal{P}_{\boldsymbol{V}} w\|_{L^2(\Omega)} / \|w\|_{L^2(\Omega)}$ is small;
2. $\dim(\boldsymbol{V})$ is approximately minimal.

We will first demonstrate this process for an initial condition, and then extend the result to functions formulated via a dynamical system.

For a fixed max level $N_{\max}$, choosing $\boldsymbol{V} = \boldsymbol{V}_{N_{\max}}$ would minimize the L^2 projection error over all possible adaptive sparse-grid spaces, but with significant costs in terms of the number of degrees of freedom. Thus we assume $w \in \boldsymbol{V}_{N_{\max}}$ is our target; then the coefficient expansion with respect to the multiwavelet basis of (4.14) is given by

$$w(\boldsymbol{y}) = \sum_{\left\{\substack{(\boldsymbol{\ell},\boldsymbol{j}): \\ |\boldsymbol{\ell}|_\infty \le N_{\max}, \boldsymbol{j}\in\mathcal{B}_{\boldsymbol{\ell}}}\right\}} \sum_{\substack{1\le i_m \le k+1 \\ 1\le m\le d}} w^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}} g^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}}(\boldsymbol{y}) \quad \text{where} \quad w^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}} = \int_\Omega w(\boldsymbol{y}) g^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}}(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}. \tag{4.23}$$

For simplification, we define $w_{\boldsymbol{\ell},\boldsymbol{j}}$ to be the multilinear rank-d tensor with $k+1$ entries in each dimension, defined by

$$[w_{\boldsymbol{\ell},\boldsymbol{j}}]_{\boldsymbol{i}} = w^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}}. \tag{4.24}$$

When taking the norm of $w_{\boldsymbol{\ell},\boldsymbol{j}}$, we first flatten the tensor into a vector in $\mathbb{R}^{(k+1)^d}$ and apply the appropriate vector norm in ℓ^p where $1 \le p \le \infty$. Then the L^2-norm of w can be written as

$$\|w\|^2_{L^2(\Omega)} = \sum_{\left\{\substack{(\boldsymbol{\ell},\boldsymbol{j}): \\ |\boldsymbol{\ell}|_\infty \le N_{\max}, \boldsymbol{j}\in\mathcal{B}_{\boldsymbol{\ell}}}\right\}} \|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2^2 \tag{4.25}$$

Additionally, for any adaptive sparse-grid space $\boldsymbol{V}$ with level index set $\{(\boldsymbol{\ell},\boldsymbol{j})\}$ we have

$$\mathcal{P}_{\boldsymbol{V}} w = \sum_{(\boldsymbol{\ell},\boldsymbol{j})} \sum_{\substack{1\le i_m \le k+1 \\ 1\le m\le d}} w^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}} g^{\boldsymbol{i}}_{\boldsymbol{\ell},\boldsymbol{j}} \quad \text{and} \quad \|\mathcal{P}_{\boldsymbol{V}} w\|^2_{L^2(\Omega)} = \sum_{(\boldsymbol{\ell},\boldsymbol{j})} \|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2^2. \tag{4.26}$$

From (4.23) and (4.26), it is clear that the relative projection error satisfies

$$\frac{\|w - \mathcal{P}_{\boldsymbol{V}} w\|^2_{L^2(\Omega)}}{\|w\|^2_{L^2(\Omega)}} = \sum_{(\boldsymbol{\ell},\boldsymbol{j}):\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \not\subseteq \boldsymbol{V}} \frac{\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2^2}{\|w\|^2_{L^2(\Omega)}}. \tag{4.27}$$

Therefore, given $\tau > 0$, called the *threshold*, we want to keep all hierarchical elements $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ such that

$$\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2 \geq \tau \|w\|_{L^2(\Omega)} \tag{4.28}$$

lest they contribute to the error in (4.27).

Refinement. We will now describe how hierarchical elements are added to the adaptive sparse-grid – which we call refinement. The primary challenge in building a grid that contains all elements satisfying (4.28) is to avoid checking all hierarchical elements in the full-grid – an operation that naively would require $\mathcal{O}(2^{Nd})$ operations.

The refinement process is iterative, where an initial grid is chosen and then added upon. For adapting an initial condition, we choose our initial grid as the sparse-grid $\boldsymbol{V} = \hat{\boldsymbol{V}}_{N_{\max}}$. Given a current grid $\boldsymbol{V}$, the coefficients $w_{\boldsymbol{\ell},\boldsymbol{j}}$ are computed for every hierarchical element in the grid. In order to determine what elements to add to the grid, we appeal to the hierarchical representation of the full-grid space which embeds the following parent-child relation.

Definition 4. *Let $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ for $(\boldsymbol{\ell},\boldsymbol{j}) = \big((\ell_1,\dots,\ell_d),(j_1,\dots,j_d)\big)$ be a hierarchical element with max level N_{max}. The children of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$, with up to two per dimension, are defined for each dimension $m = 1,\dots,d$ by the following:*

- *If $\ell_m = 0$, then $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$, where*

$$(\boldsymbol{\ell}',\boldsymbol{j}') = \big((\ell_1,\dots,\ell_{m-1},1,\ell_{m+1},\dots,\ell_d),(j_1,\dots,j_{m-1},0,j_{m+1},\dots,j_d)\big), \tag{4.29}$$

  is a child of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$.

- *If $0 < \ell_m < N_{max}$, then $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$, where*

$$(\boldsymbol{\ell}',\boldsymbol{j}') = \big((\ell_1,\dots,\ell_{m-1},\ell_m+1,\ell_{m+1},\dots,\ell_d),(j_1,\dots,j_{m-1},2j_m,j_{m+1},\dots,j_d)\big) \quad \textit{and} \tag{4.30a}$$

$$(\boldsymbol{\ell}',\boldsymbol{j}') = \big((\ell_1,\dots,\ell_{m-1},\ell_m+1,\ell_{m+1},\dots,\ell_d),(j_1,\dots,j_{m-1},2j_m+1,j_{m+1},\dots,j_d)\big), \tag{4.30b}$$

  are children of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$.

- *If $\ell_m = N_{max}$, then there are no children of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ in dimension m.*

The parents of an element $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ are all elements $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$ such that $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ is a child of $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$.

It is clear from Definition 4 that each hierarchical element can have up to $2d$ children and up to d parents. To tie Definition 4 to the wavelet representation, for a fixed dimension $m \in \{1,\dots,d\}$, the children of a wavelet given in (4.7) are the up to two wavelets of one greater level whose support is contained in the parent. Furthermore, based on the coefficient decay estimate (4.13), if w is sufficiently smooth, then it is reasonable to assume that if $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$ is a child of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$, then $\|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\| \leq \|w_{\boldsymbol{\ell},\boldsymbol{j}}\|$. Therefore, if the size of a hierarchical element in the grid is small, we assume the size of the children are also small, and we do not need to search further along this path. This assumption leads to a stopping mechanism for the refinement strategy: Given a grid $\boldsymbol{V}$ with level index set $\{(\boldsymbol{\ell},\boldsymbol{j})\}$, if

$$\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2 \geq \tau \Big(\sum_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_2^2 \Big)^{\frac{1}{2}} = \tau \|\mathcal{P}_{\boldsymbol{V}} w\|_{L^2(\Omega)}, \tag{4.31}$$

then we add all children of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ to the grid. We repeat this process iteratively until no new children are added.

Coarsening. The process of removing active elements from the current grid, i.e., coarsening, is achieved by simple thresholding of the coefficients. Let $0 < \mu < 1$ be the *coarsening factor*. For a given grid $\boldsymbol{V}$, if $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ is a hierarchical element such that

$$\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_2 \le \mu\tau\Big(\sum_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_2^2\Big)^{\frac{1}{2}} = \mu\tau\|\mathcal{P}_{\boldsymbol{V}} w\|_{L^2(\Omega)}, \tag{4.32}$$

then it is removed from the grid. We acknowledge this coarsening strategy does not preserve structural properties like parent completeness, i.e., requiring that all parents of an active element are active (see [79, Section 3]), but still yields stable and accurate approximations as evidenced in Section 5.

While the criteria for refinement (4.31) and coarsening (4.32) are based on ℓ^2-type norms, other discrete norms can be used. For instance, the ℓ^∞ norm can also be used:

$$\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_\infty \ge \tau \max_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_\infty \qquad \text{(for refinement)}, \tag{4.33a}$$

$$\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_\infty \le \mu\tau \max_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_\infty \qquad \text{(for coarsening)}. \tag{4.33b}$$

The first refinement criterion implemented in ASGarD was (4.33a). Currently both ℓ^∞ and ℓ^2 strategies are supported, but our experiments did not show a significant difference in the accuracy or the degrees of freedom. This is in contrast to the L^∞ refinement, which has been shown to yield a much denser grid compared to the L^2 criteria, e.g., see [40]. In the rest of this work, we will use (4.33) as our coarsening/refinement strategy.

4.4.2. Adaptive sparse-grids of a dynamical system

Unlike adapting initial conditions, where the coefficients are drawn from analytic or quadrature data, the adaptive strategy can also be utilized to create temporally varying grids that dynamically capture features of the solution in time. To extend our adaptive strategy to dynamical systems, consider the abstract problem

$$(\partial_t w, g) = \mathcal{A}(w, g) \quad \forall g \in \boldsymbol{V}_{N_{\max}} \tag{4.34}$$

where $\mathcal{A} : \boldsymbol{V}_{N_{\max}} \times \boldsymbol{V}_{N_{\max}} \to \mathbb{R}$ (c.f. (3.5)). Here $\mathcal{A}$ is one of the discretizations in (3.6) or (3.8). For a given adaptive sparse-grid $\boldsymbol{V} \subseteq \boldsymbol{V}_N$ define the operator $\mathcal{R} : \boldsymbol{V} \to \boldsymbol{V}$ by

$$(\mathcal{R}_{\boldsymbol{V}} w, g) = \mathcal{A}(w, g) \quad \forall g \in \boldsymbol{V}. \tag{4.35}$$

Then (4.34) can be succinctly written as $\partial_t w = \mathcal{R}_{\boldsymbol{V}_{N_{\max}}} w$.

Consider a solution w^{n} at timestep t^{n} defined on an adaptive sparse-grid $\boldsymbol{V}^{\mathrm{n}}$. To refine, we first set $\boldsymbol{V} = \boldsymbol{V}^{\mathrm{n}}$ and advance the abstract problem $\partial_t w = \mathcal{R}_{\boldsymbol{V}} w$ from t^{n} to $t^{\mathrm{n}+1}$ via a IMEX Runge–Kutta method (3.10) to produce $w^{\mathrm{n}+1} \in \boldsymbol{V}$. We then check for elements $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ of $\boldsymbol{V}$ that satisfy the same refinement requirement as the initial condition case, namely, (4.31) for a ℓ^2-norm refinement or (4.33a) for a ℓ^∞-norm refinement. If there are elements satisfying the refinement criterion, then their children are added to $\boldsymbol{V}$. We then go back to time t^{n} and advance $\partial_t w = \mathcal{R}_{\boldsymbol{V}} w$ from t^{n} to $t^{\mathrm{n}+1}$ with the updated space $\boldsymbol{V}$. Since $\boldsymbol{V}^{\mathrm{n}} \subseteq \boldsymbol{V}$, the coefficients of the state w^{n} can be extended into $\boldsymbol{V}$ by setting $w^{\mathrm{n}}_{\boldsymbol{\ell},\boldsymbol{j}} = 0$ if $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subseteq \boldsymbol{V}$ but not if $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \not\subseteq \boldsymbol{V}^{\mathrm{n}}$. This process is repeated until no new children are added into the grid $\boldsymbol{V}$ – in which case we set $\boldsymbol{V}^{\mathrm{n}+1} = \boldsymbol{V}$. Typically, only one or two refinements are needed per timestep, but more may be needed for the first few timesteps due to initial layers. Coarsening after refinement is done in a manner analogous to the initial condition case. The procedure for refining and coarsening are summed up in Algorithm 1 and Algorithm 2 respectively.

As visual illustration of the adaptive sparse-grid method is shown in Figure 4.4.1, where it is applied to $1x3v$ Riemann problem in Section 5.3. As seen in Figure 4.4.1b, the adaptive algorithm focuses on refinement around the discontinuity in the distribution, plotted in Figure 4.4.1a, while coarsening occurs near the velocity boundaries.

Algorithm 1: Adaptive refinement using ℓ^∞-norm

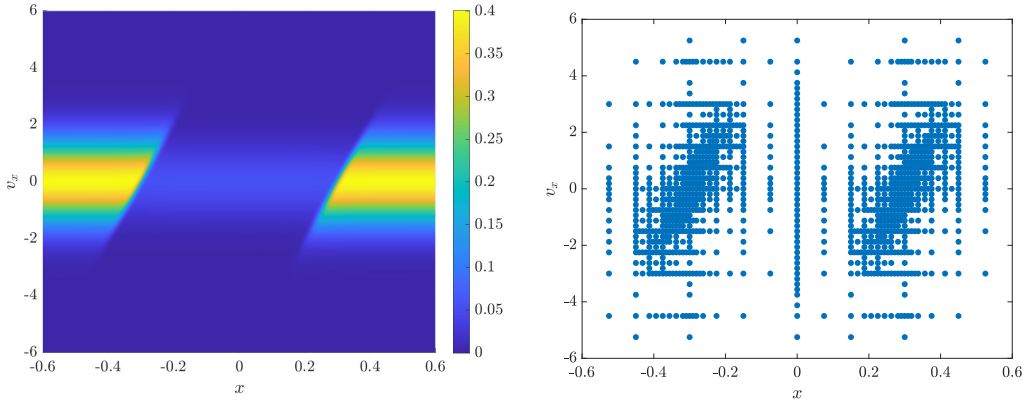
Input : Adaptive sparse-grid $\boldsymbol{V}$, threshold $\tau > 0$
Input : Distribution w or dynamical system $(w, \boldsymbol{U}) \to \partial_t w = \mathcal{R}_{\boldsymbol{U}} w$ defined in (4.35) with coefficients $w^{\mathrm{n}}_{\boldsymbol{\ell},\boldsymbol{j}}$ computed for all $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}$
Output: Adaptive sparse-grid $\boldsymbol{V}^*$
Output: Coefficients $w_{\boldsymbol{\ell},\boldsymbol{j}}$ for all $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}^*$

$\boldsymbol{V}^* := \boldsymbol{V}$;
do
  $\boldsymbol{N} := \{0\}$;
  Compute $w_{\boldsymbol{\ell},\boldsymbol{j}}$ for all $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}^*$ via w or dynamical system $\partial_t f = \mathcal{R}_{\boldsymbol{V}^*} f$;
  for $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}^*$ **do**
    if $\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_\infty \geq \tau \max_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_\infty$ **then** `/* Check if element needs refining */`
      Compute all children $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$ of $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ using Definition 4;
      for *children* $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$ *of* $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$ **do**
        if $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'} \not\subset \boldsymbol{V}^*$ *and* $\boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'} \not\subset \boldsymbol{N}$ **then**
          $\boldsymbol{N} := \boldsymbol{N} \oplus \boldsymbol{W}_{\boldsymbol{\ell}',\boldsymbol{j}'}$; `/* Add element */`
          $w^{\mathrm{n}}_{\boldsymbol{\ell}',\boldsymbol{j}'} = 0$; `/* Zero out new element at` t^{n} `*/`
  $\boldsymbol{V}^* := \boldsymbol{V}^* \oplus \boldsymbol{N}$;
while $\boldsymbol{N} \neq \{0\}$; `/* Repeat until no children are added. */`

Algorithm 2: Adaptive coarsening using ℓ^∞-norm

Input : Adaptive sparse-grid $\boldsymbol{V}$, threshold $\tau > 0$, coarsening factor $0 < \mu < 1$
Input : Coefficients $w_{\boldsymbol{\ell},\boldsymbol{j}}$ for all $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}$
Output: Adaptive sparse-grid $\boldsymbol{V}^*$

$\boldsymbol{V}^* := \{0\}$;
for $\boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}} \subset \boldsymbol{V}$ **do**
  if $\|w_{\boldsymbol{\ell},\boldsymbol{j}}\|_\infty > \mu\tau \max_{(\boldsymbol{\ell}',\boldsymbol{j}')} \|w_{\boldsymbol{\ell}',\boldsymbol{j}'}\|_\infty$ **then** `/* Check if element needs to be removed */`
    $\boldsymbol{V}^* := \boldsymbol{V}^* \oplus \boldsymbol{W}_{\boldsymbol{\ell},\boldsymbol{j}}$;



(a) Phase space plot of $\langle f \rangle_{v_y,v_z}$ in (x, v_x) where f is the adaptive sparse-grid distribution.

(b) Plot showing which hierarchical elements are active in Figure 4.4.1a. The points represent the (x, v_x) coordinates of the barycenter of the support of each active wavelet.

Figure 4.4.1: Riemann problem – Section 5.3 – $\nu = 1$: Adaptive Sparse-grid Method at $t = 0.04918$. The threshold is $\tau = 10^{-4}$ and the adaptive sparse-grid cannot refine past $\boldsymbol{\ell} = (7, 6, 6, 6)$.

5. Numerical Experiments

In this section, we present results from various test problems relevant to plasma physics. Our goals are to demonstrate the performance of the adaptive sparse-grid and mixed-grid DG methods with IMEX time stepping implemented in ASGarD, and investigate the computational benefit of the adaptive sparse-grid and mixed-grid methods over the full-grid methods (see Section 5.1 for definitions). In increasing degree of complexity, we consider: (i) relaxation to a Maxwellian velocity distribution (Section 5.2); (ii) a Riemann problem for two different values of the collision frequency ν (Section 5.3); (iii) and the collisional Landau damping problem (Section 5.4), also for two different values of the collision frequency. All the results presented in this section were obtained with quadratic polynomials, i.e., $k = 2$. This choice of k is natural considering that the velocity moments with respect to 1, $\mathbf{v}$, and $|\mathbf{v}|^2$ compose the important fluid variables.

5.1. Choice of Grids

In the simulations presented below we choose to compare results obtained with three types of grids: full-grid, mixed-grid, and adaptive sparse-grid. We provide the specifics of each grid in this section.

Our first choice is the standard full-grid $\boldsymbol{V}_{\boldsymbol{\ell}}$, where $\boldsymbol{\ell} = (\ell_x, \ell_v, \ell_v, \ell_v)$. We use the Chu reduction method of Section 2.2 in Sections 5.3 and 5.4 with $\boldsymbol{\ell} = (\ell_x, \ell_v)$ to build reference solutions and numerical approximations with the full-grid. This is because the full-grid space is too large in comparison to the other two grids and can easily fill the memory of a single-node machine. When using the Chu reduction, the discretization is performed using a local Legendre polynomial basis instead of the multiwavelets. When determining the degrees of freedom or number of active elements for a full-grid run, we will always assume that the underlying run is 4D, even if the Chu reduction method is used.

We have found that standard 4D sparse-grids such as $\hat{\boldsymbol{V}}_{\boldsymbol{\ell}}$ are unstable for the VPLB model in (2.15). This is due to both the lack of resolution in x and the lack of regularity of the distribution function in physical space. Specifically, the temperature θ_f becomes negative which causes the solution to blow up. As we expect savings to come from the smoothness in velocity space, induced by the LB collision operator, we propose a mixed-grid approach for our second choice. The mixed-grid of level $\boldsymbol{\ell} = (\ell_x, \ell_v, \ell_v, \ell_v)$ is defined by

$$\tilde{\boldsymbol{V}}_{\boldsymbol{\ell}} = \bigoplus_{\substack{\boldsymbol{\ell}' : \ell'_1 \le \ell_x, \\ |(\ell'_2, \ell'_3, \ell'_4)|_1 \le \ell_v}} \boldsymbol{W}_{\boldsymbol{\ell}'} = V_{\ell_x} \otimes \hat{\boldsymbol{V}}_{(\ell_v, \ell_v, \ell_v)}. \tag{5.1}$$

The mixed-grid space is a tensor product of a full-grid in physical space and a sparse-grid in velocity space. This can be viewed as a sparse-grid in velocity space attached to each degree of freedom in x, and thus provides computational savings relative to the full-grid (without the Chu reduction method). The dimension of $\tilde{\boldsymbol{V}}_{\boldsymbol{\ell}}$ is $\mathcal{O}((k+1)^4 2^{\ell_x+\ell_v}\ell_v^2)$. We find this space is sufficient to maintain stability of the DG method for the problems considered here.

Additionally, since 1 and $v_y^2+v_z^2$ are admissible DG functions that live on level $(0,0)$ in (v_y, v_z) when $k \geq 2$, a 4D full-grid of level $(\ell_x, \ell_v, 0, 0)$ is sufficient to recover g_1 and g_2 in a 2D full-grid of level (ℓ_x, ℓ_v). Since a full-grid of level $(\ell_x, \ell_v, 0, 0)$ is a subgrid of a mixed-grid with level $(\ell_x, \ell_v, \ell_v, \ell_v)$, the reduced moments g_1 and g_2 created by the mixed-grid solution will be similar to the full-grid. However, for $k \leq 3$, the function $v_y^4 + v_z^4$ is not a DG function, and its projection onto the DG space will excite finer level coefficients that are better captured by the full-grid than by the mixed-grid for a certain level. We therefore evolve g_3 in the Chu reduction method in order to better understand differences in accuracy between the mixed-grid and full-grid methods.

Our last grid is the adaptive sparse-grid, $\boldsymbol{V}$, that is coarsened and refined as detailed in Algorithms 1 and 2, using ℓ^∞-thresholding. The refinement threshold τ will be problem dependent, but we use the coarsening factor $\mu = 0.1$, motivated from [40, 48], for all of our examples. Instead of a max level $N_{\max}$ used in Section 4, we will not allow the adaptive sparse-grid to refine above a full-grid of specified level $\boldsymbol{\ell} = (\ell_x, \ell_v, \ell_v, \ell_v)$. The number of degrees of freedom, or active elements, presented in the results below will be of the adaptive sparse-grid solution after the refinement step but before coarsening.

It is useful to view each of these grids as a velocity grid attached to each spatial degree of freedom. The full-grid attaches a three-dimensional full-velocity grid to every spatial degree of freedom while the mixed-grid attaches a sparse-velocity grid. The adaptive sparse-grid attaches a variable velocity grid, with possibly zero elements, to each spatial degree of freedom.

Finally, we will track the number of active elements, see Definition 3, as opposed to degrees of freedom in order to more clearly present the advantages of the mixed-grid and adaptive sparse-grid methods.

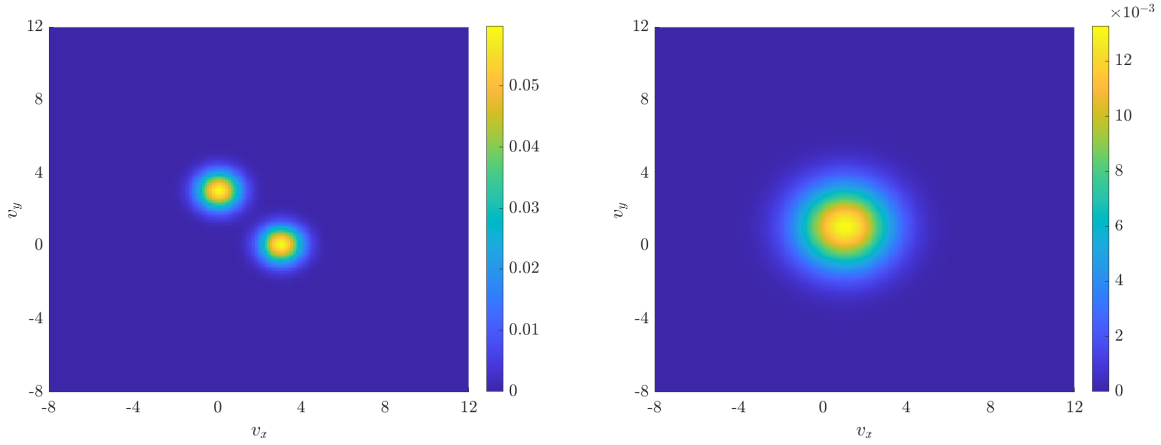
5.2. Relaxation Problem

We first consider the $0x3v$ problem in (2.14) in order to test the relaxation to equilibrium induced by the LB collision operator. In this case $f = f(\mathbf{v}, t)$ and the computational domain is truncated so that $\mathbf{v} \in (-8, 12)^3$. The initial condition is given by the sum of three Maxwellians, each sharing $n_f = 1/3$, $\theta_f = 1/2$, but differing in the bulk velocities, which are given by $[3, 0, 0]$, $[0, 3, 0]$, and $[0, 0, 3]$, respectively. This initial condition induces the following velocity moments: $n_f = 1$, $\mathbf{u}_f = [1, 1, 1]^\intercal$, and $\theta_f = 2.5$. By the properties of the LB collision operator Proposition 1, these moments are expected to remain constant in time and the velocity distribution to relax to the Maxwellian defined by the initial moments.

For this test, we will use a 3D sparse-grid of level (ℓ_v, ℓ_v, ℓ_v) as a substitute for the mixed-grid. The 4D $(1x3v)$ definitions of the full-grid and adaptive sparse-grid naturally carry to the 3D $(0x3v)$ case. We set $\nu = 10^3$, $\Delta t = 5 \times 10^{-4}$, and use backward Euler time stepping for this problem, with a tolerance of 10^{-8} for the GMRES implicit solve. Figure 5.2.1 illustrates the initial and final (equilibrium) distributions in the (v_x, v_y)-plane for a full-grid model.

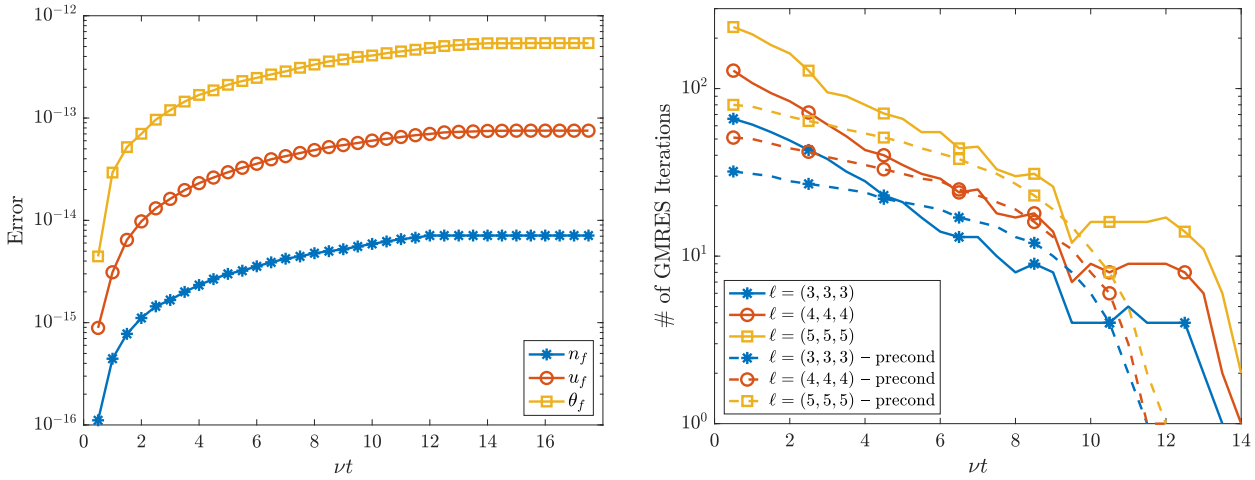
Figure 5.2.2a plots the change in the fluid variables n_f, u_f, and θ_f from their initial values as a function of νt, when using the full grid; the figure clearly shows that the loss in conservation of the moments is well below the GMRES tolerance. The error profiles for the mixed-grid and adaptive sparse-grid runs are similar, but not shown.

Figure 5.2.2b shows the number of GMRES iterations for each timestep for varying full-grid levels. The block-Jacobi preconditioner reduces the number of GMRES iterations for each simulation (dashed lines) and overall smoothly decays the iteration count as a function of timestep. However, the constant jump of the iteration count, in logarithmic scale, between velocity levels in both the standard GMRES and precondioned version shows that the preconditioner does not asymptotically lower the $\mathcal{O}(4^{\ell_v})$ conditioning of the diffusion term in the LB operator. We found that the sparse-grid's iteration count was roughly two-thirds of the full-grid for the same level. Additionally, we found that the adaptive sparse-grid method often included elements from level 9 grids which caused a significant increase in the number of GMRES iterations in the



(a) $\nu t = 0$ (b) $\nu t = 20$

Figure 5.2.1: Relaxation Problem – Section 5.2: 2D plot of the velocity distribution $f_h(v_x, v_y, v_z = 0.019)$ at the start (left) and end (right) of a relaxation simulation. These results were obtained with a full-grid run with $\boldsymbol{\ell} = (5, 5, 5)$.



(a) Plots of $|n_f(t) - n_f(0)|$, $|u_f(t) - u_f(0)|$, and $|\theta_f(t) - \theta_f(0)|$ versus time where the fluid variables are approximations with $\boldsymbol{\ell} = (4, 4, 4)$.

(b) Number of GMRES iterations per timestep with (dashed) and without (solid) a block-Jacobi preconditioner. GMRES was restarted every 100 iterations and exited when the residual norm was less than 1e-8. Instances when GMRES exited in zero iterations are not plotted.

Figure 5.2.2: Relaxation Problem – Section 5.2: Plots of interest for full-grid runs with varying levels.

adaptive sparse-grid over full-grid runs with a similar number of active elements. The results of this test suggest that better preconditioners, e.g. multigrid type, are needed in order to effectively condition the problem. This is especially crucial for the adaptive sparse-grid method as finer level basis functions can be active for mild choices of the refinement threshold τ.

Figure 5.2.3 illustrates the advantages of adaptive sparse-grids over the full- and mixed-grid methods for the relaxation problem. The L^2 error of the relaxed distribution, relative to the analytic Maxwellian, is plotted versus the number of active elements. When plotted against the number of active elements, adaptive sparse grids are more accurate and asymptotically superior when compared against the other formulations. Additionally, the mixed-grid is comparable to the full-grid with the mixed-grid only gaining an advantage when a large number of active elements is used. This is not surprising as the Maxwellian, being radially symmetric, has large mixed derivatives and the coefficients to capture mixed derivative information are

thrown away in the standard sparse-grid construction. However, adaptive sparse-grids are able to capture these mixed-derivative coefficients.

We lastly explore how the adaptive sparse-grid refinement threshold τ correlates with the relative error of the approximation. In this problem the analytic equilibrium has a L^2 norm of approximately 7.539×10^{-2}. Using the L^2-errors provided from Figure 5.2.3, the relative L^2-error for $\tau = 10^{-2}$ is approximately 0.6τ and creeps to 2.03τ when $\tau = 10^{-5}$. Therefore τ provides a good estimate for the relative error of the problem. This is in part because the equilibrium is smooth so that the chosen maximum level of 9 is sufficiently large to capture the needed coefficient data and achieve the expected errors.

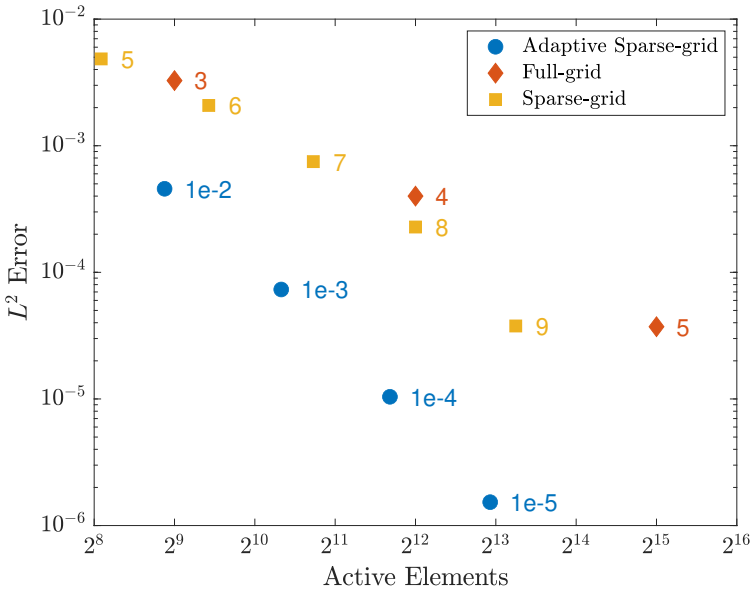


Figure 5.2.3: Relaxation Problem – Section 5.2: The L^2 error of solution versus the number of active elements used for the full-grid and adaptive sparse-grid runs. The error is calculated against the analytic equilibrium in (2.10). The full- and mixed-grid runs were set at $\boldsymbol{\ell} = (\ell_v, \ell_v, \ell_v)$ where ℓ_v is the number by the marker. The marker next to the adaptive sparse-grid runs is the tolerance τ at which the run was set, and the adaptive run was not allowed to exceed a level of $\boldsymbol{\ell} = (9, 9, 9)$.

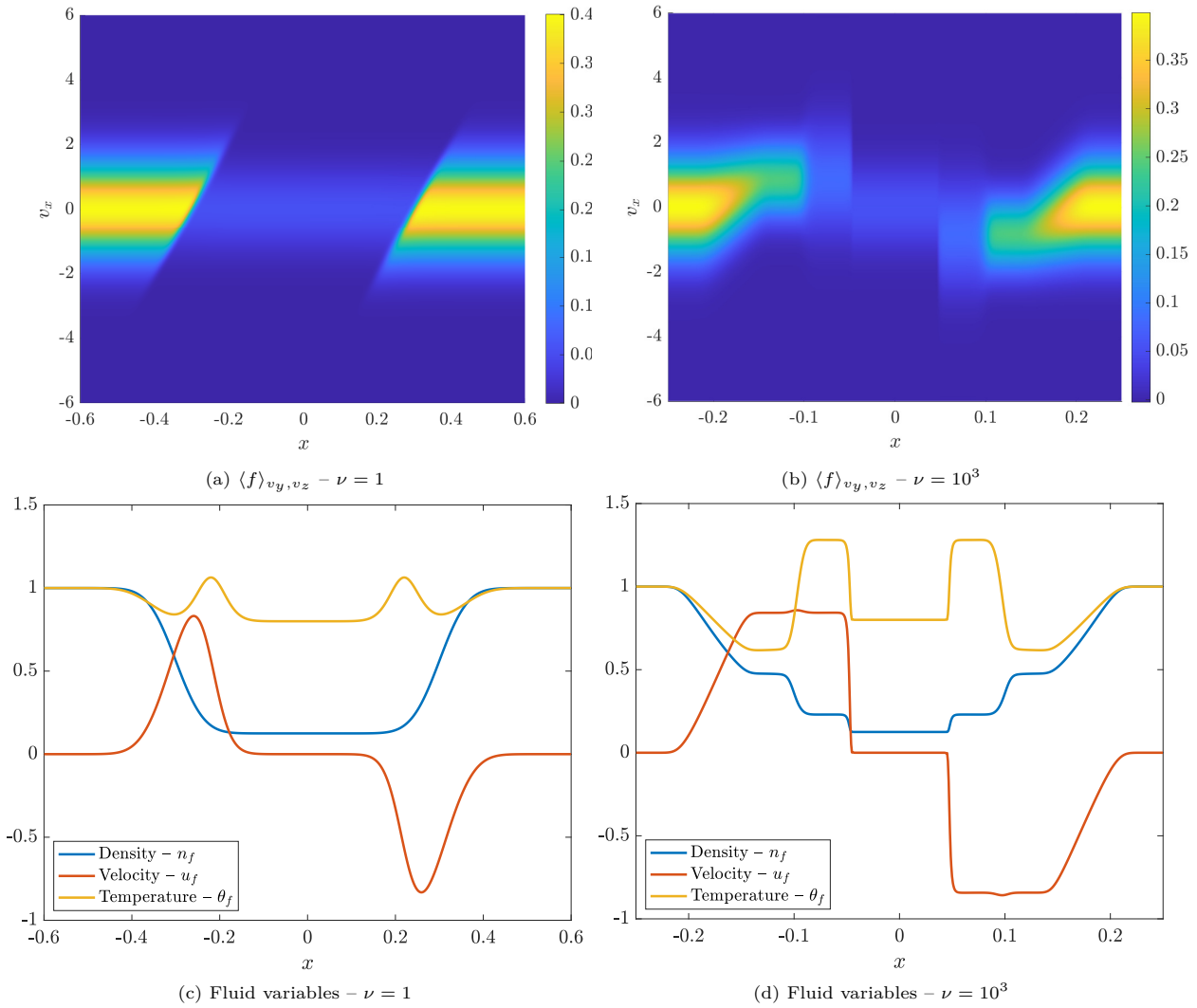
5.3. Riemann Problem

Next, we consider a problem that includes both phase-space advection and collisions. The Sod shock tube problem [78] is a standard test for numerical simulations of kinetic models with collisions (e.g., [10, 31]). For this test, the PDE is given by (2.15) We consider two regimes of collisionality: The first is an intermediate regime with $\nu = 1$, and the second is a collisional regime with $\nu = 10^3$. For both problems we fix $\mathbf{v} \in (-6, 6)^3$ and set the initial condition to a Maxwellian with moments given by:

$$\begin{bmatrix} n_f \\ \mathbf{u}_f \\ \theta_f \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \\ 1 \end{bmatrix} \quad \text{if } |x| \geq s_{\text{initial}}; \qquad \begin{bmatrix} n_f \\ \mathbf{u}_f \\ \theta_f \end{bmatrix} = \begin{bmatrix} 0.125 \\ \mathbf{0} \\ 0.8 \end{bmatrix} \quad \text{if } |x| < s_{\text{initial}} \tag{5.2}$$

where s_{initial} is the location of the initial discontinuity. We set the GMRES tolerance to 10^{-8}.

Figure 5.3.1 shows plots of the distribution in the (x, v_x)-plane and plots of the velocity moments versus position, as obtained with the full-grid using the Chu reduction technique. We will use these as reference solutions when evaluating the performance of the adaptive sparse grid method. For moderate collisionality, i.e. $\nu = 1$, the distribution, as shown in Figure 5.3.1a, deviates from the Maxwellian due to the streaming and features a discontinuity in the (x, v_x) space. Additionally, as seen in Figure 5.3.1c, the streaming effect smooths out features of the fluid variables. In the collision dominated regime ($\nu = 10^3$), the distribution,



(a) $\langle f \rangle_{v_y,v_z}$ – $\nu = 1$

(b) $\langle f \rangle_{v_y,v_z}$ – $\nu = 10^3$

(c) Fluid variables – $\nu = 1$

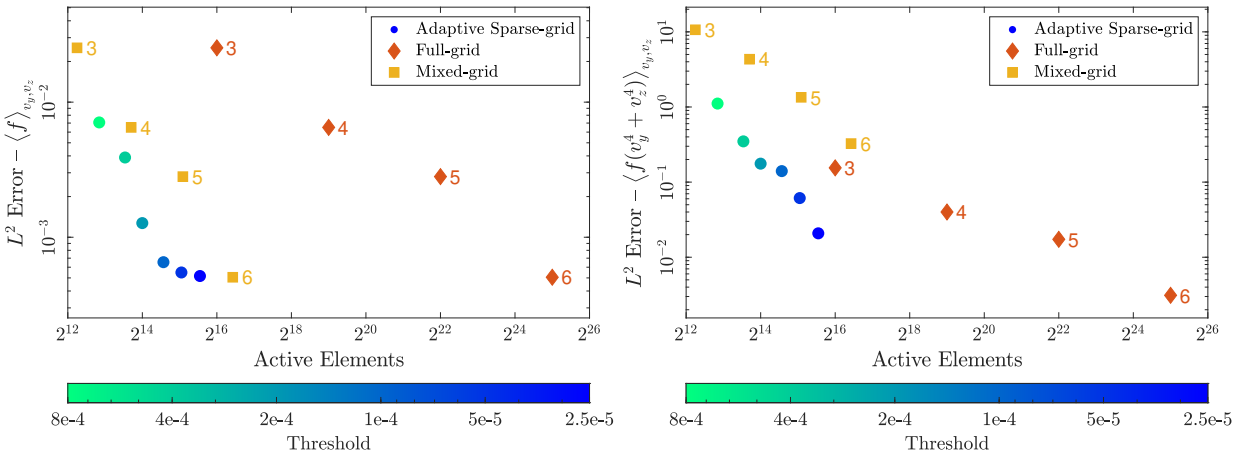
(d) Fluid variables – $\nu = 10^3$

Figure 5.3.1: Riemann Problem – Section 5.3: Plots of the distribution and fluid variables for the which is computed using the Chu reduction model in Section 2.2 with $(\ell_x, \ell_v) = (9, 8)$ for $\nu = 1$ and $(\ell_x, \ell_v) = (8, 8)$ for $\nu = 10^3$. The $\nu = 1$ and $\nu = 10^3$ plots are taken at time $t = 0.04918$ and $t = 0.05$ and with $s_{\text{initial}} = 0.3$ and $s_{\text{initial}} = 9/64$ respectively.

as seen in Figure 5.3.1b, remains close to a local Maxwellian parameterized by the local fluid variables in Figure 5.3.1d.

When comparing results obtained with different grids, we first consider the case of $\nu = 1$, and we set $x \in (-0.6, 0.6)$, $s_{\text{initial}} = 0.3$, final time $T = 0.04918$, and time step $\Delta t = 2.3419 \times 10^{-4}$. Our reference solution is the full-grid solution of level $\boldsymbol{\ell} = (9, 8, 8, 8)$, displayed in the left panels in Figure 5.3.1. Figure 5.3.2 shows the error versus the number of active elements for g_1 and g_3 (defined in Section 2.2). It is shown in Figure 5.3.2a that the mixed-grid yields the same error as the full-grid – for the same velocity space resolution level ℓ_v. This is because g_1 is embedded in the mixed-grid as mentioned in Section 5.1. The adaptive sparse-grid error saturates at the level of the mixed-grid error when $\ell_v = 6$, but with about 50% fewer active elements. The saturation is because the adaptive grid is not allowed to refine past level $\boldsymbol{\ell} = (7, 6, 6, 6)$ in the hierarchy (see Section 5.1) and therefore the associated error will not be significantly lower than the full-grid of level $\boldsymbol{\ell} = (7, 6, 6, 6)$. When viewing the same plot for the higher-order moment g_3 in Figure 5.3.2b, we see the degradation in the mixed-grid method when compared to the full-grid and

adaptive sparse-grid methods. While the slope in the error from the mixed-grid method is steeper than the full-grid method, its error constant is significantly larger. Additionally, the adaptive sparse-grid method is significantly better than both the mixed-grid and full-grid methods with respect to both the slope and error constant.



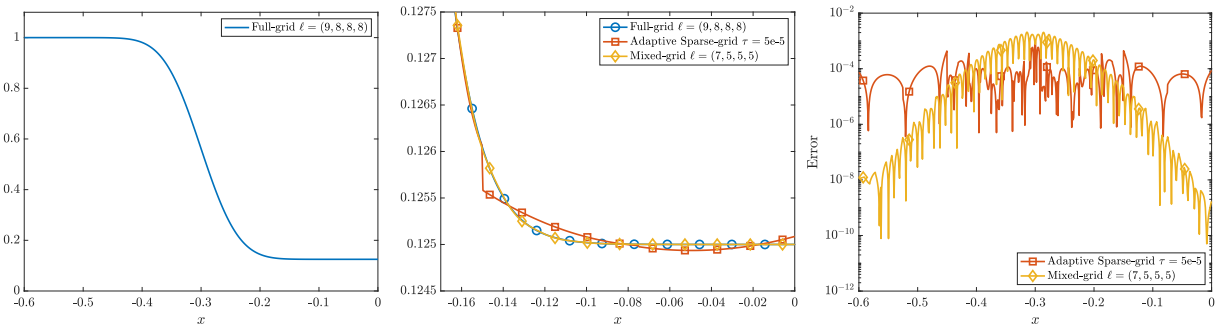
(a) $\left\| \langle f \rangle_{v_y,v_z} - g_1 \right\|_{L^2}$ (b) $\left\| \langle f(v_y^4+v_z^4) \rangle_{v_y,v_z} - g_3 \right\|_{L^2}$

Figure 5.3.2: Riemann Problem – Section 5.3 – $\nu = 1$: Errors of the distribution and fluid variables at $t = 0.01$ for the $1x3v$ Riemann problem in Section 5.3 with $\nu = 1$. All errors are measured against the full-grid solution at level $\boldsymbol{\ell} = (9, 8, 8, 8)$ (see Figure 5.3.1a). All adaptive sparse-grid runs are capped at $\boldsymbol{\ell} = (7, 6, 6, 6)$. The full- and mixed-grid runs use $\boldsymbol{\ell} = (7, \ell_v, \ell_v, \ell_v)$ where ℓ_v is the symbol by each marker. The adaptive sparse-grid method performs well in both cases while the mixed-grid method is accurate only in the low-order moment.

Figure 5.3.3 shows the particle density n_f (left and middle panels) and the pointwise error of the particle density (right panel) for a mixed-grid and an adaptive sparse-grid model with a similar number of active elements. Figure 5.3.3a shows that the density appears to be relatively constant in x toward the edges of the plot. When zooming in on a smaller x-range near the right edge, see Figure 5.3.3b, it becomes clear that the density obtained with the adaptive sparse-grid features a discontinuity (around $x = -0.15$) and exhibits more spatial variation when compared to the full-grid and mixed-grid solutions. This is primarily caused by the adaptive method uniformly distributing the error across the spatial domain, and this is further evidenced in the error plot (see Figure 5.3.3c), where the error in x is much more uniform across the spatial domain for the adaptive sparse-grid than it is with the mixed-grid method. In the mixed-grid method, where each DOF in x is attached with the same sparse-grid in v, the moment errors are much smaller away from the wave regions, i.e., the regions where the moments are constant.

Next, we consider the case with $\nu = 10^3$. Here we set $x \in (-0.25, 0.25)$, $s_{\text{initial}} = 9/64$, $T = 0.05$, and $\Delta t = 2 \times 10^{-4}$. Figure 5.3.4 shows the error of g_1 and g_3 against the number of active elements. In this higher-collisional regime, the distribution is much smoother in velocity, and the L^2 error saturates sooner than when $\nu = 1$. This saturation is due to the dominant error that appears near the discontinuities in the x-domain (see Figures 5.3.1b and 5.3.1d). In Figure 5.3.4a, the mixed-grid and adaptive sparse-grid methods are very similar. At saturation, the number of active elements for the mixed-grid and adaptive sparse grid, around 2^{14} are approximately 128 times fewer than the number of active elements in the full-grid, which is 2^{21}.

When looking at the error in the higher-order moments, Figure 5.3.4b, we observe a separation in the performance of the mixed-grid and adaptive sparse-grid that is similar to the $\nu = 1$ case. However, in this case, the adaptive sparse-grid method has nearly hit saturation while the mixed-grid with $\ell_v = 6$ is still not at saturation. In particular, the grouping of the errors for mixed-grid $\ell_v = 6$ and the full-grid $\ell_v = 3$ is similar to the grouping in the relaxation case (see Figure 5.2.3). This shows that the dominant error in the mixed-grid method is the lack of velocity resolution sufficient to capture the local Maxwellian behavior of

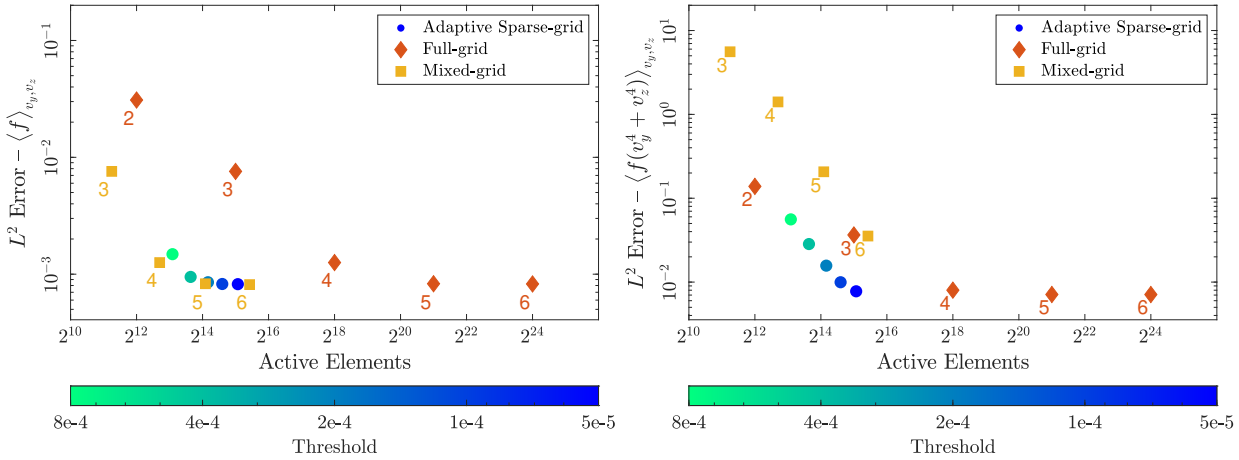


(a) n_f (b) n_f – slice for $x \in (-0.17, 0)$ (c) Error – n_f

Figure 5.3.3: Riemann Problem – Section 5.3 – $\nu = 1$: Plots of the density and error to the reference density for the $1x3v$ Riemann problem with $\nu = 1$ and $t = 0.04918$. The reference density is calculated with the full-grid method at level $\boldsymbol{\ell} = (9, 8, 8, 8)$ (see Figure 5.3.1c). The adaptive sparse-grid solution is not allowed to be refined beyond level $\boldsymbol{\ell} = (7, 6, 6, 6)$. The adaptive sparse-grid method equally spaces out the error in physical space while the mixed-grid is only accurate in the constant regions of the density.

the distribution.

We include a plot of the fourth-order moment g_3 in the (x, v_x)-plane for each grid type, each having a similar number of degrees of freedom, in Figure 5.3.5. The full-grid solution, Figure 5.3.5a, exhibits discontinuities on element interfaces in the velocity dimension (due the discontinuous basis) while the mixed-grid moment, Figure 5.3.5b, is oscillatory in the region immediately left of the contact line, i.e. $x \in (-0.15, -0.1)$. The solution obtained with the adaptive sparse-grid, Figure 5.3.5c, is the most accurate of the three and does not suffer from either of the previously mentioned artifacts.

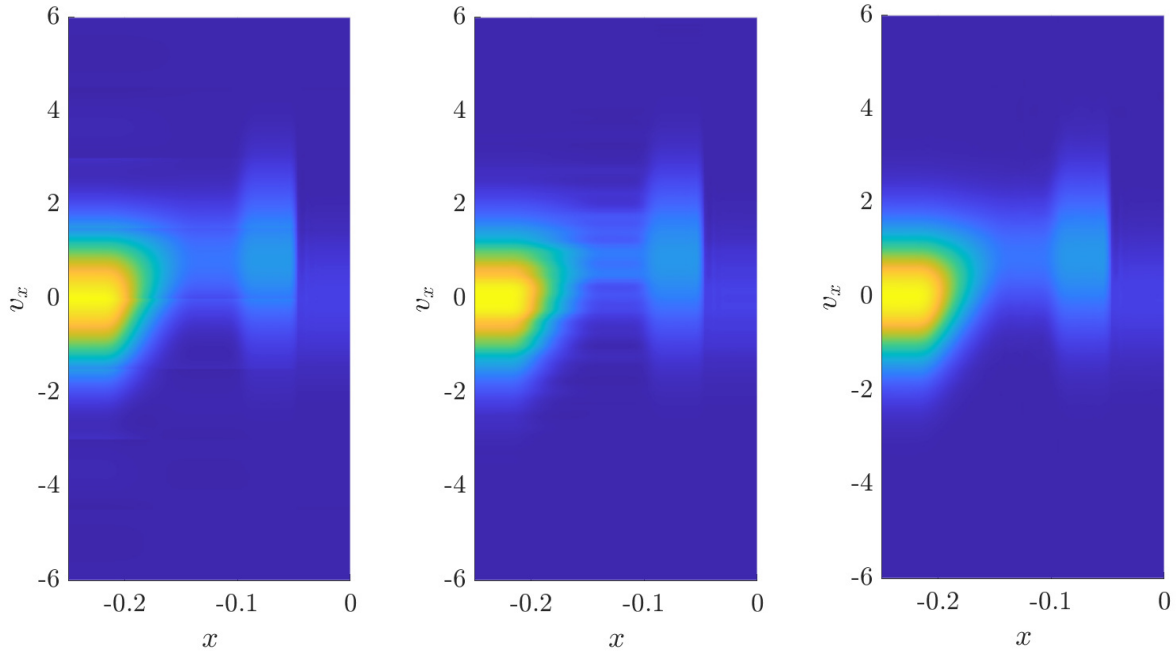


(a) $\left\| \langle f \rangle_{v_y,v_z} - g_1 \right\|_{L^2}$ (b) $\left\| \langle f(v_y^4 + v_z^4) \rangle_{v_y,v_z} - g_3 \right\|_{L^2}$

Figure 5.3.4: Riemann Problem – Section 5.3 – $\nu = 10^3$: Errors of the distribution at $t = 0.05$ for the $1x3v$ Riemann problem in Section 5.3 with $\nu = 10^3$. All errors are measured against the full-grid solution at level $\boldsymbol{\ell} = (8, 8, 8, 8)$ (see Figure 5.3.1b). All adaptive sparse-grids are capped at level $\boldsymbol{\ell} = (6, 6, 6, 6)$. The full- and mixed-grid levels are given by $\boldsymbol{\ell} = (6, \ell_v, \ell_v, \ell_v)$ where ℓ_v is the symbol to the lower left of the marker. The quick saturation of the error is due to smoothness in velocity and the discontinuities in the fluid variables (see Figure 5.3.1d). The adaptive sparse-grid method performs well in both cases while the mixed-grid method is accurate only in the low-order moment.

5.4. Collisional Landau Damping

Finally, we consider a version of the collisional Landau damping test (e.g., [25, 43, 32]), which involves phase-space advection of charged particles, influenced by a self-consistent electric field and particle collisions.

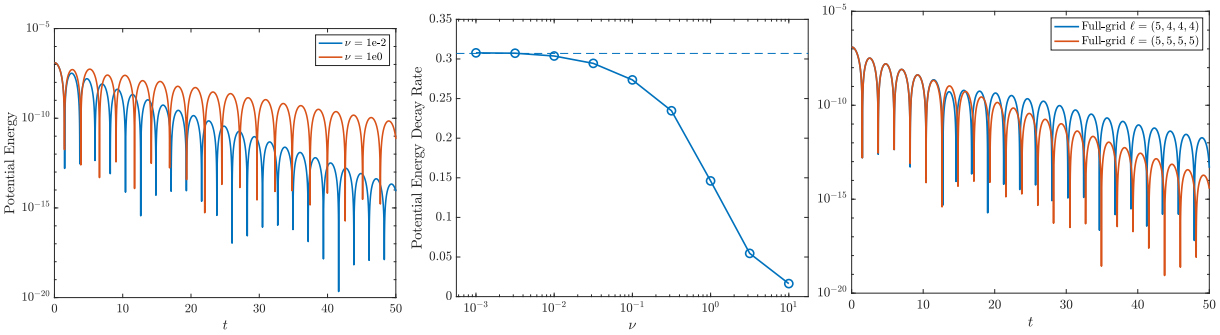


(a) Full-grid $\boldsymbol{\ell} = (6, 3, 3, 3)$ (b) Mixed-grid $\boldsymbol{\ell} = (6, 6, 6, 6)$ (c) Adaptive Sparse-Grid $\tau = 5 \times 10^{-5}$

Figure 5.3.5: Riemann Problem – Section 5.3 – $\nu = 10^3$: Plots of $\langle f(v_y^4 + v_z^4)\rangle_{v_y,v_z}$ for the $1x3v$ Riemann Problem with $\nu = 10^3$ and $t = 0.05$ in the (x, v_x) plane for $x \in (-0.25, 0)$ and $v_x \in (-6, 6)$. The adaptive sparse-grid was not allowed to refine past level $\boldsymbol{\ell} = (6, 6, 6, 6)$. The artifacts seen in the full-grid and mixed-grid solutions are not found in the adaptive- sparse grid solution.

The PDEs solved in this test are given by the VPLB system in (2.15) and (2.16).

The $1x3v$ phase-space domain is given by $x \in (-2\pi, 2\pi)$ and $\mathbf{v} \in (-6, 6)^3$, and the model is evolved to the final time $T = 50$. The initial condition is set as Maxwellian with a small spatial perturbation so that the velocity moments are $n_f = 1 + 10^{-4}\cos(\frac{x}{2})$, $u_f = 0$, $\theta_f = 1$. The timestep taken depends on the spatial resolution and $\max|v_x|$, and is taken as $\Delta t = \frac{0.75}{30}\Delta x$, where $\Delta x = \frac{4\pi}{2^{\ell_x}}$.



(a) The electric potential energy versus time for two collision frequencies ν.

(b) Exponential decay rate γ for the potential energy as a function of ν. The dashed line corresponds to $\gamma = 0.307$.

(c) Plot of the electric potential for $\nu = 10^{-2}$ and velocity resolutions $\ell_v = 4$ and $\ell_v = 5$.

Figure 5.4.1: Collisional Landau Problem – Section 5.4: Plots demonstrating collisional Landau damping. All runs use the Chu reduction method of Section 2.2. The levels set are $\ell_x = 5$ and $\ell_v = 6$ except in Figure 5.4.1c.

In the collisionless case, the Landau damping problem is characterized by exponential decay of the

potential energy with time, $E_{\text{Pot}}(t) = \frac{1}{2}\int_{\Omega_x} E^2\,\mathrm{d}x \propto \exp(-\gamma t)$, where the damping rate is $\gamma \approx 0.307$ [5]. Moreover, with evolving time, the solution will exhibit increasingly smaller-scale structures about the Maxwellian that eventually become unresolved with fixed or finite resolution (see [29]). With collisions, the damping rate decreases with increasing collision frequency (e.g., [25, 43]), tending to zero in the Euler–Poisson limit ($\nu \to \infty$).

Figure 5.4.1a shows the potential energy versus time, as obtained with the full-grid method, for $\nu = 10^{-2}$ (blue) and $\nu = 1$ (red). Figure 5.4.1b shows numerically determined damping rates as a function of collision frequency. These results were obtained with the full-grid method using the Chu reduction technique. The damping rate is determined by a least squares fit using the local maxima of the potential energy. For small collision frequencies, the damping rate tends to the expected result in the collisionless limit indicated by the horizontal dashed line. The damping rate drops rapidly for $\nu \gtrsim 0.3$, and has dropped to about 0.01 for $\nu = 10$. Figure 5.4.1c compares the evolution of the potential energy versus time for the $\nu = 10^{-2}$ case with two different velocity resolutions; $\ell_v = 4$ (blue) and $\ell_v = 5$ (red). For the simulation with the coarser velocity resolution, the damping rate is consistent with the analytic prediction until $t \approx 10$. For $t \gtrsim 10$, the potential energy increases briefly with time before decreasing again with a modified damping rate. For the finer velocity resolution, the damping rate stays constant at the correct value for all times. Based on this observation, we consider $\ell_v = 5$ the minimum resolution needed to perform satisfactory on this test when $\nu = 10^{-2}$. We performed a similar comparison with $\nu = 1$, which revealed that $\ell_v = 4$ is sufficient for this case. In the following, we consider the two cases: $\nu = 10^{-2}$ (low collisionality) and $\nu = 1$ (moderate collisionality), in more detail to compare the adaptive sparse-grid method against the full-grid method. Due to the embedding of the $1x1v$ full-grid into the $1x3v$ mixed-grid as discussed in Section 5.1, the electric field E is similar for the full- and mixed-grids of the same level. For this reason, the mixed-grid results are omitted.

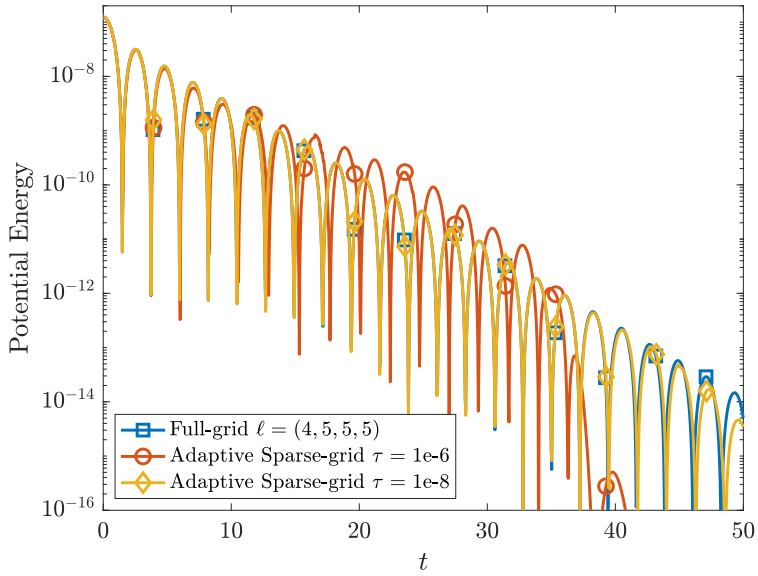


Figure 5.4.2: Collisional Landau Problem – Section 5.4: Plot of the potential energy with $\nu = 10^{-2}$. The adaptive sparse grid is not allowed to refine past level $\boldsymbol{\ell} = (4,5,5,5)$, and the GMRES tolerance is set to 10^{-14}. A tolerance of $\tau = 10^{-6}$ is not sufficient to capture the proper decay. The tolerance of $\tau = 10^{-8}$ agrees quite well with the full-grid solution except for a slight deviation at longer times.

Figure 5.4.2 compares adaptive sparse-grid against full-grid for the low collisionality case by plotting the potential energy versus time. The full-grid run with $\boldsymbol{\ell} = (4,5,5,5)$, used as reference in Figure 5.4.2, is in close agreement with the full-grid run with $\boldsymbol{\ell} = (5,5,5,5)$ plotted in the right panel of Figure 5.4.1c. When the tolerance for refinement is $\tau = 10^{-6}$, the adaptive results agree with the full-grid results up to about $t = 10$. For later times, the resolution allowed by the threshold is not sufficient to capture the correct

damping of the potential energy. Past $t = 35$, the solution coarsens to only global elements in x, i.e. $\ell_x = 0$, which forces the electric field to zero before refinement, and causes unreliable behavior in the potential energy. When the tolerance is reduced to $\tau = 10^{-8}$, the adaptive sparse-grid is in better agreement with the full-grid throughout the simulation, although some deviations near the end are observed. For the $\tau = 10^{-6}$ case, the number of active elements stays around 1.1×10^4 throughout, while for the $\tau = 10^{-8}$ case the number of active elements starts out around 3×10^4, which drops steadily to about 2.5×10^4 at the end of the simulation. In comparison, the full-grid with $\boldsymbol{\ell} = (4, 5, 5, 5)$, the maximum allowed for the adaptive spares-grid, consists of about 5.2×10^5 elements. Thus, the adaptive grid provides significant savings in terms of the number of degrees of freedom.

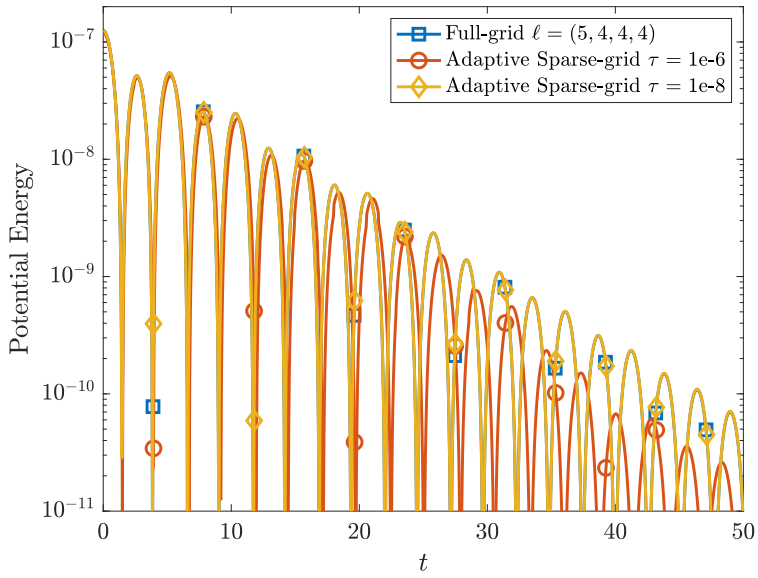
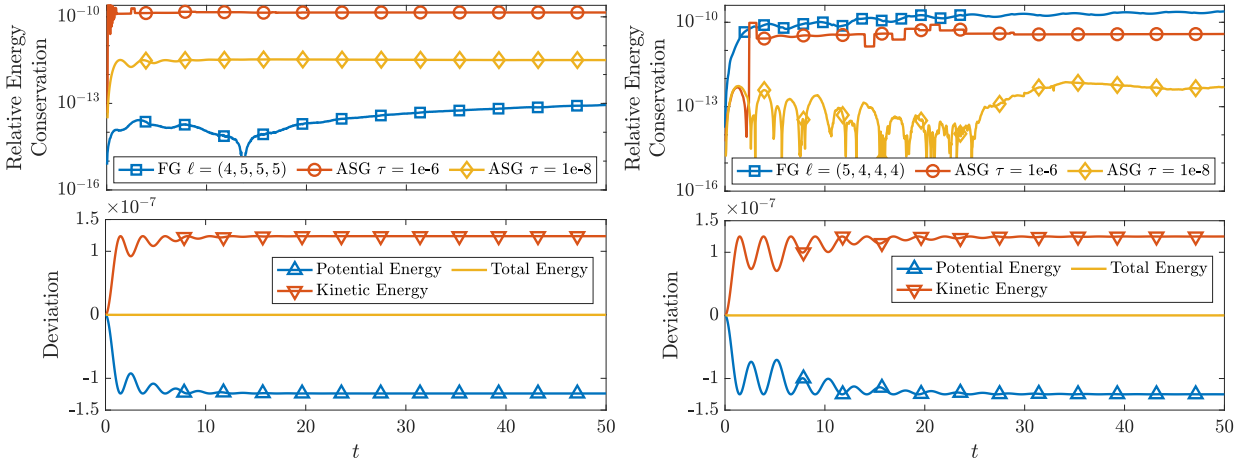


Figure 5.4.3: Collisional Landau Problem – Section 5.4: Plot of the potential energy with $\nu = 1$. The adaptive sparse grid is not allowed to refine past level $\boldsymbol{\ell} = (5, 4, 4, 4)$, and the GMRES tolerance is set to 10^{-11}. A tolerance of $\tau = 10^{-6}$ is not sufficient to capture the proper decay. The tolerance of $\tau = 10^{-8}$ agrees quite well with the full-grid solution at all times plotted.



(a) $\nu = 10^{-2}$. The GMRES tolerance is set to 10^{-14}

(b) $\nu = 1$. The GMRES tolerance is set to 10^{-11}

Figure 5.4.4: Collisional Landau Problem – Section 5.4: Top: Relative change in total energy versus time. Bottom: Deviation of the potential, kinetic, and total energy from the initial condition for the adaptive sparse-grid method with $\tau = 10^{-8}$.

Figure 5.4.3 compares adaptive sparse-grid against full-grid for the moderate collisionality case, where we plot the same quantities as in Figure 5.4.2. For this collisionality, we have determined that a full-grid resolution of $\boldsymbol{\ell} = (5, 4, 4, 4)$ is sufficient to accurately capture the evolution of the potential energy. Similar to the low collisionality case, the potential energy evolution obtained with adaptivity threshold $\tau = 10^{-6}$ is not in satisfactory agreement with the full-grid and analytic results. However, we find that the adaptive spare-grid and full-grid results are indistinguishable when the adaptivity threshold is reduced to $\tau = 10^{-8}$. For $\tau = 10^{-6}$, the number of active elements stays roughly constant at about 4.8×10^3, while for the case with $\tau = 10^{-8}$, the number of active elements starts out around 1.1×10^4, and decreases to about 7×10^3 at the end of the simulation. For comparison, the full-grid with $\boldsymbol{\ell} = (5, 4, 4, 4)$ consists of about 1.3×10^5 elements. Thus, the adaptive sparse-grid with $\tau = 10^{-8}$ is as accurate as the full-grid solution, but with substantially fewer degrees of freedom.

In Figure 5.4.4 we plot the the relative change in total energy for both collisionalities discussed above. The relative change in the total energy is at the level of GMRES tolerance for the full-grid simulation. For the adaptive sparse-grid methods, the relative change in the total energy decreases with the size of the threshold τ used; we expect this trend to continue until the GMRES tolerance pollutes the energy conservation. We hypothesize the improvement in the relative energy conservation of the adaptive sparse-grid with $\tau = 10^{-8}$ when compared with the full-grid (as seen in Figure 5.4.4b) is due to the multiwavelets not being used in the Chu reduction discretization.

The number of GMRES iterations varies between three and five for the sparse-grid runs.

6. Summary and Outlook

In this work, we presented an adaptive sparse-grid DG method for the the VPLB model on a slab geometry. The results of this project utilized the Adaptive Sparse-Grid Discretization (ASGarD) codebase. As demonstrated in Section 5, the adaptive sparse-grid method significantly decreases the storage cost of DG numerical approximations without compromising accuracy. Moreover, the adaptive sparse-grid method was able to capture physically relatively features of the distribution without the use of model specific error indicators. The results also indicate that standard sparse-grids in velocity space, i.e. the mixed-grid formulation, accurately captures low-order moments of the distribution, but are only slightly better when compared asymptotically against the full-grid for higher-order moments. This necessitates further research into using in a coordinate system that more beneficially captures the radial behavior of the Maxwellian, e.g. spherical-polar coordinates, or allowing some form of adaptivity in the mixed-grid. Other future plans include the expansion of the adaptive sparse-grid tests to full $3x3v$ phase-space simulations, efficient implementations of PDE operators on a sparse-grid basis, implementation of more efficient preconditioners in the sparse-grid framework, and the preservation of key quantities such as positivity of the discrete distribution in the multiwavelet basis.

References

[1] Abdelmalik, M., van Brummelen, E., 2016. An entropy stable discontinuous Galerkin finite-element moment method for the Boltzmann equation. Computers & Mathematics with Applications 72, 1988–1999.

[2] Adams, M.L., 2001. Discontinuous finite element transport solutions in thick diffusive problems. Nuclear science and engineering 137, 298–333.

[3] Alekseenko, A., Gimelshein, N., Gimelshein, S., 2012. An application of discontinuous Galerkin space and velocity discretisations to the solution of a model kinetic equation. International Journal of Computational Fluid Dynamics 26, 145–161.

[4] Alpert, B.K., 1993. A class of bases in L^2 for the sparse representation of integral operators. SIAM journal on Mathematical Analysis 24, 246–262.

[5] Arber, T., Vann, R., 2002. A critical comparison of Eulerian-grid-based Vlasov solvers. Journal of Computational Physics 180, 339–357.

[6] Ascher, U.M., Ruuth, S.J., Spiteri, R.J., 1997. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Applied Numerical Mathematics 25, 151–167.

[7] Balder, R., Zenger, C., 1996. The solution of multidimensional real Helmholtz equations on sparse grids. SIAM Journal on Scientific Computing 17, 631–646.

[8] Barth, T., 2006. On discontinuous Galerkin approximations of Boltzmann moment systems with levermore closure. Computer methods in applied mechanics and engineering 195, 3311–3330.
[9] Bellman, R., Kalaba, R., 1959. On adaptive control processes. IRE Transactions on Automatic Control 4, 1–9.
[10] Bennoune, M., Lemou, M., Mieussens, L., 2008. Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible navier–stokes asymptotics. Journal of Computational Physics 227, 3781–3803.
[11] Birdsall, C.K., Langdon, A.B., 2018. Plasma physics via computer simulation. CRC press.
[12] Bungartz, H.J., Griebel, M., 2004a. Sparse grids. Acta numerica 13, 147–269.
[13] Bungartz, H.J., Griebel, M., 2004b. Sparse grids. Acta Numer. 13, 147–269. doi:10.1017/S0962492904000182.
[14] Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A., 2007. Spectral methods: fundamentals in single domains. Springer Science & Business Media.
[15] Carrillo, J.A., Hu, J., Wang, L., Wu, J., 2020. A particle method for the homogeneous Landau equation. Journal of Computational Physics: X 7, 100066.
[16] Chan, J., Del Rey Fernández, D.C., Carpenter, M.H., 2019. Efficient entropy stable Gauss collocation methods. SIAM Journal on Scientific Computing 41, A2938–A2966.
[17] Chen, G., Chacón, L., Barnes, D.C., 2011. An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm. Journal of Computational Physics 230, 7018–7036.
[18] Chen, T., Shu, C.W., 2020. Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes. CSIAM Transactions on Applied Mathematics 1, 1–52.
[19] Cheng, Y., Christlieb, A.J., Zhong, X., 2014. Energy-conserving discontinuous Galerkin methods for the Vlasov–Ampère system. Journal of Computational Physics 256, 630–655.
[20] Cheng, Y., Gamba, I.M., Majorana, A., Shu, C.W., 2011. A brief survey of the discontinuous Galerkin method for the Boltzmann-Poisson equations. SeMA Journal 54, 47–64.
[21] Chkifa, A., Cohen, A., Schwab, C., 2014. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. Found. Comput. Math. 14, 601–633. doi:10.1007/s10208-013-9154-z.
[22] Chu, C., 1965. Kinetic-theoretic description of the formation of a shock wave. The Physics of Fluids 8, 12–22.
[23] Chu, R., Endeve, E., Hauck, C.D., Mezzacappa, A., 2019. Realizability-preserving DG-IMEX method for the two-moment model of fermion transport. Journal of Computational Physics 389, 62–93.
[24] Cockburn, B., Shu, C.W., 2001. Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. Journal of Scientific Computing 16, 173–261.
[25] Crestetto, A., Crouseilles, N., Lemou, M., 2012. Kinetic/fluid micro-macro numerical schemes for Vlasov-Poisson-BGK equation using particles. Kinetic and Related Models 5, 787.
[26] DeCaria, V., Hauck, C., Schnake, S., 2024. Asymptotic preserving discontinuous Galerkin methods for a linear Boltzmann semiconductor model. SIAM Journal on Numerical Analysis (accepted) .
[27] Einkemmer, L., Hu, J., Ying, L., 2021. An efficient dynamical low-rank algorithm for the Boltzmann-BGK equation close to the compressible viscous flow regime. SIAM Journal on Scientific Computing 43, B1057–B1080.
[28] Einkemmer, L., Lubich, C., 2018. A low-rank projector-splitting integrator for the Vlasov–Poisson equation. SIAM Journal on Scientific Computing 40, B1330–B1360.
[29] Endeve, E., Hauck, C., 2022. Conservative DG method for the micro-macro decomposition of the Vlasov–Poisson–Lenard–Bernstein model. Journal of Computational Physics 462, 111227.
[30] Endeve, E., Hauck, C.D., Xing, Y., Mezzacappa, A., 2015. Bound-preserving discontinuous Galerkin methods for conservative phase space advection in curvilinear coordinates. Journal of Computational Physics 287, 151–183.
[31] Filbet, F., Jin, S., 2010. A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources. Journal of Computational Physics 229, 7625–7648.
[32] Francisquez, M., Bernard, T.N., Mandell, N.R., Hammett, G.W., Hakim, A., 2020. Conservative discontinuous Galerkin scheme of a gyro-averaged Dougherty collision operator. Nuclear Fusion 60, 096021.
[33] Friedrich, L., Schnücke, G., Winters, A.R., Fernández, D.C.D.R., Gassner, G.J., Carpenter, M.H., 2019. Entropy stable space–time discontinuous Galerkin schemes with summation-by-parts property for hyperbolic conservation laws. Journal of Scientific Computing 80, 175–222.
[34] Fu, G., Shu, C.W., 2019. Optimal energy-conserving discontinuous Galerkin methods for linear symmetric hyperbolic systems. Journal of Computational Physics 394, 329–363.
[35] Garrett, C.K., Hauck, C.D., 2018. A fast solver for implicit integration of the Vlasov–Poisson system in the Eulerian framework. SIAM Journal on Scientific Computing 40, B483–B506.
[36] Gerstner, T., Griebel, M., 1998. Numerical integration using sparse grids. Numerical algorithms 18, 209–232.
[37] Green, D.L., D'azevedo, E., Elwasif, W., Lopez, M., Mu, L., 2019. Adaptive sparse grid discretization (ASGarD). doi:10.11578/dc.20201125.5.
[38] Guermond, J.L., Kanschat, G., 2010. Asymptotic analysis of upwind discontinuous Galerkin approximation of the radiative transport equation in the diffusive limit. SIAM Journal on Numerical Analysis 48, 53–78.
[39] Guo, W., Cheng, Y., 2016. A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations. SIAM Journal on Scientific Computing 38, A3381–A3409.
[40] Guo, W., Cheng, Y., 2017. An adaptive multiresolution discontinuous Galerkin method for time-dependent transport equations in multidimensions. SIAM Journal on Scientific Computing 39, A2962–A2992.
[41] Guo, W., Qiu, J.M., 2022. A low rank tensor representation of linear transport and nonlinear Vlasov solutions and their associated flow maps. Journal of Computational Physics 458, 111089.
[42] Haar, A., 1909. Zur theorie der orthogonalen funktionensysteme. Georg-August-Universitat, Gottingen.
[43] Hakim, A., Francisquez, M., Juno, J., Hammett, G., 2020. Conservative discontinuous Galerkin schemes for nonlinear

Dougherty-Fokker-Planck collision operators. Journal of Plasma Physics 86, 905860403.
[44] Hemker, P.W., 1995. Sparse-grid finite-volume multigrid for 3d-problems. Advances in computational mathematics 4, 83–110.
[45] Hirvijoki, E., Kraus, M., Burby, J.W., 2018. Metriplectic particle-in-cell integrators for the Landau collision operator. arXiv preprint arXiv:1802.05263 .
[46] Hong, X., Qiu, J.M., 2022. A generalized Eulerian-Lagrangian discontinuous Galerkin method for transport problems. Journal of Computational Physics 464, 111160.
[47] Hu, G., Krommes, J.A., 1994. Generalized weighting scheme for δf particle-simulation method. Physics of plasmas 1, 863–874.
[48] Huang, J., Cheng, Y., 2020. An adaptive multiresolution discontinuous Galerkin method with artificial viscosity for scalar hyperbolic conservation laws in multidimensions. SIAM Journal on Scientific Computing 42, A2943–A2973.
[49] Huang, J., Guo, W., Cheng, Y., 2023. Adaptive sparse grid discontinuous Galerkin method: review and software implementation. Communications on Applied Mathematics and Computation , 1–32.
[50] Jakeman, J.D., Archibald, R., Xiu, D., 2011. Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. J. Comput. Phys. 230, 3977–3997. doi:10.1016/j.jcp.2011.02.022.
[51] Jakeman, J.D., Narayan, A., Xiu, D., 2013. Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions. J. Comput. Phys. 242, 790–808. doi:10.1016/j.jcp.2013.02.035.
[52] Jakeman, J.D., Roberts, S.G., 2012. Local and dimension adaptive stochastic collocation for uncertainty quantification, in: Sparse Grids and Applications. Springer, pp. 181–203. doi:10.1007/978-3-642-31703-3_9.
[53] Khaziev, R., Curreli, D., 2018. hpic: A scalable electrostatic particle-in-cell for plasma–material interactions. Computer Physics Communications 229, 87–98.
[54] Kormann, K., Sonnendrücker, E., 2016. Sparse grids for the Vlasov–Poisson equation, in: Sparse Grids and Applications-Stuttgart 2014. Springer, pp. 163–190.
[55] Larsen, E.W., Morel, J., 1989. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii. Journal of Computational Physics 83, 212–236.
[56] Leentvaar, C., Oosterlee, C.W., 2006. Pricing multi-asset options with sparse grids and fourth order finite differences, in: Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2005, the 6th European Conference on Numerical Mathematics and Advanced Applications Santiago de Compostela, Spain, July 2005, Springer. pp. 975–983.
[57] Lenard, A., Bernstein, I., 1958. Plasma Oscillations with Diffusion in Velocity Space. Physical Review 112, 1456–1459.
[58] Ma, X., Zabaras, N., 2009. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. J. Comput. Phys. 228, 3084–3113. doi:10.1016/j.jcp.2009.01.006.
[59] Mishra, S., Schwab, C., 2012. Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data. Mathematics of computation 81, 1979–2018.
[60] Morrow, Z., Stoyanov, M., 2020. A method for dimensionally adaptive sparse trigonometric interpolation of periodic functions. SIAM Journal on Scientific Computing 42, A2436–A2460.
[61] Narayan, A., Jakeman, J.D., 2014. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. SIAM J. Sci. Comput. 36, A2952–A2983. doi:10.1137/140966368.
[62] Nobile, F., Tamellini, L., Tempone, R., 2016. Convergence of quasi-optimal sparse-grid approximation of Hilbert-space-valued functions: application to random elliptic PDEs. Numer. Math. 134, 343–388. doi:10.1007/s00211-015-0773-y.
[63] Olbrant, E., Hauck, C.D., Frank, M., 2012. A realizability-preserving discontinuous Galerkin method for the M1 model of radiative transfer. Journal of Computational Physics 231, 5612–5639.
[64] Pareschi, L., Russo, G., 2005. Implicit-Explicit Runge-Kutta Schemes and Application to Hyperbolic Systems with Relaxation. Journal of Scientific Computing 25, 129–155.
[65] Pazner, W., 2021. Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. Computer Methods in Applied Mechanics and Engineering 382, 113876.
[66] Pflüger, D., 2010. Spatially Adaptive Sparse Grids for High-Dimensional Problems. Verlag Dr. Hut, München. URL: http://www5.in.tum.de/pub/pflueger10spatially.pdf.
[67] Pflüger, D., 2012. Spatially adaptive refinement, in: Sparse Grids and Applications. Springer, pp. 243–262. doi:10.1007/978-3-642-31703-3_12.
[68] Pflüger, D., Peherstorfer, B., Bungartz, H.J., 2010. Spatially adaptive sparse grids for high-dimensional data-driven problems. J. Complexity 26, 508–522. doi:10.1016/j.jco.2010.04.001.
[69] Qiu, J.M., Shu, C.W., 2011. Positivity preserving semi-lagrangian discontinuous Galerkin formulation: theoretical analysis and application to the Vlasov–Poisson system. Journal of Computational Physics 230, 8386–8409.
[70] Reed, W., Hill, T., 1973. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479. Los Alamos National Laboratory.
[71] Rivière, B., 2008. Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation. SIAM.
[72] Rodgers, A., Venturi, D., 2020. Step-truncation integrators for evolution equations on low-rank tensor manifolds. CoRR .
[73] Schwab, C., Süli, E., Todor, R.A., 2008. Sparse finite element approximation of high-dimensional transport-dominated diffusion problems. ESAIM: Mathematical Modelling and Numerical Analysis 42, 777–819. doi:10.1051/m2an:2008027.
[74] Shen, J., Wang, L.L., 2010. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. SIAM Journal on Numerical Analysis 48, 1087–1109.
[75] Shen, J., Yu, H., 2010. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. SIAM Journal on Scientific Computing 32, 3228–3250.
[76] Sheng, Q., Hauck, C., 2021. Uniform convergence of an upwind discontinuous Galerkin method for solving scaled discrete-

ordinate radiative transfer equations with isotropic scattering. Mathematics of Computation 90, 2645–2669.
[77] Smolyak, S.A., 1963. Quadrature and interpolation formulas for tensor products of certain classes of functions, in: Doklady Akademii Nauk, Russian Academy of Sciences. pp. 1042–1045.
[78] Sod, G.A., 1978. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. Journal of Computational Physics 27, 1–31.
[79] Stoyanov, M., 2018. Adaptive sparse grid construction in a context of local anisotropy and multiple hierarchical parents, in: Sparse Grids and Applications – Miami 2016. Springer, pp. 175–199. doi:10.1007/978-3-319-75426-0_8.
[80] Stoyanov, M., Seleson, P., Webster, C., 2017. Predicting fracture patterns in simulations of brittle materials under variable load and material strength, in: 19th AIAA Non-Deterministic Approaches Conference, p. 1326. doi:10.2514/6.2017-1326.
[81] Stoyanov, M.K., Webster, C.G., 2016. A dynamically adaptive sparse grids method for quasi-optimal interpolation of multidimensional functions. Comput. Math. Appl. 71, 2449–2465. doi:10.1016/j.camwa.2015.12.045.
[82] Tang, M., Wang, Y., 2017. An asymptotic preserving method for strongly anisotropic diffusion equations based on field line integration. Journal of Computational Physics 330, 735–748.
[83] Tao, Z., Guo, W., Cheng, Y., 2019. Sparse grid discontinuous Galerkin methods for the Vlasov-Maxwell system. Journal of Computational Physics: X 3, 100022.
[84] Wang, Z., Tang, Q., Guo, W., Cheng, Y., 2016. Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations. Journal of Computational Physics 314, 244–263.
[85] Xing, Y., Chou, C.S., Shu, C.W., 2013. Energy conserving local discontinuous Galerkin methods for wave propagation problems. Inverse Problems & Imaging 7.
[86] Xiong, T., Jang, J., Li, F., Qiu, J.M., 2015. High order asymptotic preserving nodal discontinuous Galerkin IMEX schemes for the BGK equation. Journal of Computational Physics 284, 70–94.
[87] Yan, G., Kaur, S., Banks, J.W., Hicken, J.E., 2023. Entropy-stable discontinuous Galerkin difference methods for hyperbolic conservation laws. Journal of Computational and Applied Mathematics 422, 114885.
[88] Zhang, W., Xing, Y., Endeve, E., 2022. Energy conserving and well-balanced discontinuous Galerkin methods for the euler–Poisson equations in spherical symmetry. Monthly Notices of the Royal Astronomical Society 514, 370–389.
[89] Zhang, X., Shu, C.W., 2011. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. Proc. Roy. Soc. London A: Math., Phys. and Eng. Sci. 467, 2752–2776.