

ABSTRACT

Title of dissertation: OPTIMIZATION TECHNIQUES FOR
ENTROPY-BASED MOMENT MODELS
OF LINEAR TRANSPORT

Graham West Alldredge,
Doctor of Philosophy, 2012

Dissertation directed by: Professor André Tits
Department of Electrical and Computer
Engineering

Kinetic equations are used to model many physical phenomena, including gas dynamics, semiconductors, radiative transport, and more. However, high dimensionality of the domain of definition of the system makes simulation difficult. The entropy-based moment closure model of the kinetic equation reduces the dimension of the domain and has attractive theoretical and practical properties, but most implementations have avoided numerically solving the defining optimization problem. We use the linear one-dimensional slab-geometry model to expose the main challenges in the use of numerical optimization then propose an isotropic regularization and describe the benefits of using fixed quadrature. A numerical technique using adaptive polynomial bases in the optimization algorithm is also tested. We develop manufactured solutions to test our algorithm and also present its performance on two standard test problems.

OPTIMIZATION TECHNIQUES FOR ENTROPY-BASED
MOMENT MODELS OF LINEAR TRANSPORT

by

Graham West Alldredge

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor André Tits, Chair/Advisor
Dr. Cory Hauck, Co-Advisor
Professor Dianne O'Leary
Professor Isaak Mayergoyz
Professor C. David Levermore

Acknowledgments

I owe my first thanks to my advisor, Professor André Tits, for trusting me with an unusual (at least for someone with my background and professed interests) project. André's availability and supportiveness are legendary, and his attention to detail was something I needed.

Cory Hauck might deserve an even bigger thanks. I entered the project knowing as much about kinetic theory as a fourth-grader, and he patiently went through the basics with me more than once. Cory also delivered an appropriately timed and swift kick to my seat cushion when I started to take advantage of the freedoms of graduate school a little too generously. This was more than he signed up for, but I am grateful he cares that much.

This thesis would be a lot better if I had started talking to Dianne O'Leary sooner. In addition to her well-known technical prowess, her high but realistic expectations were always implicitly clear, and this motivated me to be well prepared and thorough in my work.

I would also like to thank Professors Isaak Mayergoyz and C. David Levermore for sacrificing some of their valuable time to serve on my committee.

Finally, I would like to thank the Department of Energy for financially supporting our work under Grant DESC0001862.

Table of Contents

List of Tables	v
List of Figures	vi
1 Introduction	1
2 The Linear Kinetic Equation and the Entropy-Based Moment Closure	6
2.1 The linear kinetic equation	6
2.2 Entropy-based moment closures	7
2.3 Realizability	10
2.4 An \mathcal{R} -invariant numerical solver	13
3 Challenges in the Optimization Problem	16
3.1 Basics of the optimization	16
3.2 Optimization near the realizable boundary	19
3.2.1 The M_1 model	21
3.2.2 Challenges in higher-order models	26
3.2.3 Difficulties satisfying the γ tolerance	31
3.2.4 When the optimization breaks down	33
4 Optimization Techniques	36
4.1 Fixed quadrature	36
4.2 Adaptive polynomial basis	39
4.2.1 Implementation details	44
4.2.2 Static results	46
4.2.3 Alternative orthogonalizations	50
4.3 Isotropic regularization	52
4.4 The final optimization algorithm	55
5 Numerical Results	58
5.1 Manufactured solutions	59
5.1.1 Comparing adaptive-basis and fixed-basis methods	63
5.1.2 Choosing ε_γ according to the accuracy and computational effort trade-off	69
5.1.3 The effect of quadrature on accuracy	71
5.2 Standard Test problems	72
5.2.1 Plane source	73
5.2.1.1 M_{15} simulation	73
5.2.1.2 Convergence in N	74
5.2.1.3 Effects of excessive regularization	75
5.2.2 Two-beam instability	79
5.2.2.1 M_{15} simulation	80
5.2.2.2 Convergence in N	81
5.2.2.3 Effects of excessive regularization	81

6	Conclusions and Directions for Future Work	86
A	Details and Properties of the Numerical Scheme	89
A.1	The numerical scheme	89
A.2	Proof of \mathcal{R} -invariance of the numerical scheme	92
	Bibliography	96

List of Tables

4.1	Boundary moments used for tests in Table 4.2 below. The deltas in $\mathbf{u}^{(5)}$ are at the 4-th, 7-th, 10-th, 13-th, 14-th, and 15-th nodes of the twenty-point Curtis-Clenshaw quadrature over the interval $[-1, 0]$. In $\mathbf{u}^{(6)}$, $\nu_1, \nu_3, \dots, \nu_6$ are at the 4-th, 9-th, 12-th, 15-th, and 17-th nodes of the same quadrature, while ν_2 is the 54-th node of the 153-point Curtis-Clenshaw quadrature over the interval $[-1, 0]$	48
4.2	For each $\mathbf{u}^{(k)}$ in Table 4.1 above, this table shows the largest value of ℓ for which the optimization algorithm (with either adaptive basis (A) or fixed basis (F)) can find approximately optimal multipliers $\bar{\alpha}(\mathbf{u}^{(k)})$ with tolerances $\tau = 10^{-8}$ and $\varepsilon_\gamma = 0.01$. $N = 12$	49
4.3	Same as Table 4.2 above, but here we compare the use of Cholesky factorization (C) to that of modified Gram-Schmidt (GS) in for the adaptive-basis method.	51
5.1	Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 5$ and $\lambda = 1/3$	64
5.2	Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$	64
5.3	Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 5$ and $\lambda = 1/3$	64
5.4	Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$	65
5.5	Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 5$ and $\lambda = 1/3$	66
5.6	Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$	66
5.7	Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 5$ and $\lambda = 1/3$	67
5.8	Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$	67
5.9	L^1 and L^∞ errors for adaptive-basis (A) and fixed-basis (F) methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$	68
5.10	L^1 and L^∞ errors for adaptive-basis (A) and fixed-basis (F) methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$	69

5.11	L^1 errors for n_Q tests.	72
------	---------------------------------------	----

List of Figures

3.1	Illustrating difficulties in the optimization in M_1	23
3.2	Examining the multipliers given in (3.27b).	28
3.3	Examining the multipliers given in (3.29).	30
3.4	The stopping criterion quantities for the optimization algorithm for (3.30)	32
3.5	The norm of the gradient and the condition number of the Hessian when the optimization algorithm is applied to the moments in (3.31).	34
3.6	The ansätze and polynomials for the last three iterations when the optimization algorithm is applied to the moments in (3.31).	35
4.1	Illustrating $\mathcal{R}_Q _{u_0=1}$ for M_2 . The green indicates $\mathcal{R}_Q _{u_0=1}$, and the blue indicates $\mathcal{R} _{u_0=1} \setminus \mathcal{R}_Q _{u_0=1}$	38
4.2	Comparing the fixed-basis method to the adaptive-basis method on the moments given in (3.27a). (Even at the bright pixel in the adaptive-basis figure, the adaptive-basis method converged in seventy-seven iterations.)	46
4.3	The solution for $\mathbf{u}_{27}^{(1)}$. The vertical red dashed lines indicate the locations of the deltas generating $\mathbf{u}^{(1)}$	49
4.4	The set of normalized realizable moments $\mathcal{R} _{u_0=1}$ in M_2 and the paths we take to the boundary in the moments defined in (4.19).	54
4.5	The condition number of the Hessian near the boundary in M_1 and M_2	54
5.1	The local particle concentrations for the two manufactured solutions for 400-cell simulations.	62
5.2	L^1 errors and two computation time estimates (in seconds) for several values of ε_γ . Here $k_0 = 5$ and $\lambda = 1/3$	70
5.3	A simulation of the M_{15} model of the plane source problem with $N_x = 3000$ cells.	76
5.4	The minimum value of $\rho_{\partial\mathcal{R}}(\mathbf{u}(x, t))$ over the space-time mesh for $t \leq 3$ as the cell size is decreased.	77
5.5	The convergence of the plane-source simulation as $N \rightarrow \infty$	77
5.6	The simulation with excessive regularization on the plane source problem. Here $k_0 = 2$, and $\lambda = 0$, and $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$	78
5.7	A simulation of the M_{15} model of the two-beam instability with $N_x = 1000$ cells.	83
5.8	Convergence of individual moments in the M_{15} beams simulation.	84
5.9	The convergence of transient profiles of the two-beam instability simulation as $N \rightarrow \infty$	84
5.10	Steady-state M_N beams solutions.	85

5.11 The simulation with excessive regularization on the two-beam instability. Here $k_0 = 2$, and $\lambda = 0$, and $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$. 85

Chapter 1

Introduction

Kinetic equations are used in the modeling of many physical phenomena including gas dynamics, semiconductors, quantum fluids, radiative transport, and phonon transport in solids. The linear kinetic equation for unit-speed particles is a simplified model which has applications in neutron transport [1]. The numerical simulation of kinetic systems can be difficult for many reasons, not the least of which is the high dimensionality of the domain of definition. A kinetic description for the unit-speed particles depends on position, angle, and time, where in many real problems position is three-dimensional and the angle lies on the unit sphere. The construction of a grid in these six total dimensions is generally not tractable, and so researchers often look for a reduction in the dimension of the domain.

Currently popular numerical methods for simulating linear kinetic equations include the P_N models, discrete-ordinate methods, and Monte Carlo simulations. Solutions using P_N methods, which are truncated spherical harmonic expansions in the angle variable, can have undesirable non-physical artifacts such as distributions with negative values [1,2]. Discrete-ordinate methods, defined using a quadrature in the angle variable and commonly known as S_N , suffer from ‘ray effects’ because they lack rotational symmetry [1,3]. Implicit Monte Carlo methods [4] remain popular but are, without great computational effort, fundamentally vulnerable to the effects

of statistical noise.

The end-user scientists or engineers are often only interested in a few macroscopic quantities obtained from the full kinetic descriptions [5]. Moments, such as energy or particle density and radiative fluxes and pressures, are examples of such quantities. However, according to the kinetic equation the exact evolution of a finite set of moments does not form a closed system: the evolution of moments up to order N depends on the moments up to order $N + 1$. The *closure* problem is that of approximating the moment of order $N + 1$ from the known moments, those of order up to N .

One way to close the moment system is to estimate the full kinetic description from the known N moments. This estimated distribution is called an *ansatz*, but because a finite set of moments generally does not uniquely determine a full distribution, criteria must be given for which distribution to select.

From an information theoretic perspective, the known moments are partial knowledge of an underlying distribution, and the most likely (and, respectful of the limited knowledge, least committal) distribution is the one of minimum¹ entropy [6]. From this idea Levermore [7] developed a hierarchy of entropy-based moment models (commonly referred to as the M_N models) which reconstruct the kinetic description by solving the constrained, convex optimization problem which minimizes entropy while matching the moments. Levermore showed that these models enjoy attractive theoretical properties including positivity of the solution, entropy dissipation, and

¹This corresponds to the *maximum* physical entropy due to the sign convention we choose in this work to define the entropy measure.

hyperbolicity of the moment system.

One serious obstacle to the implementation of entropy-based moment closures is that, for some moments, the defining optimization problem may not have a solution even though the moments are *realizable* (that is, they are associated with some nonnegative distribution). This problem was first exposed in [8], and recent work [9] has more generally characterized this set of problems. Indeed in [9] the set of such ‘degenerate’ moments is shown to be ‘small in both a topological and a measure-theoretic sense.’

Some work has considered computing the entropy-based solution while bypassing the optimization. Eddington factors [5, 10, 11] can be used to calculate the entropy-based closure without determining the ansatz. But beyond the first-order M_1 model, an explicit expression for the Eddington factor is difficult to derive. In [11], to simulate the second-order M_2 model the authors construct a look-up table directly mapping low-order moments to the third-order moment needed to close the model. Unfortunately, the extension of such a method to higher-order models is impractical, and higher-order models are needed to mitigate the non-physical artifacts associated with the moment solution [12].

The burden of solving the optimization problem at every point on the space-time mesh makes the M_N model generally impractical for serial computer implementation, but this cost is mitigated significantly in the emerging paradigm of massively parallel computing where data communication is much more of a bottleneck than floating-point operations. In a moment model, the optimization problems in each spatial cell are independent and thus ideal for parallelization. Furthermore, the

compactness of the entropy-based description means that only a small amount of data needs to be communicated between computational cells in the solution of the moment system. Thus, the timing is right to reassess the expense and feasibility of using a numerical optimization algorithm in the PDE solver.

The numerical solution of single, moment-constrained minimum-entropy optimization problems has received much attention in the literature. The ‘iterative scaling’ algorithm first proposed in [13] and later improved in [14, 15] has been useful in areas as diverse as computational linguistics [16] and ecology [17]. It is a coordinate-descent method which, while appropriate for single large problems, is much too slow to solve the massive number of problems in the M_N model even in a parallel implementation. A promising algorithm was proposed by Abramov in [18, 19] which uses an adaptive polynomial basis to work around inherent poor conditioning of the problem. The adaptive-basis method was successfully used to solve single two-dimensional problems over an unbounded integration domain. The applicability of this method in M_N models is unclear.

We begin by introducing the linear transport equation in slab geometry, its entropy-based moment model, and a numerical solver for this moment model in Chapter 2. Next we narrow our focus in Chapter 3 to the optimization and discuss why its numerical solution can be challenging. Once these difficulties are exposed, we explain and test some ways to address them in Chapter 4, including fixed quadrature, adaptive-basis methods inspired by Abramov’s method, and isotropic regularization. We test our resulting algorithm numerically on manufactured solutions and two standard test problems extensively in Chapter 5. Finally, we draw conclusions and

propose directions for future work in Chapter 6.

Chapter 2

The Linear Kinetic Equation and the Entropy-Based Moment

Closure

In this chapter we review the linear one-dimensional kinetic equation we use for our experiments and introduce the entropy-based moment closure for this model. We also introduce the key concept of realizability and describe the numerical PDE solver we implement.

2.1 The linear kinetic equation

Following [12], we consider a simplified one-dimensional model of imaginary unit-speed particles (somewhat similar to neutrons) in slab geometry. This model avoids degenerate problems [8,9] (in e.g. gas dynamics models) because the velocity variable lies in a compact space and thus the optimization problem always has a solution.

The kinetic model tracks a non-negative density $F = F(x, \mu, t)$ whose independent variables here are the scalar spatial coordinate $x \in (x_L, x_R)$ along the direction perpendicular to the slab, the angle variable $\mu \in [-1, 1]$ corresponding to the cosine of the angle between the x -axis and the direction of particle travel, and time $t \geq 0$. The particles are absorbed by or scattered isotropically off of the background medium, and these interactions are modeled by non-negative scattering

and absorption interaction coefficients, $\sigma_s(x)$, and $\sigma_a(x)$ respectively. We define the total interaction coefficient $\sigma_t(x) := \sigma_s(x) + \sigma_a(x)$. This leads to the kinetic equation of the form

$$\partial_t F + \mu \partial_x F + \sigma_t F = \frac{\sigma_s}{2} \langle F \rangle + S. \quad (2.1)$$

where $S = S(x, \mu, t)$ is an external source. For the rest of this work, except in Section 5.1, we assume $S \equiv 0$. The angle brackets denote integration over μ : for any integrable function $g = g(\mu)$,

$$\langle g \rangle := \int_{-1}^1 g(\mu) d\mu. \quad (2.2)$$

Equation (2.1) is supplemented by boundary and initial conditions

$$F(x_L, \mu, t) = F_L(\mu, t), \quad \mu > 0, t > 0, \quad (2.3a)$$

$$F(x_R, \mu, t) = F_R(\mu, t), \quad \mu < 0, t > 0, \quad (2.3b)$$

$$F(x, \mu, 0) = F_0(x, \mu), \quad \mu \in [-1, 1], x \in [x_L, x_R], \quad (2.3c)$$

where F_0 , F_L , and F_R are given.

2.2 Entropy-based moment closures

Moments $\mathbf{u} = \mathbf{u}(x, t)$ are defined by angular averages with respect to basis polynomials $\mathbf{m} = \mathbf{m}(\mu)$. We follow the common convention [1] by choosing \mathbf{m} to be the first $N + 1$ Legendre polynomials, which are orthogonal on $[-1, 1]$ with respect

to $L^2(d\mu)$. Exact equations for the moments

$$\mathbf{u}(x, t) = [u_0, \dots, u_N]^T := \langle \mathbf{m}F(x, \cdot, t) \rangle \quad (2.4)$$

are found by multiplying the kinetic equation (2.1) by \mathbf{m} and integrating over all angles. This gives the system

$$\partial_t \mathbf{u} + \partial_x \langle \mu \mathbf{m}F \rangle + \sigma_t \mathbf{u} = \sigma_s Q \mathbf{u}, \quad (2.5)$$

where the $(N + 1) \times (N + 1)$ matrix Q is given by

$$Q_{lm} = \delta_{lm} \delta_{l0}, \quad (2.6)$$

δ_{lm} being the Kronecker δ , so that $Q \mathbf{u} = [u_0, \dots, 0]^T$.

The flux term $\langle \mu \mathbf{m}F \rangle$ cannot be computed from \mathbf{u} , so (2.5) is not closed. We close this system by choosing an *ansatz* to approximate F and substituting it into (2.5). Entropy-based methods specify the ansatz for a given (x, t) as the solution to the constrained, strictly convex optimization problem

$$\underset{g}{\text{minimize}} \quad \langle \eta(g) \rangle \quad \text{subject to} \quad \langle \mathbf{m}g \rangle = \mathbf{u}. \quad (2.7)$$

Here the minimization is with respect to $g: [-1, 1] \rightarrow \mathbb{R}$, and the *entropy* function

$\eta: \mathbb{R} \rightarrow \mathbb{R}$ is strictly convex. If a minimizer exists, it takes the form [7]

$$G_{\boldsymbol{\alpha}}(\mu) := \eta'_* (\boldsymbol{\alpha}^T \mathbf{m}(\mu)) , \quad (2.8)$$

where $\eta_*: \mathbb{R} \rightarrow \mathbb{R}$ is the Legendre dual of η , η'_* is its derivative, and the vector of Lagrange multipliers $\hat{\boldsymbol{\alpha}}(\mathbf{u}) \in \mathbb{R}^{N+1}$ (also called *dual variables*) solve the dual problem:

$$\hat{\boldsymbol{\alpha}}(\mathbf{u}) = \operatorname{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^{N+1}} \{ \langle \eta_*(\boldsymbol{\alpha}^T \mathbf{m}) \rangle - \boldsymbol{\alpha}^T \mathbf{u} \} . \quad (2.9)$$

We focus on the Maxwell-Boltzmann entropy $\eta(z) = z \log(z) - z$. Thus $\eta_*(y) = \eta'_*(y) = e^y$ and

$$G_{\boldsymbol{\alpha}} = \exp(\boldsymbol{\alpha}^T \mathbf{m}). \quad (2.10)$$

Then we define \mathbf{f} as the flux associated with the entropy-based ansatz:

$$\mathbf{f}(\mathbf{u}) := \langle \mu \mathbf{m} G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})} \rangle , \quad (2.11)$$

so that the entropy-based closure of (2.5) is

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) + \sigma_t \mathbf{u} = \sigma_s Q \mathbf{u} . \quad (2.12)$$

Correct boundary conditions for the moment system (which represents integrals of (2.3) over the entire μ space) are not obviously determined because kinetic data is only given for values of μ which correspond to incoming data. The issue of proper boundary conditions remains an open question, although some progress

has been made for linear systems [20–23]. We use ghost cells to implement the boundaries; see Appendix A.1.

The numerical scheme we use to solve (2.12) is introduced in Section 2.4.

2.3 Realizability

The feasible set of the primal (2.7) is nonempty only if the moments \mathbf{u} are *realizable*.

Definition 1. *Let the vector-valued function \mathbf{m} be given and let $L_+^1(d\mu)$ be the set of all non-negative Lebesgue integrable functions g such that $\langle g \rangle > 0$. A vector \mathbf{v} is said to be realizable (with respect to \mathbf{m}) if there exists a $g \in L_+^1(d\mu)$, such that $\langle \mathbf{m}g \rangle = \mathbf{v}$. The set of all realizable vectors is denoted by \mathcal{R} .*

The following theorem characterizes the set \mathcal{R} when the components of \mathbf{m} are monomials. It is a classical result in the theory of *reduced moments*; see, for example, [24] and references therein.

Theorem 1. *Let $\mathbf{p} = [1, \mu, \dots, \mu^N]^T$. A necessary and sufficient condition for a vector \mathbf{v} to be realizable with respect to \mathbf{p} is that*

1. *in the case that N is odd, the $(N + 1)/2 \times (N + 1)/2$ Hankel matrices B^\pm , defined by*

$$B_{kl}^\pm := v_{k+l} \pm v_{k+l+1}, \quad k, l \in \{0, \dots, (N - 1)/2\}, \quad (2.13)$$

are positive definite;

2. in the case that N is even, the $(N + 2)/2 \times (N + 2)/2$ Hankel matrix B^0 and the $N/2 \times N/2$ Hankel matrix B^1 , defined by

$$B_{kl}^0 := v_{k+l}, \quad k, l \in \{0, \dots, N/2\},$$

$$B_{kl}^1 := v_{k+l} - v_{k+l+2}, \quad k, l \in \{0, \dots, (N - 2)/2\},$$

are positive definite.

The realizability of moments with respect to any vector-valued function \mathbf{m} whose components form a basis for \mathbb{P}^N can be determined by simply applying a change of basis from \mathbf{m} to \mathbf{p} and then invoking Theorem 1.

We can use the Hankel matrices of Theorem 1 to define a number describing how close a vector of moments is to the boundary of realizability:

$$\rho_{\partial\mathcal{R}}(\mathbf{u}) := \begin{cases} \min(\lambda_{\min}(B^+), \lambda_{\min}(B^-)) & \text{if } N \text{ is odd} \\ \min(\lambda_{\min}(B^0), \lambda_{\min}(B^1)) & \text{if } N \text{ is even} \end{cases} \quad (2.14)$$

where B^+ and B^- or B^0 and B^1 are the Hankel matrices formed from the monomial moments associated with \mathbf{u} .

The next theorem characterizes the geometry of \mathcal{R} .

Theorem 2. *The set \mathcal{R} is a convex cone, i.e., for any moments $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{R}$ and nonnegative constants c_1 and c_2 , with $c_1 + c_2 > 0$, $c_1\mathbf{v}_1 + c_2\mathbf{v}_2 \in \mathcal{R}$. Furthermore, it is open.*

Proof. The fact that \mathcal{R} is a convex cone follows from the fact that $L_+^1(d\mu)$ is also a convex cone. Openness is a corollary of Theorem 3 below. \square

The issue of realizability raises many questions. Two are:

1. Does a solution to (2.7) exist for all $\mathbf{u} \in \mathcal{R}$?
2. Can we enforce invariance of \mathcal{R} in a numerical solution of (2.12)?

In general the answer to the first question is ‘no.’ The lack of existence is related to the fact that the constraints in (2.7) are not always continuous in the L^1 norm [8, 9, 25, 26]. However, for the case under consideration, the domain of integration is bounded and the components of \mathbf{m} are bounded on that domain. These properties ensure that L^1 continuity holds and a solution exists, leading to the following theorem.

Theorem 3 ([25]). *The function $\hat{\boldsymbol{\alpha}}$ which maps moments \mathbf{u} to dual variables $\boldsymbol{\alpha}$ via the solution of (2.9) is a smooth bijection from \mathcal{R} onto \mathbb{R}^{N+1} . Its inverse is the moment map $\hat{\mathbf{v}}$ given by*

$$\hat{\mathbf{v}}(\boldsymbol{\alpha}) = \langle \mathbf{m}G_{\boldsymbol{\alpha}} \rangle. \tag{2.15}$$

Proof. See [25] and also [8, 27, 28] for similar results. \square

Notwithstanding Theorem 3, realizability presents significant numerical challenges. Indeed, near the boundary of \mathcal{R} , the Hessian of the dual objective (2.9) is ill-conditioned at the solution. This is a consequence of the fact that, on the boundary itself, the constraint equations are uniquely solved by atomic measure—that is,

an ansatz made up of delta functions [29]. These challenges are discussed in detail in Chapter 3.

2.4 An \mathcal{R} -invariant numerical solver

We implement a numerical solution to (2.12) using a kinetic scheme which is second-order in both space and time [30]. In the context of entropy-based closures, the main benefit of this scheme is that it preserves realizability. In addition, it avoids the direct computation of eigenvalues and (approximate) Riemann solvers [31] which, for most entropy-based moment systems, is expensive due to the complicated relationship between \mathbf{u} and \mathbf{f} in (2.11).

Details of the numerical solver are in Appendix A, but we describe it briefly here. It is a second-order finite-volume scheme of the form

$$\partial_t \mathbf{u}_j + \frac{\mathbf{f}_{j+1/2} - \mathbf{f}_{j-1/2}}{\Delta x} + \sigma_v \mathbf{u}_j = \sigma_s Q \mathbf{u}_j \quad \text{for } j \in \{1, \dots, N_x\}. \quad (2.16)$$

Here \mathbf{u}_j approximates the cell average of $\mathbf{u}(x, t)$ over the interval $I_j := (x_{j-1/2}, x_{j+1/2})$, where $x_{j\pm 1/2} := x_j \pm \Delta x/2$. With an exact solution to the dual optimization problem, the numerical flux is given by

$$\mathbf{f}_{j+1/2} = \left\langle \mu \mathbf{m} \hat{G}_{j+1/2} \right\rangle, \quad (2.17)$$

where $\hat{G}_{j+1/2}$ is an approximation of the entropy ansatz at the cell edge $x_{j+1/2}$ based on a linear reconstruction of the cell averages $\hat{G}_{j'} := G_{\hat{\alpha}(\mathbf{u}_{j'})}$ on the stencil

$j' \in \{j - 1, j, j + 1, j + 2\}$ and a standard minmod-type limiter. The boundary conditions are implemented using ghost cells indexed by $j \in \{-1, 0, N_x + 1, N_x + 2\}$.

Time integration is performed using the second-order strong-stability-preserving Runge-Kutta (SSP-RK2) method [32], also known as Heun's method or the improved Euler method. We let \mathbf{u}_j^n denote the numerical solution at time step n in cell j .

This kinetic scheme invokes a numerical solution of the dual problem (2.9), so we use $\bar{\alpha}(\mathbf{u})$ to denote the approximate solution returned by the optimizer. The use of these approximate multipliers to compute the flux means we replace (2.17) by

$$\mathbf{f}_{j+1/2} := \langle \mu \mathbf{m} \bar{G}_{j+1/2} \rangle . \quad (2.18)$$

The fact that the dual problem can only be solved approximately must be taken into consideration when attempting to maintain realizability of the moments in the numerical solution. It turns out that the ratios between the ansätze $G_{\bar{\alpha}}$ and $G_{\hat{\alpha}}$ at each stage of the Runge-Kutta scheme play a key role. Therefore, for the n -th time step we define

$$\gamma^{j,(m)} := \left(\frac{\bar{G}_j^{(m)}}{\hat{G}_j^{(m)}} \right), \quad m \in \{0, 1\}, \quad \text{and} \quad \gamma_{\max} := \max_{\substack{m \in \{0, 1\} \\ j \in \{-1, \dots, N_x + 2\} \\ \mu \in [-1, 1]}} \{ \gamma^{j,(m)}(\mu) \}, \quad (2.19)$$

where the index m distinguishes the two stages of the Runge-Kutta method, $\hat{G}_j^{(m)} := G_{\hat{\alpha}(\mathbf{u}_j^{(m)})}$, and $\bar{G}_j^{(m)} := G_{\bar{\alpha}(\mathbf{u}_j^{(m)})}$.¹

The following theorem shows that with an appropriate time-step restriction

¹Here the dependence on n is suppressed for clarity.

and appropriate boundary conditions, the kinetic scheme preserves realizable moments.

Theorem 4. *Suppose that $\mathbf{u}_j^n \in \mathcal{R}$ for $j \in \{-1, \dots, N_x + 2\}$. If \mathbf{u}^{n+1} is defined via the kinetic scheme (A.2), the SSP-RK2 numerical integrator in time, and time-step restriction*

$$\gamma_{max} \frac{\Delta t}{\Delta x} \frac{2 + \theta}{2} + \sigma_t \Delta t < 1 \quad (2.20)$$

and if the moments in the ghost cells are realizable at each stage of the Runge-Kutta scheme, then $\mathbf{u}_j^{n+1} \in \mathcal{R}$ for $j \in \{1, \dots, N_x\}$.

The proof is in Appendix A.2.

Chapter 3

Challenges in the Optimization Problem

This chapter considers the numerical solution of the optimization and exposes what makes certain problems difficult. We expose why the problem is hard near the realizable boundary in M_1 and higher-order models, and present how these problems are manifested numerically in computation of the Newton direction and the stopping criterion. Much of the work in this chapter was published in [30].

3.1 Basics of the optimization

We denote the objective function in (2.9) and its gradient and Hessian, respectively, by

$$f(\boldsymbol{\alpha}) := \langle G_{\boldsymbol{\alpha}} \rangle - \boldsymbol{\alpha}^T \mathbf{u}, \quad (3.1)$$

$$\mathbf{g}(\boldsymbol{\alpha}) := \langle \mathbf{m} G_{\boldsymbol{\alpha}} \rangle - \mathbf{u} = \hat{\mathbf{v}}(\boldsymbol{\alpha}) - \mathbf{u}, \quad (3.2)$$

$$H(\boldsymbol{\alpha}) := \langle \mathbf{m} \mathbf{m}^T G_{\boldsymbol{\alpha}} \rangle. \quad (3.3)$$

Note that f is smooth and strictly convex and H is positive definite for all $\boldsymbol{\alpha}$.

We approach $\hat{\boldsymbol{\alpha}}(\mathbf{u})$ using Newton's method with an Armijo backtracking line search [33] to guarantee global convergence and fast (quadratic) local convergence.

Given an initial guess $\boldsymbol{\alpha}_0$, the iterates $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots$ are constructed by

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \chi^i \mathbf{d}(\boldsymbol{\alpha}_k), \quad k \in \{0, 1, 2, \dots\}, \quad (3.4)$$

where $\mathbf{d}(\boldsymbol{\alpha}_k) := -H^{-1}(\boldsymbol{\alpha}_k)\mathbf{g}(\boldsymbol{\alpha}_k)$ is the Newton direction at $\boldsymbol{\alpha}_k$, $\chi \in (0, 1)$ is the step-size parameter, i is the smallest non-negative integer such that

$$f(\boldsymbol{\alpha}_k + \chi^i \mathbf{d}(\boldsymbol{\alpha}_k)) \leq f(\boldsymbol{\alpha}_k) + \chi^i \xi \mathbf{g}(\boldsymbol{\alpha}_k)^T \mathbf{d}(\boldsymbol{\alpha}_k), \quad (3.5)$$

and $\xi \in (0, 1/2)$.

There are two conditions in the stopping criterion. Given parameters $\tau > 0$ and $\varepsilon_\gamma > 0$, we terminate the optimization process at the first iterate where $\boldsymbol{\alpha}_k$ satisfies

$$\|\mathbf{g}(\boldsymbol{\alpha}_k)\| \leq \tau, \quad \text{and} \quad (3.6a)$$

$$\exp(5\zeta \|\mathbf{d}(\boldsymbol{\alpha}_k)\|_1) \leq 1 + \varepsilon_\gamma \quad (3.6b)$$

where

$$\zeta := \max_{\mu \in [-1, 1]} \|\mathbf{m}(\mu)\|_\infty. \quad (3.7)$$

For the Legendre polynomials, $\zeta = 1$.

The first condition (3.6a) measures how close the primal constraints (2.7) are to being satisfied (see (3.2)) and bounds the flux error $\mathbf{f}(\mathbf{u}) - \mathbf{f}(\hat{\mathbf{v}}(\boldsymbol{\alpha}_k))$ [34].

The second condition is related to realizability of the moments generated by

the kinetic scheme. Due to (2.20) in Theorem 4, an upper bound on γ_{\max} is needed to ensure realizability with a reasonable time step $\Delta t = O(\Delta x)$. We conservatively bound $\gamma_k := G_{\alpha_k}/G_{\hat{\alpha}}$ using the inequality

$$\gamma_k(\mu) = \exp((\alpha_k - \hat{\alpha})^T \mathbf{m}(\mu)) \leq \exp(\zeta \|\alpha_k - \hat{\alpha}\|_1) . \quad (3.8)$$

Since $\hat{\alpha}$ is unknown, we make the approximation $\|\alpha_k - \hat{\alpha}\|_1 \approx \|\mathbf{d}(\alpha_k)\|_1$, which is a good asymptotic estimate because Newton's method locally converges quadratically. For our implementation, we further insert a factor of five inside the exponential in the right-hand side of (3.8) to increase confidence that we are bounding γ_k from above. This gives the second condition in (3.6b). With this conservative estimate of γ_k , we are typically able to use time steps of at least 90% of the maximum theoretical value for Δt —that is the value of Δt which corresponds to an exact solution of the dual problem and is computed from (2.20) with $\gamma_{\max} = 1$.

As mentioned above, we let $\bar{\alpha}(\mathbf{u})$ denote the approximate solution returned by the optimizer, namely the first iterate which satisfies (3.6).

The optimization routine is always applied to moments which are normalized by dividing by the zeroth-order moment u_0 (the local particle concentration) since

$$\frac{\mathbf{u}}{u_0} \equiv \frac{1}{u_0} \langle \mathbf{m} G_{\hat{\alpha}(\mathbf{u})} \rangle \equiv \langle \mathbf{m} \exp(-\log(u_0) + \hat{\alpha}(\mathbf{u})^T \mathbf{m}) \rangle . \quad (3.9)$$

Since $m_0 \equiv 1$, the constant $\log(u_0)$ can be absorbed into the zeroth-order multiplier

so that

$$\hat{\boldsymbol{\alpha}}(\mathbf{u}) = \hat{\boldsymbol{\alpha}}(\mathbf{u}/u_0) + (\log(u_0), 0, \dots, 0)^T. \quad (3.10)$$

This normalization makes it simpler to specify tolerances and analyze performance.

The PDE provides a natural warm start for the optimization algorithm: for $t > 0$, we can use the optimal multipliers from the last time step as initial conditions. At $t = 0$, we choose $\boldsymbol{\alpha}_0 := (\log(u_0/2), 0, \dots, 0)^T$, the multipliers of the isotropic distribution. Generally, we use the term ‘isotropic multipliers’ to refer to multipliers $\boldsymbol{\alpha}$ where $\alpha_1 = \alpha_2 = \dots = \alpha_N = 0$. We also use `eps` to denote the machine precision which, in double precision, is $2^{-52} \approx 2.22 \times 10^{-16}$.

3.2 Optimization near the realizable boundary

For most realizable moments, the dual problem (2.9) is easy to solve in only a few iterations with the warm start. The problem becomes difficult when the moments \mathbf{u} lie near $\partial\mathcal{R}$, the boundary of the set of realizable moments. These moments are associated with highly anisotropic distributions (or the vacuum state $G \equiv 0$), and often occur in the presence of strong sources or when particles enter a void. Refining the spatial mesh in the PDE solver tends to exacerbate the problem since then the sharp dynamics are more fully resolved.¹

Moments exactly on $\partial\mathcal{R}$ are uniquely generated by a linear combination of no more than $(N + 1)/2$ delta functions when N is odd, and no more than $N/2$ deltas when N is even [29]. As a consequence, the matrix of moments which make up the

¹On the other hand, reducing the time step Δt makes the optimization easier because the warm start is closer to the solution.

Hessian $H(\hat{\boldsymbol{\alpha}}(\mathbf{u}))$ when $\mathbf{u} \in \mathcal{R}$ is singular: Let

$$G(\mu) := \sum_{i=1}^d c_i \delta(\mu - \nu_i) \quad (3.11)$$

be a linear combination of deltas at ν_i with strengths $c_i > 0$ and $d \leq (N + 1)/2$. Notice that (3.11) is not of the form (2.8). The $(N + 1) \times (N + 1)$ matrix

$$H = \langle \mathbf{m} \mathbf{m}^T G \rangle = \sum_{i=1}^d c_i \mathbf{m}(\nu_i) \mathbf{m}(\nu_i)^T G(\nu_i) \quad (3.12)$$

is a sum of $d \leq (N + 1)/2$ rank-one matrices, so it is singular.

We have defined \mathcal{R} as open, and this $G \notin L^1(d\mu)$ is not an entropy-ansatz, but the ansätze of moments near $\partial\mathcal{R}$ approach the extreme anisotropy of the atomic densities on the boundary, and the condition number of the Hessian $\kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{u})))$ approaches infinity as \mathbf{u} approaches $\partial\mathcal{R}$. The condition number can be roughly understood using the following bounds:

$$\lambda_{\min}(H(\boldsymbol{\alpha})) \geq \lambda_{\min}(\langle \mathbf{m} \mathbf{m}^T \rangle) \min_{\mu \in [-1, 1]} (G_{\boldsymbol{\alpha}}) = \frac{2}{2N + 1} \min_{\mu \in [-1, 1]} (G_{\boldsymbol{\alpha}}), \quad (3.13)$$

$$\lambda_{\max}(H(\boldsymbol{\alpha})) \leq \lambda_{\max}(\langle \mathbf{m} \mathbf{m}^T \rangle) \max_{\mu \in [-1, 1]} (G_{\boldsymbol{\alpha}}) = 2 \max_{\mu \in [-1, 1]} (G_{\boldsymbol{\alpha}}), \quad (3.14)$$

where the constants in the far right terms come from the orthogonality of the Legendre polynomials: $\langle m_k m_l \rangle = 2\delta_{kl}/(2k + 1)$. Equations (3.13) and (3.14) yield an

upper bound on the condition number of the Hessian,

$$\kappa(H(\boldsymbol{\alpha})) = \frac{\lambda_{\max}(H(\boldsymbol{\alpha}))}{\lambda_{\min}(H(\boldsymbol{\alpha}))} \leq (2N + 1) \frac{\max_{\mu}(G_{\boldsymbol{\alpha}})}{\min_{\mu}(G_{\boldsymbol{\alpha}})}. \quad (3.15)$$

When the ratio $\max_{\mu} G_{\boldsymbol{\alpha}} / \min_{\mu} G_{\boldsymbol{\alpha}}$ is moderate, then the problem is well-conditioned. However, if $\min_{\mu} G_{\boldsymbol{\alpha}} \ll \max_{\mu} G_{\boldsymbol{\alpha}}$ —as in the case that $G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})}$ is approaching a linear combination of deltas—then the problem may be poorly conditioned. Indeed, $\max_{\mu} G_{\boldsymbol{\alpha}} / \min_{\mu} G_{\boldsymbol{\alpha}}$ was at least $O(10^{10})$ in the difficult problems we encountered in numerical experimentation.

We now present several examples to illustrate this second point.

3.2.1 The M_1 model

The M_1 model is the simplest example of an entropy-based moment system and uses only the first two Legendre polynomials: $\mathbf{m} = (m_0, m_1)^T = (1, \mu)^T$. The model was first introduced in [35] in the context of photon radiation and later analyzed in much greater detail in [36]. Unlike the case for most entropy-based models, the relationship between the moments and the multipliers in M_1 can be expressed without the use of integral formulas. This makes M_1 a useful tool for understanding the challenges of solving the dual problem (2.9).

First consider the first-order necessary condition for optimality of the M_1 dual problem. Let $\mathbf{u} = (u_0, u_1)^T$ and $\hat{\boldsymbol{\alpha}}(\mathbf{u}) = (\hat{\alpha}_0(\mathbf{u}), \hat{\alpha}_1(\mathbf{u}))^T$. Here

$$\mathcal{R} = \{(u_0, u_1) : |u_1| < u_0\}, \quad (3.16)$$

which simply follows from the fact that $\mu \in [-1, 1]$. By solving $\mathbf{g}(\hat{\boldsymbol{\alpha}}(\mathbf{u})) = 0$, one can show that the optimal multipliers satisfy (see [36])

$$u_0 = \frac{2 \exp(\hat{\alpha}_0(\mathbf{u}))}{\hat{\alpha}_1(\mathbf{u})} \sinh(\hat{\alpha}_1(\mathbf{u})), \quad (3.17)$$

$$\frac{u_1}{u_0} = \coth(\hat{\alpha}_1(\mathbf{u})) - \frac{1}{\hat{\alpha}_1(\mathbf{u})}. \quad (3.18)$$

A plot of the right-hand side of (3.18) is given in Figure 3.1(a). Appropriately in light of (3.16), the range of $\coth(\alpha_1) - 1/\alpha_1$ is $(-1, 1)$. From (3.18), one can show that, if u_0 is held constant, as $|u_1|/u_0 \rightarrow 1$

$$\hat{\alpha}_0(\mathbf{u}) \rightarrow -\infty \quad \text{and} \quad \frac{\hat{\alpha}_1(\mathbf{u})}{\hat{\alpha}_0(\mathbf{u})} \rightarrow \text{sign}(u_1). \quad (3.19)$$

The unbounded growth in the components of $\boldsymbol{\alpha}$ quickly causes numerical overflow and underflow when evaluating the exponential involving $\boldsymbol{\alpha}$ in the objective function and its derivatives.²

We can also see directly in the M_1 case that as \mathbf{u} approaches $\partial\mathcal{R}$ the Hessian of the dual problem at $\hat{\boldsymbol{\alpha}}(\mathbf{u})$ becomes singular: Let $v_2 := \langle \mu^2 G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})} \rangle$ be the second-order monomial moment. Then a standard calculation shows that the eigenvalues of H are

$$\lambda_{\pm} = \frac{1}{2}(u_0 + v_2) \pm |u_1| \sqrt{1 + \left(\frac{u_0 - v_2}{2u_1}\right)^2} \quad (3.20)$$

²In double-precision arithmetic, $\exp(-750) = 0$, and $\exp(710) = \text{Inf}$.

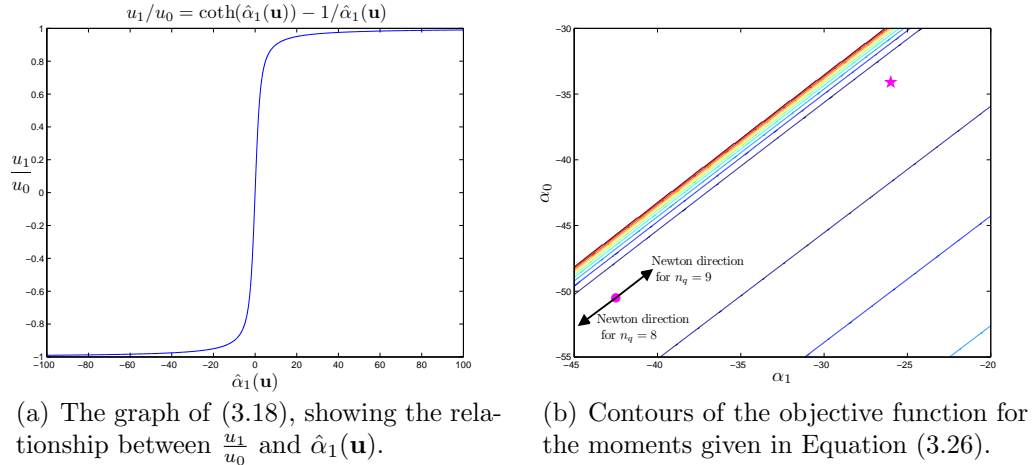


Figure 3.1: Illustrating difficulties in the optimization in M_1 .

and that the bound

$$|u_0 - v_2| = |\langle (1 - \mu^2)G_{\hat{\alpha}(\mathbf{u})} \rangle| \leq 2|\langle (1 \pm \mu)G_{\hat{\alpha}(\mathbf{u})} \rangle| = 2|u_0 \pm u_1| \quad (3.21)$$

holds, so that $v_2 \rightarrow u_0$ as $u_1 \rightarrow \pm u_0$. Thus the ratio λ_+/λ_- tends to ∞ as $u_1 \rightarrow \pm u_0$.

The numerical difficulties above are compounded by the fact that, in general, the integrals in f , \mathbf{g} , and H must be approximated by quadrature. Given a quadrature \mathcal{Q} , the approximation $f_{\mathcal{Q}}$ has the form

$$f_{\mathcal{Q}}(\boldsymbol{\alpha}) = \sum_{\mu_i \in \mathcal{Q}} w_i G_{\boldsymbol{\alpha}}(\mu_i) - \boldsymbol{\alpha}^T \mathbf{u}, \quad (3.22)$$

where $w_i > 0$ and $\mu_i \in \mathcal{Q}$ are the quadrature weights and nodes, respectively. For M_1 , the first-order necessary conditions for $f_{\mathcal{Q}}$ yield an analog to the second equation of (3.18):

$$\frac{u_1}{u_0} = \frac{\sum w_i \mu_i \exp(\hat{\alpha}_{\mathcal{Q},1} \mu_i)}{\sum w_i \exp(\hat{\alpha}_{\mathcal{Q},1} \mu_i)}, \quad (3.23)$$

where $\hat{\boldsymbol{\alpha}}_{\mathcal{Q}} = (\hat{\alpha}_{\mathcal{Q},0}, \hat{\alpha}_{\mathcal{Q},1})^T$ denotes the minimizer of $f_{\mathcal{Q}}$. Assuming the quadrature contains at least one node $\mu_i < 0$ and at least one node $\mu_i > 0$, consideration of the range of the right-hand side of (3.23) (with respect to $\hat{\alpha}_{\mathcal{Q},1}$) shows that (3.23) is solvable if and only if

$$\min_{\mu_i \in \mathcal{Q}} \{\mu_i\} < \frac{u_1}{u_0} < \max_{\mu_i \in \mathcal{Q}} \{\mu_i\}. \quad (3.24)$$

Thus $f_{\mathcal{Q}}$ does not have a minimizer for all $\mathbf{u} \in \mathcal{R}$.³

The Hessian of the approximate objective function, used in calculating the Newton direction, is given by the following sum of rank-one matrices:

$$H_{\mathcal{Q}}(\boldsymbol{\alpha}) = \sum_{\mu_i \in \mathcal{Q}} w_i G_{\boldsymbol{\alpha}}(\mu_i) \mathbf{m}(\mu_i) \mathbf{m}^T(\mu_i). \quad (3.25)$$

Suppose that \mathbf{u} is near $\partial\mathcal{R}$ and $\boldsymbol{\alpha} \rightarrow \hat{\boldsymbol{\alpha}}(\mathbf{u})$. Then as a consequence of (3.19), $G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})}$ can vary by arbitrarily many orders of magnitude over the interval of integration as it attempts to approximate something closer and closer to a delta function at $\mu = \pm 1$. In such cases, the limits of finite precision arithmetic mean that many of the terms in (3.25) underflow to exactly zero; this makes it harder for $H_{\mathcal{Q}}(\boldsymbol{\alpha})$ to build rank. Equation 3.25 suggests that adding quadrature points would make it easier for $H_{\mathcal{Q}}$ to build rank and thereby improve its condition number.

In numerical experimentation we encountered the moments

$$(u_0, u_1)^T = (1.19788813813286, -1.15179519716325)^T \times 10^{-5} \quad (3.26)$$

³Generalizations of (3.24) are discussed in Section 4.1 below.

which occurred as particles entered the vacuum surrounding an initial impulse. Here $|u_1|/u_0 \approx 0.962$. Figure 3.1(b) shows contours of the objective function for this problem and the effect of the quadrature approximation on the Newton direction. The minimizer of the true objective function, which is marked with a star in the upper right of the figure, is $\hat{\boldsymbol{\alpha}}(\mathbf{u}) \approx (-34.1, -26.0)^T$. A particular iterate $\boldsymbol{\alpha}_k$ is marked with a dot in the lower left corner of the figure along with the approximate Newton direction computed with an eight-point Gauss-Legendre quadrature. For this particular quadrature $\min_i\{\mu_i\} \approx -0.9603$, and thus according to (3.24), (3.23) is not solvable, i.e., $f_{\mathcal{Q}}$ does not have a minimizer. The figure shows that consequently the Newton direction points in the wrong direction. However, as seen in Figure 3.1(b), increasing $n_{\mathcal{Q}}$ by only one suffices to orient the approximation Newton direction correctly. For the nine-point Gauss-Legendre quadrature $\min_i\{\mu_i\} \approx -0.9682$ so that (3.23) is then solvable.

3.2.2 Challenges in higher-order models

Numerical difficulties also arise in higher-order models. Consider the following M_{15} example with the (normalized) moments and multipliers

$$\mathbf{u} = [1.0, \quad 0.83787256, \quad 0.57281969, \quad 0.29407137, \\ 0.07951925, \quad -0.03489476, \quad -0.06042812, \quad -0.03707798, \\ -0.00614557, \quad 0.00933745, \quad 0.00792086, \quad 0.00007545, \\ -0.00435021, \quad -0.00283280, \quad 0.00107465, \quad 0.00302283]^T, \quad (3.27a)$$

$$\boldsymbol{\alpha} = [-196.592892, \quad 230.276988, \quad 139.825688, \quad -201.869970, \\ -183.792885, \quad 351.679176, \quad -10.219892, \quad -278.491308, \\ 58.767548, \quad 304.081955, \quad -258.762434, \quad -112.854768, \\ 341.813503, \quad -269.754579, \quad 104.308274, \quad -17.093335]^T. \quad (3.27b)$$

We encountered these moments in the course of solving the two-beam instability problem discussed in Section 5.2.2, and the multipliers $\boldsymbol{\alpha}$ were an iterate of the optimization algorithm.

For \mathbf{u} in (3.27a), $\rho_{\partial\mathcal{R}}(\mathbf{u}) \approx 2.1 \times 10^{-10}$, and for the multipliers in (3.27b), $\rho_{\partial\mathcal{R}}(\hat{\mathbf{v}}(\boldsymbol{\alpha})) \approx 1.3 \times 10^{-10}$. As a reference, for the normalized isotropic moment $\mathbf{u}_{\text{iso}} = (1, 0, \dots, 0)^T$, $\rho_{\partial\mathcal{R}}(\mathbf{u}_{\text{iso}}) \approx 1.3 \times 10^{-5}$.

From Figure 3.2(a), it is clear that all the structure in the polynomial $\boldsymbol{\alpha}^T \mathbf{m}$ is on the left-hand side of the interval. However, because the pointwise values of $\boldsymbol{\alpha}^T \mathbf{m}$ are large and negative there, this structure is essentially destroyed when the exponential is applied (Figure 3.2(b)). Even though the function $G_{\boldsymbol{\alpha}}$ appears rela-

tively benign—nothing close to the delta functions which generate the moments on $\partial\mathcal{R}$ —the condition number of the numerical Hessian $H_{\mathcal{Q}}(\boldsymbol{\alpha})$ is quite large. Even using a very fine 800-point Gauss-Legendre quadrature on each of the subintervals $[-1, 0]$ and $[0, 1]$ to compute $H_{\mathcal{Q}}$, we find that $\lambda_{\min}(H_{\mathcal{Q}}) \approx 4.98 \times 10^{-12}$ and $\lambda_{\max}(H_{\mathcal{Q}}) \approx 2.21$, so that the condition number of $H_{\mathcal{Q}}$ is approximately 4.44×10^{11} .

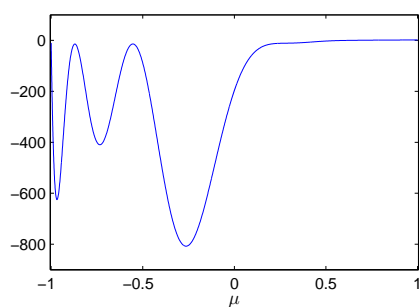
We denote the unit-length eigenvector associated with $\lambda_{\min}(H_{\mathcal{Q}})$ by $\mathbf{c}_{\mathcal{Q}}$, so $H_{\mathcal{Q}}\mathbf{c}_{\mathcal{Q}} = \lambda_{\min}(H_{\mathcal{Q}})\mathbf{c}_{\mathcal{Q}}$. Let

$$U(\mathbf{c}, \boldsymbol{\alpha}) := \mathbf{c}^T H(\boldsymbol{\alpha}) \mathbf{c} = \langle |\mathbf{c}^T \mathbf{m}|^2 G_{\boldsymbol{\alpha}} \rangle \quad (3.28)$$

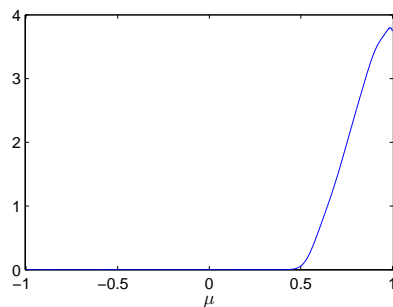
denote the quadratic form weighted by the Hessian for multipliers $\boldsymbol{\alpha}$ evaluated at $\mathbf{c} = \mathbf{c}_{\mathcal{Q}}$. (Note that $U(\mathbf{c}_{\mathcal{Q}}, \boldsymbol{\alpha}) = \lambda_{\min}(H_{\mathcal{Q}}(\boldsymbol{\alpha}))$ when quadrature \mathcal{Q} is used to evaluate the integral.) The results, given in Figure 3.2(c), show a combination of two effects. First, on the right-hand side of the interval, the polynomial $|\mathbf{c}_{\mathcal{Q}}^T \mathbf{m}|^2$ is very small, but due to the orthogonality of the Legendre polynomials, this cannot hold everywhere on the interval; indeed, on the left-hand side $|\mathbf{c}_{\mathcal{Q}}^T \mathbf{m}|^2$ becomes $O(1)$. However, on the left-hand side, $G_{\boldsymbol{\alpha}}$ is so small that any contribution to the integral in (3.28) is strongly damped.

Over the entire interval, the most significant contribution to the integral comes from the three peaks in $G_{\boldsymbol{\alpha}}$ on the left-hand hand side. (One of these is at the boundary $\mu = -1$.) It is interesting to note that the value of $|\mathbf{c}_{\mathcal{Q}}^T \mathbf{m}|^2$ dips significantly at these peaks so that the product $|\mathbf{c}_{\mathcal{Q}}^T \mathbf{m}|^2 G_{\boldsymbol{\alpha}}$ is $O(10^{-10})$. When \mathcal{Q} is coarsened, the number of quadrature points contained in the support of these peaks decreases,

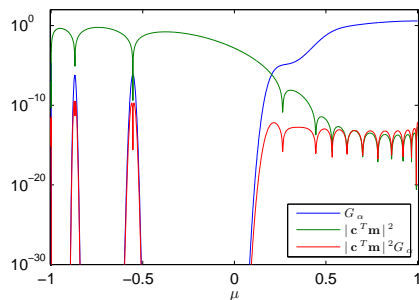
eventually causing the computed value of the integral (3.28) to decrease and the condition number of H_Q to increase. This effect is displayed in Figure 3.2(d), where we plot the condition number versus the number of quadrature points. In each case, the points are evenly divided into two Gauss-Legendre quadrature sets on the right and left sides. This result illustrates the need for a highly accurate quadrature set when \mathbf{u} is close to $\partial\mathcal{R}$.



(a) The polynomial $\alpha^T \mathbf{m}$.



(b) The ansatz G_α .



(c) The integrand of the quadratic form. (d) The condition number for $H_Q(\alpha)$.

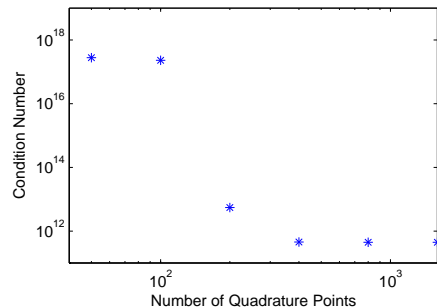


Figure 3.2: Examining the multipliers given in (3.27b).

A small perturbation⁴ of the multipliers from the example in (3.27b) gives:

$$\begin{aligned}
\boldsymbol{\alpha} = & [-1.9930449, \quad 2.357373, \quad 1.369548, \quad -2.042432, \\
& -1.787705, \quad 3.489758, \quad -0.116955, \quad -2.764542, \\
& 0.617561, \quad 2.948295, \quad -2.491223, \quad -1.139953, \\
& 3.304448, \quad -2.511948, \quad 0.897775, \quad -0.148746]^T \times 10^2.
\end{aligned} \tag{3.29}$$

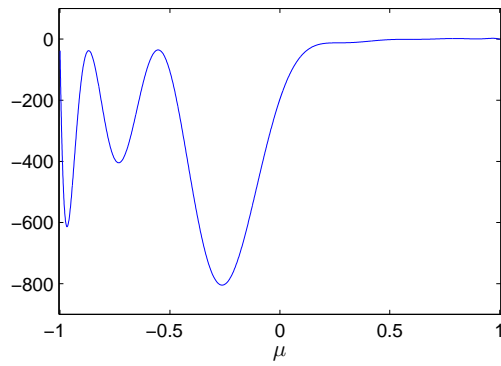
For these multipliers, $\rho_{\partial\mathcal{R}}(\hat{\mathbf{v}}(\boldsymbol{\alpha})) \approx 3.3 \times 10^{-15}$. By this measure, the $\hat{\mathbf{v}}(\boldsymbol{\alpha})$ is significantly closer to the $\partial\mathcal{R}$ than is the moment generated by the multiplier in (3.27b).

Figure 3.3 contains the same results as Figure 3.2, except that multipliers in (3.27b) are replaced by those in (3.29). It is interesting to note that the profile of $G_{\boldsymbol{\alpha}}$ in Figure 3.3(b) is apparently beginning to form deltas, although it is still fairly smooth.

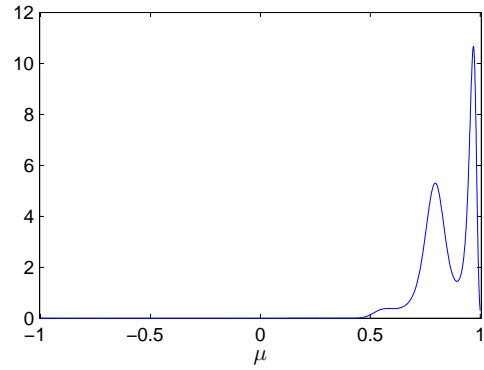
We again compute $H_{\mathcal{Q}}$ using a very fine 800-point Gauss-Legendre quadrature on each half interval, which is accurate enough to resolve the structure in $G_{\boldsymbol{\alpha}}$, and find that $\lambda_{\min}(H_{\mathcal{Q}}) \approx -1.30 \times 10^{-16}$ and $\lambda_{\max}(H_{\mathcal{Q}}) \approx 2.89$. (The fact that the computed value of $\lambda_{\min}(H_{\mathcal{Q}})$ is negative is a result of roundoff error from double precision arithmetic.) Thus the condition number of $H_{\mathcal{Q}}$ is at least $O(10^{16})$. It may in fact be larger, but no further conclusions can be drawn without increasing the working precision. The large condition number means that the relative error in the computed Newton step may be $O(1)$ or greater. Figure 3.3(d) shows that, unlike

⁴The relative difference between the multipliers in (3.27b) and (3.29) is roughly 5.3% when measured in the ℓ_{∞} norm. The multipliers in (3.29) were generated by fitting a polynomial to the one shown in Figure 3.2(a) but reducing the height of the maxima in $\mu \in [-1, 0]$.

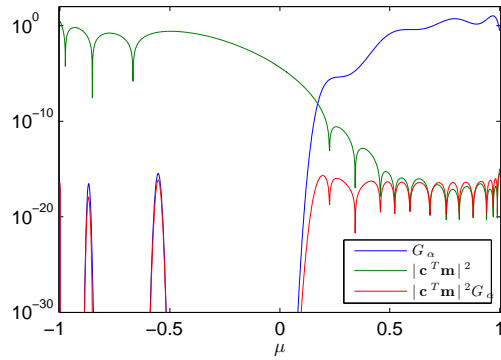
the multipliers in (3.27b), refining the quadrature over two orders of magnitude has little effect on the calculated condition number. Because G_α is still relatively smooth, we again conclude that the limitations in the optimization algorithm are not due to the quadrature in the case, but rather to the conditioning of the true Hessian H .



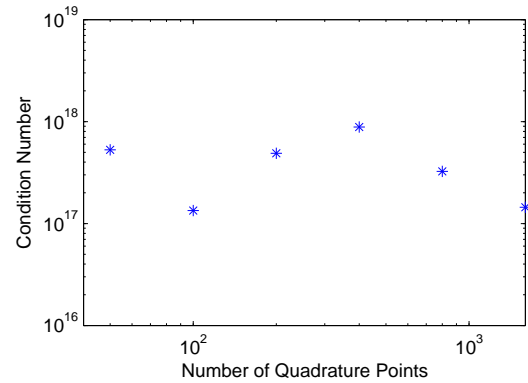
(a) The polynomial $\alpha^T \mathbf{m}$.



(b) The ansatz G_α .



(c) The integrand of the quadratic form.



(d) The condition number for $H_Q(\alpha)$.

Figure 3.3: Examining the multipliers given in (3.29).

3.2.3 Difficulties satisfying the γ tolerance

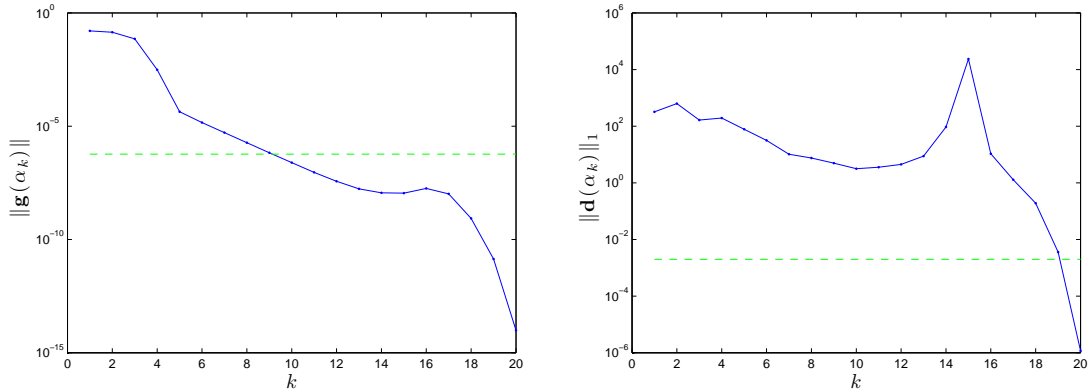
The behavior of the optimization algorithm for the moments and initial multipliers given by

$$\mathbf{u} = \begin{bmatrix} 1.0, & 0.94719513, & 0.84831958, & 0.71563896, \\ 0.56480997, & 0.41224747, & 0.27254228, & 0.15643088, \\ 0.06966859, & 0.01293677, & -0.01731948, & -0.02741629, \\ -0.02471554, & -0.01606500, & -0.00669578, & 0.00027475 \end{bmatrix}^T, \quad (3.30a)$$

$$\boldsymbol{\alpha}_0 = \begin{bmatrix} -16.650320, & -1.088112, & 6.202177, & 14.182838, \\ 13.819944, & 3.712406, & -8.787535, & -13.304092, \\ -7.246537, & 3.095360, & 8.571096, & 6.142392, \\ -0.581505, & -5.124053, & -2.724318, & 2.598485 \end{bmatrix}^T, \quad (3.30b)$$

illustrates another difficulty in solving the optimization. Here, $\rho_{\partial\mathcal{R}}(\mathbf{u}) = 7.6 \times 10^{-11}$, and in this section we use twenty-point Curtis-Clenshaw quadratures over both $\mu \in [-1, 0]$ and $\mu \in [0, 1]$. In this problem, as seen in Figure 3.4, the norm of the gradient satisfies the stopping criterion long before the Newton direction is small enough to satisfy (3.6b). It takes ten iterations to move the gradient below the tolerance and then ten additional iterations for the Newton direction to become small enough to satisfy the tolerance. In the figure, dashed green lines represent the stopping tolerance. (For the Newton direction, according to (3.6b), this is $\log(1 + \varepsilon_\gamma)/(5\zeta)$.)

Figure 3.4(a) shows typical behavior for this situation: the gradient has two



(a) The norm of gradient at each iteration. (b) The one-norm of the Newton direction at each iteration.

Figure 3.4: The stopping criterion quantities for the optimization algorithm for (3.30)

stages of quick decrease. The first moves it below the gradient tolerance, and the second happens after a few iterations of at best small improvements. This second stage then quickly moves the gradient down to $O(10^{-14})$ in a few iterations, and further improvement is difficult to achieve in double precision. In more pathological examples that we have observed, the optimization may take several more steps with $\|\mathbf{g}(\boldsymbol{\alpha}_k)\| = O(10^{-14})$ while it searches for a point where the Newton direction will satisfy the tolerance.

A large Newton step in these situations may be partially explained by the fact that large changes to the multipliers become increasingly necessary even to make the slight but significant changes to the ansatz needed near the solution. We must keep in mind that near the $\partial\mathcal{R}$, all of \mathbb{R}^N is squeezed into the bounded set $\mathcal{R}|_{u_0=1}$, and that the ansatz $G_{\boldsymbol{\alpha}} = \exp(\boldsymbol{\alpha}^T \mathbf{m})$ is insensitive to large changes to the polynomial $\boldsymbol{\alpha}^T \mathbf{m}$ when that polynomial is large and negative.

3.2.4 When the optimization breaks down

The problems in Sections 3.2.2 and 3.2.3 are challenging but ultimately solvable. The moments

$$\mathbf{u} = [1.0, \quad -0.98713923, \quad 0.96593546, \quad -0.93438151, \\ 0.89296807, \quad -0.84241937, \quad 0.78360175, \quad -0.71746459, \\ 0.64504944, \quad -0.56750364, \quad 0.48605581, \quad -0.40197628, \\ 0.31656192, \quad -0.23113041, \quad 0.14699352, \quad -0.06541041]^T, \quad (3.31)$$

however, give an example of where the optimization breaks down. Here, we start from the isotropic multipliers, which are generally the safest initial-condition choice, and we use twenty-point Curtis-Clenshaw quadratures over both $\mu \in [-1, 0]$ and $\mu \in [0, 1]$. The distance to the boundary is only $\rho_{\partial\mathcal{R}}(\mathbf{u}) = 1.7 \times 10^{-8}$, which is surprisingly higher than the previous examples.

In Figure 3.5, we see that over the first seven iterations the norm of the gradient is high and not decreasing, and the condition number of the Hessian is increasing quickly. At the seventh iteration, the Cholesky factorization fails. Not surprisingly, $\kappa(H(\boldsymbol{\alpha}_7))$ is $O(10^{17})$, which is above $1/\mathbf{eps}$.

Figure 3.6 illustrates what happens to the ansatz and the polynomial $\boldsymbol{\alpha}_k^T \mathbf{m}$ in the final three iterations. The single peak seen in the ansatz at $k = 5$ in Figure 3.6(a) forms in the first iteration and remains the only peak until the sixth iteration. Then a few small peaks form at $k = 6$, including one near $\mu = 0$ which blows up at $k = 7$ to $O(10^4)$ (off-scale in Figure 3.6(c)).

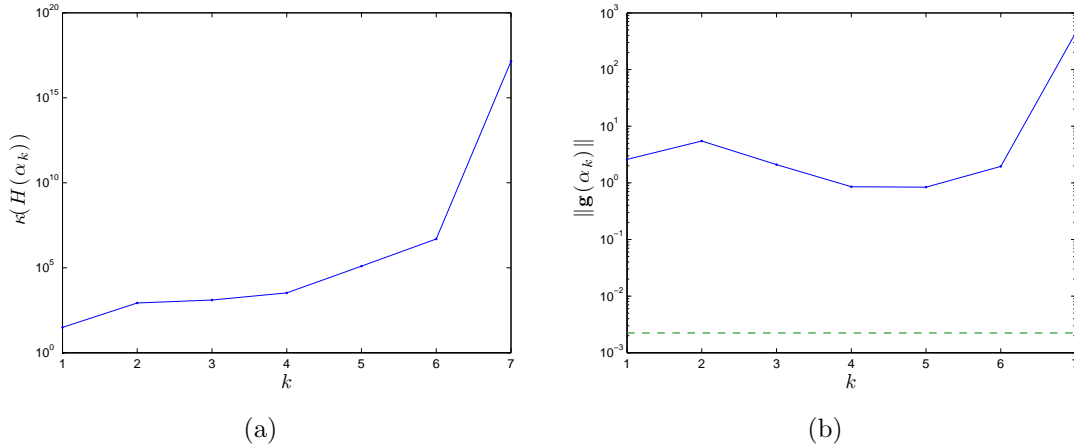


Figure 3.5: The norm of the gradient and the condition number of the Hessian when the optimization algorithm is applied to the moments in (3.31).

The polynomials maintain their shape from the fifth to the seventh iteration but grow by an order of magnitude in each step. Because $\log(\mathbf{eps}) \approx -36$, most of these terms make no contribution to quadrature approximations of $\mathbf{g}(\alpha_k)$ and $H(\alpha_k)$. Crucially, none of the peaks in the ansatz—including the one near $\mu = 0$ —are ever noticeable in the plots of the polynomials, Figures 3.6(d) to 3.6(f).

In this chapter we reviewed the troubles numerical optimization faces in solving the defining optimization problem for entropy-based moment closures. When the moments are near the realizable boundary, support of the ansatz shrinks and causes the Hessian of the dual objective function to be poorly conditioned in low- and high-order problems. We showed how these difficulties are manifested in the calculation of the stopping criterion and illustrated what happens to the ansatz when Newton’s method fails.

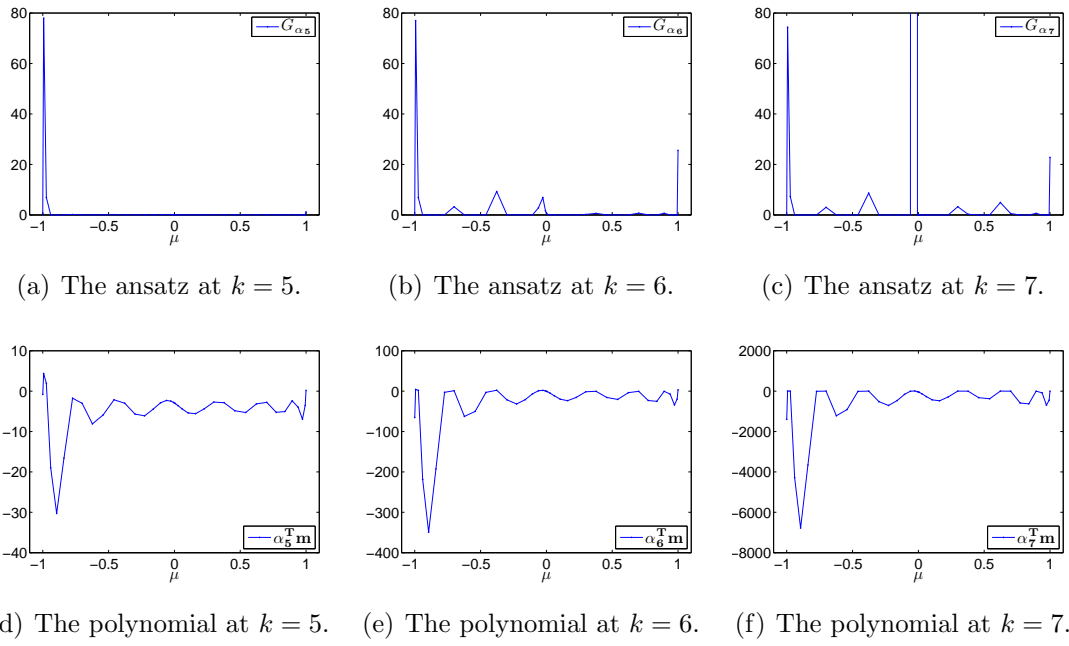


Figure 3.6: The ansätze and polynomials for the last three iterations when the optimization algorithm is applied to the moments in (3.31).

Chapter 4

Optimization Techniques

In this chapter, we discuss practical numerical techniques to address the difficulties exposed in Chapter 3. We explain the advantages of using fixed quadrature, modify and test an adaptive-basis algorithm, and introduce an isotropic regularization for modifying problems we cannot solve in finite precision. Most of this work will also be reported in [37].

4.1 Fixed quadrature

After making the observations in Chapter 3, we first experimented with adaptive quadrature to add quadrature points when the approximation of the objective function needed refinement [30]. But not only does an adaptive quadrature create complications for the optimizer (smartly predicted by [38]), it is important to realize that the full realizable set \mathcal{R} is only approximated using quadrature. Further, this approximation changes with the quadrature.

Given a quadrature \mathcal{Q} made up of nodes $\{\mu_i\}_{i=1}^{n_{\mathcal{Q}}} \subset [-1, 1]$ and weights $\{w_i\}_{i=1}^{n_{\mathcal{Q}}}$, we define the set of moments that are *realizable with respect to a quadrature \mathcal{Q}* as¹

$$\mathcal{R}_{\mathcal{Q}} := \{\mathbf{u} \mid \mathbf{u} = \sum w_i \mathbf{m}(\mu_i) g_i, g_i > 0\}. \quad (4.1)$$

¹The strict inequalities in the definition $\mathcal{R}_{\mathcal{Q}}$ are chosen so that Slater's condition is satisfied by the discretized primal. Slater's condition guarantees that the duality gap is zero [39].

It is easily shown to be a subset of \mathcal{R} , and the following result shows that only in special cases does $\mathcal{R}_{\mathcal{Q}}$ reach the boundary of \mathcal{R} .

Theorem 5. *For any quadrature \mathcal{Q} with positive weights w_i ,*

$$\mathcal{R}_{\mathcal{Q}}|_{u_0=1} = \text{int co}\{\mathbf{m}(\mu_i)\}_{\mu_i \in \mathcal{Q}}, \quad (4.2)$$

where co indicates the convex hull, and int indicates interior.

Proof. If $\mathbf{u} \in \mathcal{R}_{\mathcal{Q}}|_{u_0=1}$, then $\lambda_i := w_i g_i > 0$ and $\sum \lambda_i = 1$ (using $u_0 = 1$, $m_0 \equiv 1$, and assuming $w_i > 0$) show that \mathbf{u} is a convex combination of $\{\mathbf{m}(\mu_i)\}$ with positive coefficients λ . Therefore $\mathcal{R}_{\mathcal{Q}}|_{u_0=1} \subseteq \text{int co}\{\mathbf{m}(\mu_i)\}$.

For the other direction, assume $\mathbf{u} \in \text{int co}\{\mathbf{m}(\mu_i)\}$, that is $\mathbf{u} = \sum \lambda_i \mathbf{m}(\mu_i)$. Choosing $g_i := \lambda_i / w_i > 0$ shows that $\mathbf{u} \in \mathcal{R}_{\mathcal{Q}}|_{u_0=1}$, and so we have $\text{int co}\{\mathbf{m}(\mu_i)\} \subseteq \mathcal{R}_{\mathcal{Q}}|_{u_0=1}$. □

The rest of $\mathcal{R}_{\mathcal{Q}}$ (i.e. the unnormalized moments) is simply the cone generated by $\mathcal{R}_{\mathcal{Q}}|_{u_0=1}$:

$$\mathcal{R}_{\mathcal{Q}} = \{\mathbf{u} \mid \mathbf{u} = c\mathbf{v}, c > 0, \mathbf{v} \in \mathcal{R}_{\mathcal{Q}}|_{u_0=1}\}. \quad (4.3)$$

Figure 4.1 demonstrates the M_2 case for two low-order quadratures. $\mathcal{R}|_{u_0=1}$ is flat on the top, but curved at the bottom, as shown in Figure. If $\mu = \pm 1$ are nodes in \mathcal{Q} (as in Figure 4.1(b)), the entire flat portion at the top is in $\mathcal{R}_{\mathcal{Q}}|_{u_0=1}$. But on the bottom (where moments are generated by a single delta located in $(-1, 1)$), no moment on $\partial\mathcal{R}|_{u_0=1}$ is also on $\partial\mathcal{R}_{\mathcal{Q}}|_{u_0=1}$ unless the locations of the delta functions are nodes in \mathcal{Q} . In fact, every quadrature node adds a vertex of $\mathcal{R}_{\mathcal{Q}}$ which is on $\partial\mathcal{R}$.

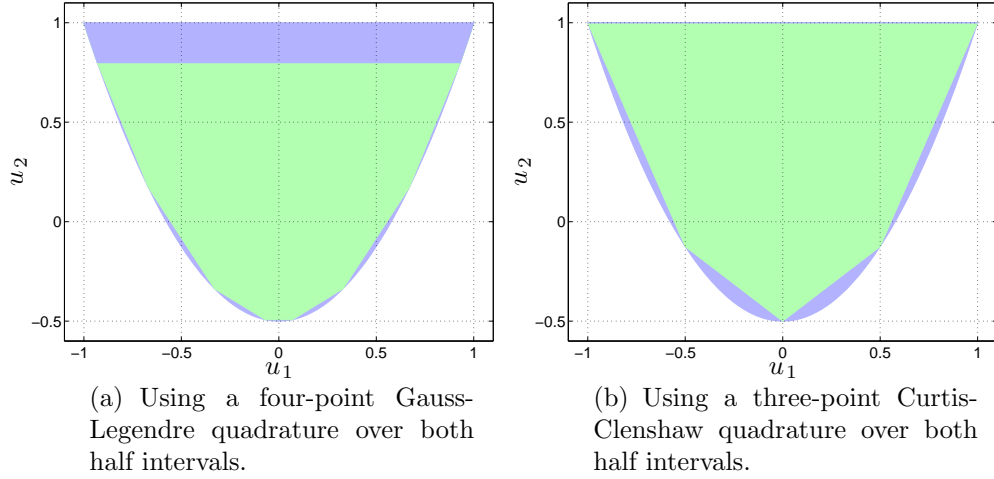


Figure 4.1: Illustrating $\mathcal{R}_{\mathcal{Q}}|_{u_0=1}$ for M_2 . The green indicates $\mathcal{R}_{\mathcal{Q}}|_{u_0=1}$, and the blue indicates $\mathcal{R}|_{u_0=1} \setminus \mathcal{R}_{\mathcal{Q}}|_{u_0=1}$.

We settled on two important features for the quadrature. First, because the angular integral takes different forms for positive and negative angles in the numerical scheme,² we apply separate quadratures for $\mu \in [-1, 0]$ and $\mu \in [0, 1]$. Second, we use Curtis-Clenshaw quadrature on each half interval because, as we can see from Figure 4.1(a), the fact that Gauss-Legendre quadrature does not include the endpoints $\mu = \pm 1$ cuts out a large region of $\mathcal{R}_{\mathcal{Q}}$. Furthermore, moments where $|u_1|$ approaches 1 are essential for solving problems where particles are concentrated near $\mu \pm 1$, as we have in the plane source problem below.

A fixed quadrature \mathcal{Q} also allows us to keep the numerical solution within the \mathcal{Q} -realizable set $\mathcal{R}_{\mathcal{Q}}$:

Theorem 6. *Let γ_{max} be the maximum value of (2.19) over all quadrature nodes, spatial cells and stages of the Runge-Kutta method. Suppose that $\mathbf{u}_j^n \in \mathcal{R}_{\mathcal{Q}}$ for $j \in \{-1, \dots, N_x + 2\}$. If \mathbf{u}^{n+1} is defined via the kinetic scheme described in Section*

²See, for example, (A.3)–(A.6) in the Appendix

2.4 and with time-step restriction

$$\gamma_{max} \frac{\Delta t}{\Delta x} \frac{\theta + 2}{2} + \sigma_t \Delta t < 1 \quad (4.4)$$

and if the moments in the ghost cells are in $\mathcal{R}_{\mathcal{Q}}$ at each stage of the Runge-Kutta scheme, then $\mathbf{u}_j^{n+1} \in \mathcal{R}_{\mathcal{Q}}$ for $j \in \{1, \dots, N_x\}$.

The proof of Theorem 6 is a trivial modification of the proof of Theorem 4 (see Appendix A.2).

4.2 Adaptive polynomial basis

We have seen that when $\hat{\mathbf{v}}(\boldsymbol{\alpha})$ is near $\partial\mathcal{R}$, poor conditioning of the numerical Hessian is unavoidable. Using a large number of quadrature points is sometimes one way to improve the condition number of the Hessian in many problems (see Figures 3.2(d) and 3.3(d)), but it often requires an unreasonable number of quadrature points. The work of [18] provides another approach: to switch to a polynomial basis for which the Hessian is well-conditioned.

We consider basis polynomials of a particular form:

Definition 2. A vector of polynomials $\mathbf{p} = \mathbf{p}(\mu) = (p_0(\mu), \dots, p_N(\mu))^T$ is said to be triangular if $\deg(p_i) = i$ for $i \in \{0, \dots, N\}$.

Let an invertible matrix T define a new basis $\mathbf{p} = T^{-1}\mathbf{m}$ and define $f_T :$

$\mathbb{R}^{N+1} \rightarrow \mathbb{R}$ by

$$f_T(\boldsymbol{\beta}) := f(T^{-T}\boldsymbol{\beta}) = \langle \exp(\boldsymbol{\beta}^T T^{-1}\mathbf{m}) \rangle - \boldsymbol{\beta}^T T^{-1}\mathbf{u}, \quad (4.5)$$

so that $f(\boldsymbol{\alpha}) \equiv f_T(T^T\boldsymbol{\alpha})$. If T is lower-triangular, then \mathbf{p} is a triangular basis because \mathbf{m} is.

The transformed objective function f_T is of course also strictly convex with gradient

$$\mathbf{g}_T(\boldsymbol{\beta}) = T^{-1} \langle \mathbf{m} \exp(\boldsymbol{\beta}^T T^{-1}\mathbf{m}) \rangle - T^{-1}\mathbf{u} = T^{-1}\mathbf{g}(T^{-T}\boldsymbol{\beta}) \quad (4.6)$$

and positive definite Hessian

$$H_T(\boldsymbol{\beta}) = T^{-1} \langle \mathbf{m}\mathbf{m}^T \exp(\boldsymbol{\beta}^T T^{-1}\mathbf{m}) \rangle T^{-T} = T^{-1}H(T^{-T}\boldsymbol{\beta})T^{-T}. \quad (4.7)$$

The sequence generated by Newton's method is invariant under affine transformations of the domain of the function being minimized, so in exact arithmetic a Newton iteration for f is equivalent to one for f_T in the following sense: given initial iterates, respectively $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$, related by $\boldsymbol{\beta}_k = T^T\boldsymbol{\alpha}_k$, the respective next iterates $\boldsymbol{\alpha}_{k+1}$ and $\boldsymbol{\beta}_{k+1}$ satisfy $\boldsymbol{\beta}_{k+1} = T^T\boldsymbol{\alpha}_{k+1}$. (This still holds true when line search is used, of the type used below.) As noted and exploited in [18] though, this is no longer true in finite precision, where appropriate sequential changes of basis can dramatically improve conditioning.

We switch to a new basis at every iteration where we calculate the Newton direction for the minimization of f , in such a way that the Hessian at the current

iterate expressed in the new basis is the identity. However we depart from [18] (for reasons discussed in Section 4.2.3 below) in that we compute the change of basis using the Cholesky decomposition rather than the Gram-Schmidt orthogonalization. In exact arithmetic and when applied to the same original triangular basis, both yield the same new basis except possibly for a reversal of some of the axes.³

The algorithm proceeds as follows. Let β_k indicate the multipliers at the start of iteration k , when the polynomial basis⁴ is $\mathbf{p}_{k-1} = T_{k-1}^{-1}\mathbf{m}$, so that the ansatz at this point is $\exp(\beta_k^T \mathbf{p}_{k-1})$. A Cholesky factorization of the Hessian $H_{T_{k-1}}(\beta_k)$ uncovers the new basis where the Hessian is identity, since

$$H_{T_{k-1}}(\beta_k) = \langle \mathbf{p}_{k-1} \mathbf{p}_{k-1}^T \exp(\beta_k^T \mathbf{p}_{k-1}) \rangle = L_k \underbrace{\langle \mathbf{p}_k \mathbf{p}_k^T \exp(\beta_k^T \mathbf{p}_{k-1}) \rangle}_{=I} L_k^T, \quad (4.8)$$

where L_k is the lower-triangular Cholesky factor of $H_{T_{k-1}}(\beta_k)$

$$H_{T_{k-1}}(\beta_k) = L_k L_k^T. \quad (4.9)$$

Hence the new basis is $\mathbf{p}_k := L_k^{-1} \mathbf{p}_{k-1}$. Note that $\mathbf{p}_k = L_k^{-1} T_{k-1}^{-1} \mathbf{m}$, so

$$T_k := T_{k-1} L_k. \quad (4.10)$$

Since T_{-1} and L_k are lower-triangular, T_k also is for all k .

We denote the multipliers at iterate k in the new \mathbf{p}_k basis by β'_k . The obser-

³This is due to the easily shown fact that the triangular basis \mathbf{p} which is orthonormal with respect to $G_\alpha d\mu$ is unique up to the signs of the component polynomials.

⁴The polynomial basis is initialized to $\mathbf{p}_{-1} = T_{-1}^{-1} \mathbf{m}$, where T_{-1} is lower-triangular.

vation that

$$\boldsymbol{\beta}_k^T \mathbf{p}_{k-1} = \boldsymbol{\beta}_k^T L_k \mathbf{p}_k, \quad (4.11)$$

shows that $\boldsymbol{\beta}'_k = L_k^T \boldsymbol{\beta}_k$. To check that $H_{T_k}(\boldsymbol{\beta}'_k)$ is identity, simply apply (4.7) and (4.9):

$$\begin{aligned} H_{T_k}(\boldsymbol{\beta}'_k) &= T_k^{-1} \langle \mathbf{m} \mathbf{m}^T \exp(\boldsymbol{\beta}_k^T T_k^{-1} \mathbf{m}) \rangle T_k^{-T} \\ &= L_k^{-1} T_{k-1}^{-1} \langle \mathbf{m} \mathbf{m}^T \exp(\boldsymbol{\beta}_k^T L_k^{-1} T_{k-1}^{-1} \mathbf{m}) \rangle T_{k-1}^{-T} L_k^{-T} \\ &= L_k^{-1} H_{T_{k-1}}(L_k^{-T} \boldsymbol{\beta}'_k) L_k^{-T} = I. \end{aligned} \quad (4.12)$$

The Newton direction \mathbf{d}_{T_k} for f_{T_k} from this iterate is simply the steepest descent direction

$$\mathbf{d}_{T_k}(\boldsymbol{\beta}'_k) = -\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k) = \mathbf{u}_k - \langle \mathbf{p}_k \exp(\boldsymbol{\beta}'_k^T \mathbf{p}_k) \rangle, \quad (4.13)$$

where $\mathbf{u}_k := L_k^{-1} \mathbf{u}_{k-1} = T_k^{-1} \mathbf{u}$ is the vector of moments in the \mathbf{p}_k basis. In implementation, the use of $-\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k)$ as the search direction is a robust choice: even though round-off errors cause the Hessian in the new basis not to be exactly identity, this search direction is at least a descent direction.

In fact, the gradient in the new basis is given by

$$\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k) = (p_{0,k} \langle \exp(\boldsymbol{\beta}'_k^T \mathbf{p}_k) \rangle, 0, \dots, 0)^T - \mathbf{u}_k. \quad (4.14)$$

Since T_k is lower triangular, the first component of \mathbf{p}_k is a constant, and so by the orthogonality of the polynomials \mathbf{p}_k with respect to $\exp(\boldsymbol{\beta}'_k^T \mathbf{p}_k) d\mu$, only the first

component of $\langle \mathbf{p}_k \exp(\boldsymbol{\beta}_k^T \mathbf{p}_k) \rangle$ (the integral term on the right in (4.13)) is nonzero.

The overall process is described in Algorithm 1.

Algorithm 1 The optimization algorithm using the adaptive-basis method.

Input: $\mathbf{u} \in \mathcal{R} \subset \mathbb{R}^{N+1}$, $\boldsymbol{\beta}_0 \in \mathbb{R}^{N+1}$, $T_{-1} \in \mathbb{R}^{(N+1) \times (N+1)}$, $P_{-1} \in \mathbb{R}^{(N+1) \times n_{\mathcal{Q}}}$
 { \mathbf{u} is assumed to be in the Legendre basis \mathbf{m} ; P_{-1} holds the evaluations of the initial basis $\mathbf{p}_{-1} = T_{-1}^{-1} \mathbf{m}$ at the quadrature nodes.}

Parameters: $\tau > 0$, $\varepsilon_\gamma > 0$

```

 $\mathbf{u}_{-1} \leftarrow T_{-1}^{-1} \mathbf{u}$ 
for  $k \in \{0, 1, 2, \dots\}$  do
   $H \leftarrow \langle \mathbf{p}_{k-1} \mathbf{p}_{k-1}^T \exp(\boldsymbol{\beta}_k^T \mathbf{p}_{k-1}) \rangle$ .
   $L \leftarrow \text{chol}(H)$ 
   $T_k \leftarrow T_{k-1} L$ 
   $P_k \leftarrow L^{-1} P_{k-1}$ 
   $\mathbf{u}_k \leftarrow L^{-1} \mathbf{u}_{k-1}$ 
   $\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k) \leftarrow (p_{0,\text{out}} \langle G \rangle, 0, \dots, 0)^T - \mathbf{u}_k$ 
   $\boldsymbol{\beta}'_k \leftarrow L^T \boldsymbol{\beta}_k$ 
   $\mathbf{d}_{T_k}(\boldsymbol{\beta}'_k) \leftarrow -\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k)$ 
  if  $\|\mathbf{g}(\boldsymbol{\alpha}_k)\| < \tau$  and  $\exp(5 \max_\mu |\mathbf{d}(\boldsymbol{\beta}'_k)^T \mathbf{p}_k|) < 1 + \varepsilon_\gamma$  then
     $\bar{\boldsymbol{\alpha}} \leftarrow T_k^{-T} \boldsymbol{\beta}'_k$  {The optimal multipliers in the Legendre basis}
    return  $\bar{\boldsymbol{\alpha}}, \boldsymbol{\beta}'_k, T_k$ 
  else
    Choose stepsize  $\xi_k$  using a line search
     $\boldsymbol{\beta}_{k+1} \leftarrow \boldsymbol{\beta}'_k + \xi_k \mathbf{d}_{T_k}(\boldsymbol{\beta}'_k)$ 
  end if
end for

```

Before moving on, we should make one note about how Algorithm 1 works in practice. When the stepsize ξ_k becomes practically zero in the working precision (possibly indicating a miscalculated search direction), the Hessian—for the same ansatz—is recalculated and refactorized at iteration $k + 1$. Thus the basis \mathbf{p}_{k+1} becomes a reorthogonalization of the basis \mathbf{p}_k and should result in a more accurate search direction.

4.2.1 Implementation details

In a computer implementation of Algorithm 1, basis polynomials \mathbf{p} are stored as a $(N + 1) \times n_{\mathcal{Q}}$ matrix, here denoted P , which holds their evaluation at the quadrature nodes, i.e. $P_{ij} := p_i(\mu_j)$, where p_i is the i -th component of \mathbf{p} .

Solving an optimization problem in a changing basis requires careful book-keeping. As written here, our algorithms update both the matrix T_k , defining the relationship between the variable basis and the Legendre basis, and the matrix P_k containing the evaluation of the basis polynomials \mathbf{p}_k at the quadrature points. The matrix P_k is used repeatedly in quadratures, and it is be updated incrementally ($P_k \leftarrow L_k^{-1} P_{k-1}$).⁵

There are two ways we can use T_k to convert the gradient in the T_k basis back to the Legendre basis for use in the stopping criterion. The first is simply to compute $T_k \mathbf{g}_{T_k}(\boldsymbol{\beta}'_k)$, using only $(N + 1)^2$ multiplications. A second way is to compute $\mathbf{g}(T_k^{-T} \boldsymbol{\beta}'_k)$; here we first convert the multipliers back to the Legendre basis and then compute the gradient by quadrature there. At a cost of $(N + 1)^2 + n_{\mathcal{Q}}(N + 1)$ multiplications, this computation is more expensive, but it is a way to test exactly how the multipliers will be used in the flux. This gives us confidence that the conversion back to the Legendre basis via T_k —a matrix which becomes ill-conditioned in hard problems—does not introduce new errors.

The estimated upper bound on γ in the stopping criterion, which boils down to estimated the maximum of the polynomial $\boldsymbol{\beta}_k^T \mathbf{p}_k = \boldsymbol{\alpha}_k^T \mathbf{m}$, can be computed in

⁵In fact, strictly speaking it is not necessary to iteratively update T_k (which costs $(N + 1)^3$ multiplications per iteration). At the final iteration, the multipliers in the Legendre basis (see Remark 1 below) can be recovered from P_k and M .

either basis. The Newton direction $\mathbf{d}_{T_k}(\boldsymbol{\beta}'_k)$ can be converted back to the Legendre basis using T_k to use (3.6b) directly, or we can recompute ζ in (3.7) for the basis polynomials \mathbf{p}_k on the quadrature nodes by finding $\max_{ij} |p_i(\mu_j)|$ (this maximum only needs to be computed over the quadrature nodes; see Theorem 4) and use the one-norm of $\mathbf{d}_{T_k}(\boldsymbol{\beta}'_k)$. We chose the former approach because we found it to be slightly cheaper and less conservative while still reliably maintaining realizability in the PDE solver.

Remark 1. *The edge values for the flux calculations (see Section 2.4, particularly (2.17), and Appendix A) around spatial cell j are calculated using the ansätze at cells $j - 2, j - 1, j, j + 1, j + 2$. Therefore, each ansatz needs to be communicated to neighboring cells. Since in a parallel implementation data communication is a bottleneck, we want to use the most compact representation of the ansatz we can. The multipliers in a known, spatially consistent basis give the most compact representation because this requires $N + 1$ numbers to be communicated. The Legendre basis, the basis used by the PDE, is the natural choice for this basis. If the multipliers $\boldsymbol{\beta}_k$ are not converted back to a consistent basis, then in order to be interpreted they must be passed along either with the $(N + 1) \times (N + 1)$ matrix defining the basis or with the $(N + 1) \times n_{\mathcal{Q}}$ matrix storing the evaluation of the basis polynomials \mathbf{p}_k at the quadrature nodes. Alternatively, we could communicate the ansatz itself evaluated at the quadrature points, but the quadratures we typically use have $n_{\mathcal{Q}} \approx 2N$ nodes (and the quadrature must have at least $N + 1$ nodes to form a nonsingular estimate of the Hessian), so this also requires more communication.*

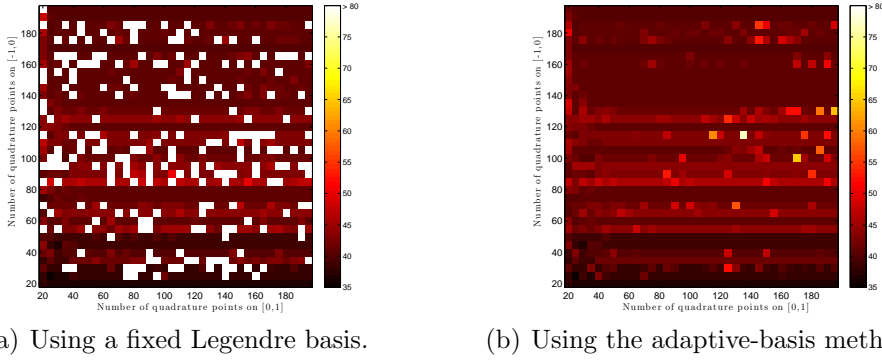


Figure 4.2: Comparing the fixed-basis method to the adaptive-basis method on the moments given in (3.27a). (Even at the bright pixel in the adaptive-basis figure, the adaptive-basis method converged in seventy-seven iterations.)

The initial bases specified by T_{-1} and P_{-1} are the Legendre basis at $t = 0$. For $t > 0$, the initial basis is the final basis from the previous time step in the same spatial cell.

4.2.2 Static results

Revisiting the example moments

Consider again the M_{15} moments from (3.27a). In Figure 4.2, we compare what happens when you try the fixed Legendre basis versus the adaptive-basis method of Algorithm 1. For each pixel in Figure 4.2, we fix a number of Curtis-Clenshaw quadrature points for $[-1, 0]$ and a number of quadrature points for $[0, 1]$. When the optimizer converges ($\tau = 10^{-8}$ and $\varepsilon_\gamma = 0.01$, with isotropic multipliers as initial conditions), it takes roughly thirty-five to eighty iterations, and we indicate the number of iterations by the shadings of the pixels in the figure. When the algorithm does not converge in fewer than 200 iterations, the pixel is white.

The figure gives striking evidence that the convergence of the fixed-basis algorithm is highly unpredictable. However, when using the adaptive-basis method, the

algorithm converges for all quadratures presented.

Approaching the boundary

The adaptive-basis method also allows us to solve optimization problems closer to the realizable boundary $\partial\mathcal{R}$, where moments are uniquely described by densities made of linear combinations of no more than $(N + 1)/2$ delta functions [29]: For $2m \leq N$, moments given by

$$\mathbf{u} = \left\langle \mathbf{m} \sum_{i=1}^m c_i \delta(\mu - \nu_i) \right\rangle = \sum_{i=1}^m c_i \mathbf{m}(\mu_i), \quad (4.15)$$

(where $c_i \geq 0$ and $\nu_i \in [-1, 1]$) lie on $\partial\mathcal{R}$ [29]. We can then define a sequence $\{\mathbf{u}_\ell^{(k)}\}$ which approaches $\partial\mathcal{R}$ as $\ell \rightarrow \infty$ by considering a convex combination between a $\mathbf{u}^{(k)}$ of the form (4.15) and the moments of the isotropic distribution. We use the isotropic moments $Q\mathbf{u}$ so that the local particle concentration remains constant for all ℓ . The moments in the sequence are therefore given by

$$\mathbf{u}_\ell^{(k)} = (1 - 2^{-\ell})\mathbf{u}^{(k)} + 2^{-\ell}Q\mathbf{u}, \quad \text{for } \ell \in \{0, 1, 2, \dots\}. \quad (4.16)$$

We performed experiments with $N = 12$ and $m = 6$ delta functions and Curtis-Clenshaw quadrature. Table 4.1 gives the strengths c_i and locations ν_i of the delta functions generating example moments $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(6)}$, each of which lies on $\partial\mathcal{R}$. Then for $k = 1, \dots, 6$ and for several values of n_Q (the number of quadrature points over each half interval $[-1, 0]$ and $[0, 1]$), Table 4.2 shows the value of ℓ^* for which $\bar{\alpha}(\mathbf{u}_\ell^{(k)})$ can be computed satisfying the tolerances $\tau = 10^{-8}$ and $\varepsilon_\gamma = 0.01$ for all $\ell \leq \ell^*$.

Table 4.1: Boundary moments used for tests in Table 4.2 below. The deltas in $\mathbf{u}^{(5)}$ are at the 4-th, 7-th, 10-th, 13-th, 14-th, and 15-th nodes of the twenty-point Curtis-Clenshaw quadrature over the interval $[-1, 0]$. In $\mathbf{u}^{(6)}$, $\nu_1, \nu_3, \dots, \nu_6$ are at the 4-th, 9-th, 12-th, 15-th, and 17-th nodes of the same quadrature, while ν_2 is the 54-th node of the 153-point Curtis-Clenshaw quadrature over the interval $[-1, 0]$.

i		1	2	3	4	5	6
$\mathbf{u}^{(1)}$	ν_i	0.2	0.3	0.4	0.5	0.6	0.7
	c_i	0.167	0.167	0.167	0.167	0.167	0.167
$\mathbf{u}^{(2)}$	ν_i	0.2	0.3	0.4	0.5	0.6	0.7
	c_i	0.0833	0.0833	0.0833	0.333	0.0833	0.333
$\mathbf{u}^{(3)}$	ν_i	0.2	0.4	0.6	0.88	0.89	0.9
	c_i	0.0833	0.0833	0.0833	0.333	0.0833	0.333
$\mathbf{u}^{(4)}$	ν_i	-0.8	-0.5	-0.1	0.59999	0.6	0.8
	c_i	0.167	0.167	0.167	0.167	0.167	0.167
$\mathbf{u}^{(5)}$	ν_i	-0.94	-0.773	-0.541	-0.299	-0.227	-0.161
	c_i	0.417	0.0417	0.0417	0.0417	0.417	0.0417
$\mathbf{u}^{(6)}$	ν_i	-0.94	-0.729	-0.623	-0.377	-0.161	-0.0603
	c_i	0.167	0.167	0.167	0.167	0.167	0.167

The values of n_Q are chosen so that the quadratures with higher n_Q include the nodes from those with fewer quadrature points. (The nodes of the Curtis-Clenshaw quadrature of order $2n_Q - 1$ include all nodes of the quadrature of order n_Q .)

The moments $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(4)}$ are chosen to show the effects of changing the strengths c_i and the distance between locations of the deltas. $\mathbf{u}^{(5)}$ and $\mathbf{u}^{(6)}$ are chosen to illustrate the case when quadrature nodes are co-located with the deltas generating the moments on the boundary.

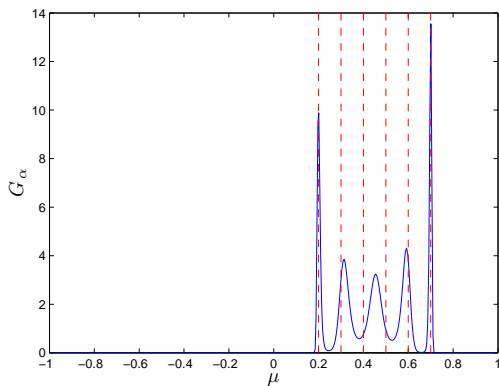
Table 4.2 shows that the adaptive-basis method never performs worse than the fixed-basis method, and usually outperforms it in all but the hardest problems. The hardest problems here are when two of the deltas generating the boundary moments

Table 4.2: For each $\mathbf{u}^{(k)}$ in Table 4.1 above, this table shows the largest value of ℓ for which the optimization algorithm (with either adaptive basis (A) or fixed basis (F)) can find approximately optimal multipliers $\bar{\alpha}(\mathbf{u}_\ell^{(k)})$ with tolerances $\tau = 10^{-8}$ and $\varepsilon_\gamma = 0.01$. $N = 12$.

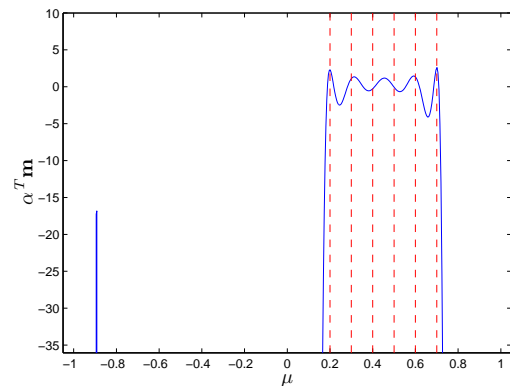
n_Q	$\mathbf{u}^{(1)}$		$\mathbf{u}^{(2)}$		$\mathbf{u}^{(3)}$		$\mathbf{u}^{(4)}$		$\mathbf{u}^{(5)}$		$\mathbf{u}^{(6)}$	
	A	F	A	F	A	F	A	F	A	F	A	F
20	16	13	13	13	5	5	3	3	41	18	29	22
39	25	21	25	16	17	17	6	6	38	18	28	17
77	27	14	35	19	22	18	8	8	33	18	29	15
153	26	16	20	16	25	20	10	8	21	18	28	19
305	25	15	32	20	26	14	12	12	27	15	28	16

are close together.

Figure 4.3 shows the final ansatz for $\bar{\alpha}(\mathbf{u}_{25}^{(1)})$, the moment closest to $\partial\mathcal{R}$ that we could solve with the highest-resolution quadrature. Interestingly, the peaks of the ansatz do not always line up with the deltas generating $\mathbf{u}^{(1)}$, and the polynomial includes a spurious peak near $\mu = -0.9$.



(a) The ansatz at the solution.



(b) The polynomial at the solution. The lower-limit of the vertical axis corresponds to $\log(\text{eps})$.

Figure 4.3: The solution for $\mathbf{u}_{27}^{(1)}$. The vertical red dashed lines indicate the locations of the deltas generating $\mathbf{u}^{(1)}$.

4.2.3 Alternative orthogonalizations

There are of course many different orthogonal polynomial bases with respect to $G_{\alpha}d\mu$. As mentioned above, in [18], the author also introduces an adaptive-basis method to transform the Hessian to identity but with slightly different motivations from ours: There the integration domain is unbounded, and the author noticed that as a consequence the objective function was much more sensitive to changes in multipliers associated with higher-order polynomials. The author orthogonalized the polynomial basis before every Newton iteration using the modified Gram-Schmidt process initialized with a random basis \mathbf{p} where each polynomial in the basis was of the same order: $\deg p_i = N$ for $i \in \{0, \dots, N\}$.

If the Gram-Schmidt process is instead initialized to a triangular basis such as the monomials or the Legendre polynomials, then the bases defined by the Gram-Schmidt process and the Cholesky factorization are exactly the same (up to round-off errors). Since our integration domain is bounded, a triangular basis does not cause poor conditioning of the Hessian, and we prefer to use triangular bases, as they lead to simpler matrix operations. It is particularly helpful to include a zeroth order polynomial in the basis so that scaling the problem is easy.

The Cholesky method is also less computationally expensive. Considering only the highest-order terms, the Cholesky method uses $n_{\mathcal{Q}}N^2/2$ multiplications to form the Hessian, $N^3/6$ multiplications to factorize the Hessian, and $n_{\mathcal{Q}}N^2/2$ multiplications to update the array P_k storing the evaluation of the basis polynomials at the quadrature nodes. Typically, we choose $n_{\mathcal{Q}} \approx 2N$.

Table 4.3: Same as Table 4.2 above, but here we compare the use of Cholesky factorization (C) to that of modified Gram-Schmidt (GS) in for the adaptive-basis method.

$n_{\mathcal{Q}}$	$\mathbf{u}^{(1)}$		$\mathbf{u}^{(2)}$		$\mathbf{u}^{(3)}$		$\mathbf{u}^{(4)}$		$\mathbf{u}^{(5)}$		$\mathbf{u}^{(6)}$	
	C	GS	C	GS	C	GS	C	GS	C	GS	C	GS
20	16	17	13	13	5	5	3	3	46	49	29	29
39	25	25	25	25	17	17	6	6	50	52	32	32
77	28	30	36	35	22	22	8	8	33	33	29	34
153	26	26	33	33	19	25	10	10	23	22	28	28
305	24	25	32	32	26	27	12	12	27	27	53	28

On the other hand, the modified Gram-Schmidt method does not need to form the Hessian but instead requires $n_{\mathcal{Q}}N^2$ multiplications to evaluate the necessary inner products and also performs $n_{\mathcal{Q}}N^2/2$ multiplications to update the array storing the evaluation of the basis polynomials at the quadrature nodes.

The extra cost of the modified Gram-Schmidt is well-known to lead to better stability properties, so we repeated the boundary-moment tests of Table 4.2 in Table 4.3. The results show that the Cholesky method generally performs just as well as the modified Gram-Schmidt method. This may be explained by the automatic reorthogonalization feature of Algorithm 1 mentioned above, and we have observed this working as expected in practice.

The singular value decomposition of the Hessian $H = U\Lambda U^T$, where U is unitary and Λ is diagonal, can also define an orthogonal basis $\mathbf{m} = U\Lambda^{1/2}\mathbf{p}$. We experimented with this way of defining the new basis, but found it to perform comparably to the change-of-basis defined by the Cholesky factorization. Since the singular value decomposition gives a non-triangular basis, is more expensive, and is

also harder to parallelize, we again conclude that the Cholesky factorization makes more sense for our problem.

4.3 Isotropic regularization

Even with Theorem 6 and the adaptive-basis algorithm increasing the number of numerically tractable problems, for any quadrature \mathcal{Q} some realizable moments are so close to the boundary of $\mathcal{R}_{\mathcal{Q}}$ that the support of the ansatz $G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})}$ is too small for the numerically approximated Hessian near $\hat{\boldsymbol{\alpha}}(\mathbf{u})$ to well-conditioned enough to use Newton’s method.

Equation (3.15) tells us that when $\min G_{\boldsymbol{\alpha}}$ is small, even small increases to it can dramatically improve $\kappa(H(\boldsymbol{\alpha}))$. Since

$$\kappa(\langle \mathbf{m}\mathbf{m}^T(G_{\boldsymbol{\alpha}} + \varepsilon) \rangle) \leq \frac{\max G_{\boldsymbol{\alpha}} + \varepsilon}{\varepsilon}, \quad (4.17)$$

choosing ε even as small as, say, $O(10^{-8})$ can yield a numerically usable Hessian.⁶ Unfortunately $G_{\boldsymbol{\alpha}+\varepsilon}$ is not an entropy ansatz, but of course the isotropic distribution is. Since we think of ε as much smaller than $\max G_{\boldsymbol{\alpha}}$, $G_{\boldsymbol{\alpha}} + \varepsilon$ can be thought of as a slightly more isotropic version of $G_{\boldsymbol{\alpha}}$, and so we hypothesize that moving closer to the isotropic distribution should improve $\kappa(H(\boldsymbol{\alpha}))$.

Thus to find an easier optimization problem, we further hypothesize that mov-

⁶Of course we also need $\max G_{\boldsymbol{\alpha}}$ to be not too large, and typically it is not. For example, in Figure 3.2(b), $\max G_{\boldsymbol{\alpha}}$ is $O(1)$, and even for difficult problems faced below in Section 5.2.1, $\max G_{\boldsymbol{\alpha}}$ is no worse than $O(10^3)$. To approximate a delta of strength c near a quadrature node μ_i , the ansatz need only be as large as c/w_i . Thus, the poor conditioning of the Hessian we have observed is caused by $\min G_{\boldsymbol{\alpha}}$ going to zero and not $\max G_{\boldsymbol{\alpha}}$ exploding.

ing the moments \mathbf{u} closer to the isotropic moments $Q\mathbf{u}$ ⁷ should decrease $\kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{u})))$. The isotropic problem is the only trivially solvable problem⁸ and $\kappa(H(\hat{\boldsymbol{\alpha}}(Q\mathbf{u})))$ is only $O(N)$. Therefore, keeping in mind the convexity of \mathcal{R} and \mathcal{R}_Q , we define the *isotropically regularized* moment

$$\mathbf{v}(r) := (1 - r)\mathbf{u} + rQ\mathbf{u}, \quad (4.18)$$

where we think of $r \in (0, 1)$ as a regularization factor that should be chosen as small as possible (see below). The form of $\mathbf{v}(r)$ is also chosen so that the local particle concentration is unchanged: $\langle G_{\hat{\boldsymbol{\alpha}}(\mathbf{u})} \rangle \equiv \langle G_{\hat{\boldsymbol{\alpha}}(\mathbf{v}(r))} \rangle$

Analysis of $\hat{\kappa}(r) := \kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{v}(r))))$ is challenging due to the complexity of the function κ , but there is strong numerical evidence that $\hat{\kappa}$ is a decreasing function of r if \mathbf{u} is near $\partial\mathcal{R}_Q$. Figure 4.5(a) shows that, for M_1 , $\kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{u})))$ is reasonable in the interior of \mathcal{R}_Q but rapidly blows up only when \mathbf{u} is very close to $\partial\mathcal{R}_Q$.

To illustrate this in M_2 , we first define three moments of the form (4.18):

$$\mathbf{v}^{(1)}(r) := \begin{pmatrix} 1 \\ 1 - r \\ 1 - r \end{pmatrix}, \quad \mathbf{v}^{(2)}(r) := \begin{pmatrix} 1 \\ \frac{1-r}{\sqrt{3}} \\ 0 \end{pmatrix}, \quad \mathbf{v}^{(3)}(r) := \begin{pmatrix} 1 \\ 0 \\ 1 - r \end{pmatrix}. \quad (4.19)$$

The first set of moments approach the those of a single delta at $\mu = 1$; the second approach those of a pair of equal deltas at $\mu = \pm 1$; and the third approach those of a single delta at $\mu = 1/\sqrt{3}$. The curves parametrized by r in \mathcal{R} are in Figure 4.4.

⁷Recall that Q is defined in (2.6).

⁸ $\hat{\boldsymbol{\alpha}}(Q\mathbf{u}) = (\log(u_0/2), 0, \dots, 0)^T$

Figure 4.5(b) shows that the M_2 case behaves similar to the M_1 case in 4.5(b): $\kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{u})))$ increases rapidly as $\mathbf{u} \rightarrow \partial\mathcal{R}_Q$. Thus even a small value of r —which gives a small move towards the interior of \mathcal{R}_Q —can drastically improve the conditioning of the optimization problem.

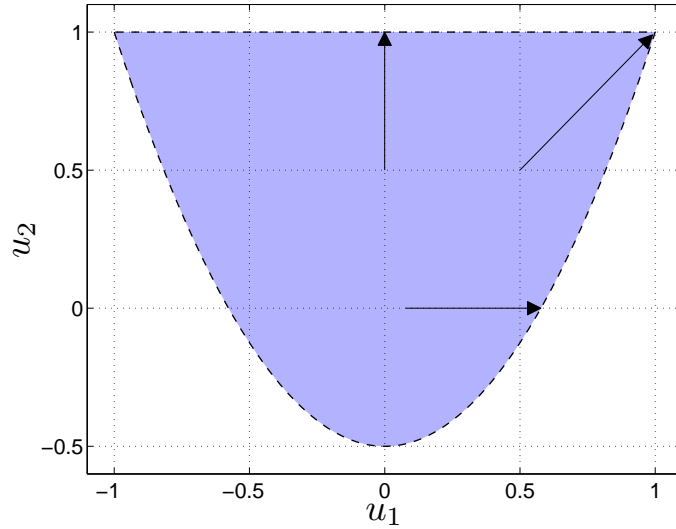
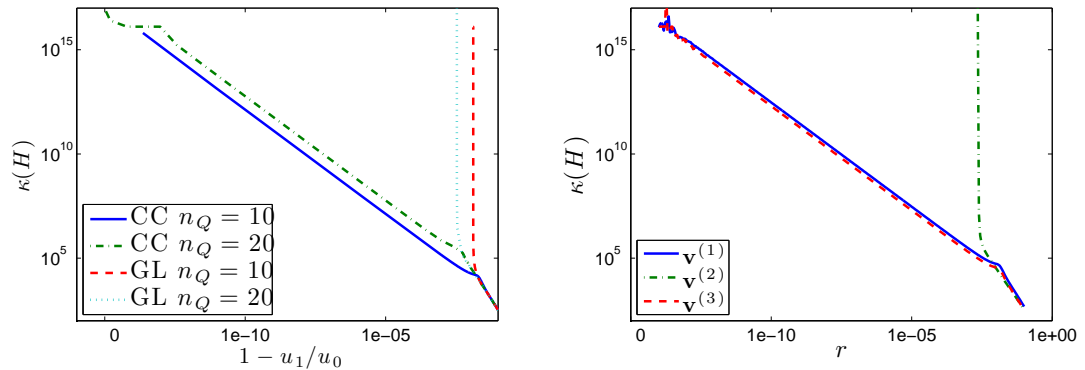


Figure 4.4: The set of normalized realizable moments $\mathcal{R}|_{u_0=1}$ in M_2 and the paths we take to the boundary in the moments defined in (4.19).



(a) The condition number of $H(\hat{\boldsymbol{\alpha}}(\mathbf{u}))$ in M_1 as $u_1/u_0 \rightarrow 1$ for four different quadratures. (b) The condition number of $H(\hat{\boldsymbol{\alpha}}(\mathbf{v}^{(k)}(r)))$ for the M_2 examples in (4.19).

Figure 4.5: The condition number of the Hessian near the boundary in M_1 and M_2 .

In order to maintain realizability in the PDE algorithm (i.e, to apply Theorem

6), we must also replace each subvector of $\mathbf{u}^{(m)}$, $m \in \{0, 1\}$, by its regularized version at each Runge-Kutta stage (cf. (A.9)) of the kinetic scheme for (2.12).

This regularization technique can be compared to the use of ‘numerical’ scattering for moment problems in [40, 41]. The regularization acts as barrier that prevents the entropy-ansatz from getting too close to the delta-type distributions which characterize the boundary $\partial\mathcal{R}$. In this sense, it can be viewed as numerical dissipation in μ -space.

Remark 2. *In some cases, the regularization can be used to solve the dual problem for a non-regularized moment \mathbf{u} when direct application of Newton’s method fails. This is done as follows. Define a decreasing sequence $r_\ell \searrow 0$ and successively solve the dual problem to find $\hat{\boldsymbol{\alpha}}(\mathbf{v}(r_\ell))$, using $\hat{\boldsymbol{\alpha}}(\mathbf{v}(r_{\ell-1}))$ as an initial condition. This defines a new path in $\boldsymbol{\alpha}$ -space to the minimizer $\hat{\boldsymbol{\alpha}}(\mathbf{u})$ of the original problem. In practice, we were indeed able to solve some M_{15} problems for which Newton’s method either failed or needed thousands of quadrature points. However, the fraction of moments for which this method worked when the Newton method did not was relatively small, and hence we did not include it in our implementation.*

4.4 The final optimization algorithm

A scheme to select the regularization parameter r is the final component for our algorithm. First we pick an increasing sequence $\{r_\ell\}_{\ell=0}^{\ell_f}$, starting with $r_0 = 0$, so that the optimizer first attempts to solve the original problem. Then if that problem cannot be solved quickly enough or if the Cholesky factorization fails, we increase

the regularization parameter r by incrementing ℓ .

We say that the optimization is not converging quickly enough if after an initial k_0 iterations the norm of the gradient in the Legendre basis is not decreasing by at least a factor of $\lambda \in [0, 1)$ at each iteration, namely if

$$\|\mathbf{g}(\boldsymbol{\alpha}_k)\| > \lambda \|\mathbf{g}(\boldsymbol{\alpha}_{k-1})\| \tag{4.20}$$

for some $k > k_0$. When $\lambda = 0$, this simply means we increment ℓ anytime the algorithm has not converged after k_0 iterations.

We let $r_{\max} := \max_{\ell} \{r_{\ell}\} = r_{\ell_f}$ and assume that r_{\max} is chosen large enough that $\bar{\boldsymbol{\alpha}}(\mathbf{v}(r_{\max}))$ can be computed for any \mathbf{u} . Therefore, if $r = r_{\max}$, then we do not check condition (4.20) and simply continue until a solution is found. In fact, to ensure a completely robust scheme the value of r_{\max} should be set to 1, but from experiments such as those in Table 4.2, we can see that 10^{-4} is a reasonable choice. The resulting algorithm is given in Algorithm 2.

In this chapter we discussed some methods for numerically handling the troubles discussed in Chapter 3. We presented advantages of using a fixed quadrature, modified and tested an adaptive-basis method for our context, and introduced an isotropic regularization method to find nearby tractable problems when the original problem is too poorly conditioned to solve in a given finite precision.

Algorithm 2 The optimization algorithm using the adaptive-basis method and a regularization scheme.

Input: $\mathbf{u} \in \mathcal{R} \subset \mathbb{R}^{N+1}$, $\boldsymbol{\beta}_0 \in \mathbb{R}^{N+1}$, $T_{-1} \in \mathbb{R}^{(N+1) \times (N+1)}$, $P_{-1} \in \mathbb{R}^{(N+1) \times n_Q}$
 { \mathbf{u} is assumed to be in the Legendre basis \mathbf{m} ; P_{-1} holds the evaluations of the initial basis $\mathbf{p}_{-1} = T_{-1}^{-1} \mathbf{m}$ at the quadrature nodes.}

Parameters: $\tau > 0$, $\varepsilon_\gamma > 0$, $\{r_\ell\} \subset [0, 1]$

for $\ell \in \{0, 1, \dots, \ell_f\}$ **do**
 $\mathbf{v}_{-1} \leftarrow T_{-1}^{-1}((1 - r_\ell)\mathbf{u} + r_\ell Q\mathbf{u})$
 for $k \in \{0, 1, 2, \dots\}$ **do**
 $H \leftarrow \langle \mathbf{p}_{k-1} \mathbf{p}_{k-1}^T \exp(\boldsymbol{\beta}_k^T \mathbf{p}_{k-1}) \rangle$.
 $L \leftarrow \text{chol}(H)$
 if chol fails or $(k > k_0$ and $\|\mathbf{g}(\boldsymbol{\alpha}_k)\|/\|\mathbf{g}(\boldsymbol{\alpha}_{k-1})\| > \lambda$ and $r < r_{\max})$ **then**
 {Exit the inner for loop, so that we subsequently increase r }
 break for
 else
 $T_k \leftarrow T_{k-1} L$
 $P_k \leftarrow L^{-1} P_{k-1}$
 $\mathbf{v}_k \leftarrow L^{-1} \mathbf{v}_{k-1}$
 $\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k) \leftarrow (p_{0,\text{out}} \langle G \rangle, 0, \dots, 0)^T - \mathbf{v}_k$
 $\boldsymbol{\beta}'_k \leftarrow L^T \boldsymbol{\beta}_k$
 $\mathbf{d}_{T_k}(\boldsymbol{\beta}'_k) \leftarrow -\mathbf{g}_{T_k}(\boldsymbol{\beta}'_k)$
 if $\|\mathbf{g}(\boldsymbol{\alpha}_k)\| < \tau$ and $\exp(5 \max_\mu |\mathbf{d}(\boldsymbol{\beta}'_k)^T \mathbf{p}_k|) < 1 + \varepsilon_\gamma$ **then**
 {Convert the optimal multipliers to the Legendre basis}
 $\bar{\boldsymbol{\alpha}} \leftarrow T_k^{-T} \boldsymbol{\beta}'_k$
 return $\bar{\boldsymbol{\alpha}}, \boldsymbol{\beta}'_k, T_k$
 else
 Choose stepsize ξ_k using a line search
 $\boldsymbol{\beta}_{k+1} \leftarrow \boldsymbol{\beta}'_k + \xi_k \mathbf{d}_{T_k}(\boldsymbol{\beta}'_k)$
 end if
 end if
 end for
end for

Chapter 5

Numerical Results

Unless otherwise noted, we use the following parameter values:

$\tau = 10^{-8},$	upper bound for $\ \mathbf{g}(\boldsymbol{\alpha}_k)\ $ in the stopping criterion,
$\varepsilon_\gamma = 0.01,$	upper bound on $\gamma_{\max} - 1$ to maintain realizability,
$k_0 = 40,$	number of iterations before testing the decrease in the norm of the gradient
$\lambda = 0,$	expected factor of decrease in the norm of the gradient
$\{r_\ell\} = \{0, 10^{-8}, 10^{-6}\}$	the sequence of values of the regularization parameter considered
$\theta = 2.0,$	slope limiting parameter, see (A.6).

We always choose

$$\Delta t = \frac{0.95}{1 + \varepsilon_\gamma} \frac{2}{\theta + 2} \Delta x, \quad (5.1)$$

in accordance with Theorem 6.

5.1 Manufactured solutions

The accuracy of a numerical solution to the moment system is difficult to compute because, generally speaking, the true solution is unknown. Even high-resolution solutions we are able to compute contain errors due to regularization, the effects of which we have not yet been able to quantify precisely. The method of manufactured solutions provides an approach where the true solution is known from the start. For this method, we numerically solve

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \mathbf{s}, \quad (5.2)$$

where the source $\mathbf{s} = \mathbf{s}(x, t)$ is calculated from the given ‘manufactured solution’ $\mathbf{v}(x, t)$ so that $\mathbf{s} = \partial_t \mathbf{v} + \partial_x \mathbf{f}(\mathbf{v})$. Applying a numerical method to (5.2) gives an approximation of \mathbf{v} .

We must choose \mathbf{v} carefully. It is crucial that the moments of the numerical solution remain in \mathcal{R} , but Theorem 6 does not apply when a source term is present. Thus errors in the numerical solution (particularly when \mathbf{s} calls for a removal of particles from a cell) can cause the numerical solution to leave \mathcal{R} . We attempt to avoid these difficulties with two choices: First, we choose ansätze which move most particles in one direction (here we choose the positive direction). Second, we choose \mathbf{v} such that v_0 only increases along this direction.

We choose \mathbf{v} by specifying multipliers $\boldsymbol{\alpha}(x, t)$,

$$\mathbf{v}(x, t) = \langle \mathbf{m} \exp(\boldsymbol{\alpha}(x, t)^T \mathbf{m}) \rangle, \quad (5.3)$$

so that its flux is

$$\mathbf{f}(\mathbf{v}(x, t)) = \langle \mu \mathbf{m} \exp(\boldsymbol{\alpha}(x, t)^T \mathbf{m}) \rangle . \quad (5.4)$$

We set

$$\alpha_2(x, t) \equiv \alpha_3(x, t) \equiv \dots \equiv \alpha_N(x, t) \equiv 0 \quad (5.5)$$

so that integrals of the form $\langle \mu^k \exp(\boldsymbol{\alpha}^T \mathbf{m}) \rangle$ can be explicitly computed.

One choice we tested for the remaining two multipliers is

$$\alpha_0(x, t) = \log \left(\frac{(0.1 + tx)\alpha_1(x, t)}{2 \sinh(\alpha_1(x, t))} \right) , \quad (5.6a)$$

$$\alpha_1(x, t) = 0.1 + Ktx . \quad (5.6b)$$

The zeroth multiplier α_0 is chosen so that $v_0(x, t) = 0.1 + tx$. The additional 0.1 term is added to avoid the case $\alpha_1 = 0$, where the exact evaluation of integrals of the form $\langle \mu^k \exp(\alpha_1 \mu) \rangle$ is numerically unstable. For this choice of multipliers, $(x_L, x_R) := (0, 1)$ and $t \in [0, 1]$. The boundary conditions are chosen to match the manufactured solution \mathbf{v} on the boundaries. At the right and left boundaries, we use the exact multipliers for \mathbf{v} in the ghost cells.

Another choice we experimented with is

$$\alpha_0(x, t) = \delta + tK(\cos(\pi x) + 1) + 0.1 , \quad (5.7a)$$

$$\alpha_1(x, t) = tK(\cos(\pi x) + 1) + 0.1 , \quad (5.7b)$$

where

$$\delta = \log \left(\frac{\exp(2K + 0.1)(2K + 0.1)}{2 \sinh(2K + 0.1)} \right) \quad (5.8)$$

is chosen so that the largest value of v_0 (which occurs at $(x, t) = (0, 1)$) is one. Here we let $(x_L, x_R) := (-1, 1)$ and use periodic boundary conditions. These multipliers specify a manufactured solution whose ansatz is always nondecreasing in space, time, and angle for $(x, t) \in (-1, 0) \times (0, 1)$. Consequently, we only simulate the system for $x \in (-1, 0)$ and $t \in [0, 1]$. At the boundary edges $x = -1$ and $x = 0$, we specify the flux exactly.

For either choice of multipliers, \mathbf{v} approaches $\partial\mathcal{R}$ as the parameter K increases. The ansatz $G_{\hat{\alpha}(\mathbf{v})}$ looks more and more like a single delta function at $\mu = 1$ as K increases. Below, we use $K = 53$ for the multipliers given in (5.6) and $K = 25$ for those in (5.7). These are the largest values of K we could use without numerical errors in the source term causing the solution to leave the realizable set.

To find the cell averages of the source term $\mathbf{s}_j(t) = \int_{I_j} \mathbf{s}(x, t) dx$ used in the finite volume scheme, we must integrate the time derivative term of the manufactured solution, $\partial_t \mathbf{v}$, over each cell I_j . For the \mathbf{v} we choose, this integral does not have an analytical form, so we approximate it with quadrature. As with the angular quadratures, we use twenty Curtis-Cleenshaw points each over the half intervals of I_j , $(x_{j-1/2}, x_j)$ and $(x_j, x_{j+1/2})$.

The particle density $u_0(x, t)$ for 400-cell numerical simulations of these two systems are shown in Figure 5.1. Right away, we see one problem with using multipliers (5.7): here, the solution varies by many orders of magnitude over the spatial

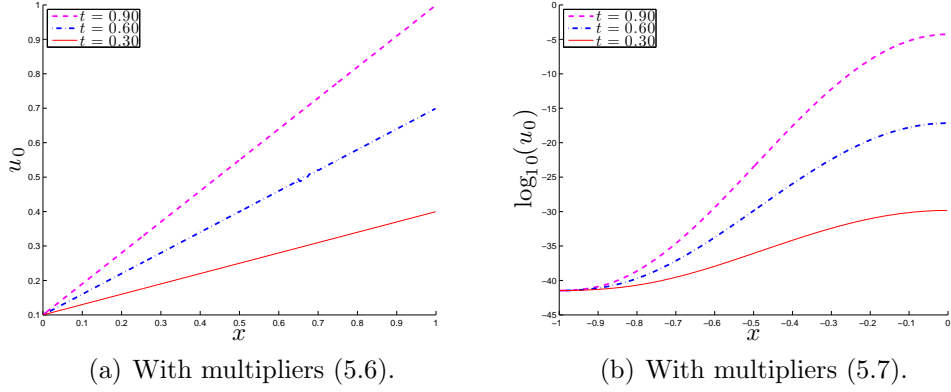


Figure 5.1: The local particle concentrations for the two manufactured solutions for 400-cell simulations.

domain. This is a consequence of the fact that the multipliers, which grow linearly in time, are exponentiated. We still report on results using this solution because it has an advantage which we discuss below.

Our knowledge of the true solution is $\mathbf{v}(x, t)$ allows us to calculate errors for every numerical simulation. We first interpolate the cell averages using second-order affine reconstructions in each spatial cell:

$$\mathbf{u}_{\Delta x}(x) = \mathbf{u}_j + (x - x_j) \frac{\mathbf{u}_{j+1} - \mathbf{u}_{j-1}}{2\Delta x}, \quad x \in I_j, \quad j \in \{1, \dots, N_x\}, \quad (5.9)$$

where the moments are all taken at the final time $t_f = 1$. Then the L^1 and L^∞ errors are given by

$$\mathbf{e}_{\Delta x}^1 := \int_{-1}^1 |\mathbf{v}(x, t_f) - \mathbf{u}_{\Delta x}(x)| dx, \text{ and} \quad (5.10a)$$

$$\mathbf{e}_{\Delta x}^\infty := \max_{x \in [-1, 1]} |\mathbf{v}(x, t_f) - \mathbf{u}_{\Delta x}(x)|, \quad (5.10b)$$

respectively. To approximate the integral in $\mathbf{e}_{\Delta x}^1$, we split each cell I_j into 100 equally sized subintervals, and then apply a twenty-point Gaussian quadrature on each

subinterval. $\mathbf{e}_{\Delta x}^\infty$ is then approximated as the maximum value over those quadrature points. Below, we only report errors in the particle density u_0 , i.e. the zeroth component of $\mathbf{e}_{\Delta x}^1$ and $\mathbf{e}_{\Delta x}^\infty$.

In this section, we use the M_5 model and choose $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}, 10^{-4}\}$.

5.1.1 Comparing adaptive-basis and fixed-basis methods

The results of Section 4.2.2 suggest that we should have to regularize less often with the adaptive-basis method. Since regularization is an artificial change made for numerical convenience, we then expect the resulting solution to be more accurate.

In this section, we include two regularization schemes. The first, where $k_0 = 5$ and $\lambda = 1/3$, is a more aggressive scheme chosen to keep the number of iterations down. The second, where $k_0 = 40$ and $\lambda = 0$, is chosen to favor accuracy over extra computational effort.

First, we examine the difference in computational effort for the two methods in Tables 5.1 to 5.4. The CPU time is generally larger for the adaptive basis. It can be as much as 33% higher, with that percentage decreasing as the number of cells increases. We attribute this difference largely to the more computationally expensive stopping criterion and extra matrix computations (such as updating the basis polynomials) in the adaptive-basis method. Table 5.4 shows one exceptional case where the adaptive-basis method needs fewer iterations and consequently uses less computational effort.

Tables 5.5 to 5.8 display the three statistics measuring how much regulariza-

Table 5.1: Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 5$ and $\lambda = 1/3$.

N_x	CPU time (s)		mean iterations	
	A	F	A	F
400	8.31e+02	6.19e+02	1.62	1.57
800	2.81e+03	2.12e+03	1.52	1.53
1200	6.22e+03	4.67e+03	1.54	1.55
1600	1.03e+04	7.93e+03	1.57	1.56

Table 5.2: Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$.

N_x	CPU time (s)		mean iterations	
	A	F	A	F
400	8.88e+02	6.56e+02	1.70	1.62
800	3.12e+03	2.34e+03	1.51	1.53
1200	6.41e+03	4.80e+03	1.54	1.54
1600	1.07e+04	8.26e+03	1.57	1.56

Table 5.3: Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 5$ and $\lambda = 1/3$.

N_x	CPU time (s)		mean iterations	
	A	F	A	F
400	3.06e+02	2.29e+02	1.52	1.58
800	1.12e+03	8.59e+02	1.39	1.43
1200	2.34e+03	1.85e+03	1.28	1.30
1600	3.97e+03	3.18e+03	1.19	1.22

Table 5.4: Statistics on the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$.

N_x	CPU time (s)		mean iterations	
	A	F	A	F
400	3.22e+02	3.18e+02	1.52	1.77
800	1.14e+03	1.34e+03	1.38	1.57
1200	2.71e+03	2.29e+03	1.28	1.42
1600	4.01e+03	4.14e+03	1.20	1.32

tion was performed in each simulation. The first column, ‘% regularized’ gives the percentage of nontrivial problems (those where the number of optimization iterations was at least one) for which the regularization parameter r was not zero. The second column is the mean value of r over the whole space-time mesh. As opposed to the percentage, this number takes into account the value of r chosen from $\{r_\ell\}$.

Finally, the statistic labeled $\sum_{j,n} r_{j,n} u_{j,0}^n$ is a rough estimate of the error introduced by regularization to the moment system. In one cell, the L^1 -norm of the error introduced by regularization is bounded by

$$\|\mathbf{u} - \mathbf{v}(r)\|_1 = r\|Q\mathbf{u} - \mathbf{u}\|_1 = r\|[0, -u_1, \dots, -u_N]\|_1 \leq rNu_0, \quad (5.11)$$

where, since $\|m_i\|_\infty = 1$, we have used $|u_i| \leq u_0$ for $i \in \{1, \dots, N\}$.

Thus in $\sum_{j,n} r_{j,n} u_{j,0}^n$, $j \in \{1, \dots, N_x\}$ corresponds to the spatial index and $n \in \{0, \dots, N_t - 1\}$ ¹ corresponds to the time index. Then $r_{j,n}$ indicates the value of r used in by the optimization algorithm in cell j at time-step n , and $u_{j,0}^n$ indicates

¹The optimization problem is not solved at the final time step $n = N_t$. We include both Runge-Kutta stages, but omit this from the notation for clarity of exposition.

Table 5.5: Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 5$ and $\lambda = 1/3$.

N_x	% regularized		mean r		$\sum_{j,n} r_{j,n} u_{j,0}^n$	
	A	F	A	F	A	F
400	2.68e-02	1.46e-01	1.79e-08	1.02e-08	2.99e-02	1.89e-02
800	7.43e-03	4.75e-02	2.99e-09	1.15e-09	3.02e-02	7.21e-03
1200	3.78e-03	7.51e-02	9.17e-10	8.38e-10	1.45e-02	1.45e-02
1600	2.23e-03	7.33e-02	9.65e-10	5.16e-10	2.78e-02	1.72e-02

Table 5.6: Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$.

N_x	% regularized		mean r		$\sum_{j,n} r_{j,n} u_{j,0}^n$	
	A	F	A	F	A	F
400	3.44e-01	3.54e-01	3.70e-11	1.64e-09	6.68e-05	2.15e-03
800	1.50e-03	1.02e-01	1.47e-13	6.85e-11	1.21e-06	6.52e-04
1200	1.46e-03	1.06e-01	1.32e-13	5.05e-11	2.40e-06	1.10e-03
1600	1.36e-03	1.31e-01	2.06e-13	3.76e-11	7.12e-06	1.44e-03

the particle density u_0 at the same space-time grid point. Therefore this sum over the space-time grid gives us an estimate of the error introduced by regularization, and as an improvement over the mean statistic, it is weighted by the magnitude of the moments where the regularization is applied.

The results in the tables show that usually the adaptive-basis method indeed regularizes far less often. However, the adaptive-basis method does tend to select higher values of r on the problems it does regularize, particularly when the more aggressive regularization scheme is used (Tables 5.5 and 5.7).

Finally, Tables 5.9 and 5.10 compute L^1 and L^∞ errors for the manufactured

Table 5.7: Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 5$ and $\lambda = 1/3$.

N_x	% regularized		mean r		$\sum_{j,n} r_{j,n} u_{j,0}^n$	
	A	F	A	F	A	F
400	0	0.226	0	7.79e-09	0	8.84e-06
800	0.028	0.225	8.88e-09	2.80e-09	8.99e-06	6.75e-05
1200	0.025	0.164	6.47e-09	1.85e-09	7.11e-03	2.78e-04
1600	0.021	0.156	5.75e-09	1.31e-09	2.57e-03	1.78e-04

Table 5.8: Statistics on regularization in the manufactured solution problem for adaptive-basis (A) and fixed-basis (F) optimization methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$.

N_x	% regularized		mean r		$\sum_{j,n} r_{j,n} u_{j,0}^n$	
	A	F	A	F	A	F
400	0	0.454	0	5.94e-11	0	2.47e-06
800	5.82e-04	0.368	2.20e-14	3.86e-11	3.01e-11	4.95e-06
1200	6.54e-04	0.292	2.28e-14	2.76e-11	1.67e-08	9.12e-06
1600	1.13e-04	0.258	3.67e-15	1.91e-11	1.51e-11	1.86e-05

Table 5.9: L^1 and L^∞ errors for adaptive-basis (A) and fixed-basis (F) methods for multipliers (5.6) with $k_0 = 40$ and $\lambda = 0$.

N_x	L^1		L^∞	
	A	F	A	F
400	2.3943e-04	3.3546e-05	1.9773e-03	8.3711e-04
800	3.1521e-06	1.1869e-05	4.5740e-04	6.6927e-04
1200	2.0612e-06	1.0938e-05	3.4598e-04	3.8160e-04
1600	2.1411e-06	6.2723e-06	2.6841e-04	2.9112e-04

solutions according to (5.10). We only include the less aggressive regularization scheme here because the errors with the more aggressive regularization scheme are very similar. Indeed, for the first multipliers, the adaptive-basis method is nearly one digit more accurate than the fixed-basis method here. However, the errors are almost exactly the same when the second multipliers are used, which also corresponds to the case when the difference in regularization was greatest. This may be evidence that the errors introduced by regularization are small compared to those due to space-time discretization.

Unfortunately, careful inspection of Table 5.9 shows that the solutions are not showing second-order convergence. At the present time, we are not sure what the reason for this is. The solution using the second multipliers does show second-order convergence, and this is the advantage mentioned earlier.

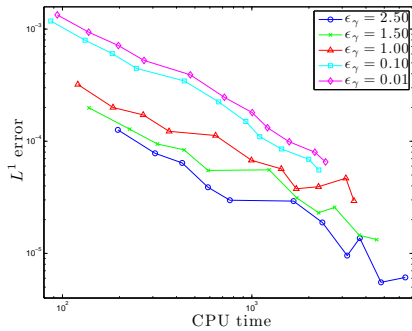
Table 5.10: L^1 and L^∞ errors for adaptive-basis (A) and fixed-basis (F) methods for multipliers (5.7) with $k_0 = 40$ and $\lambda = 0$.

N_x	L^1		L^∞	
	A	F	A	F
400	3.9073e-04	3.9073e-04	1.6779e-02	1.6779e-02
800	9.8737e-05	9.8875e-05	8.0776e-03	8.1133e-03
1200	4.4805e-05	4.4789e-05	5.3932e-03	5.3869e-03
1600	2.5755e-05	2.5776e-05	4.0963e-03	4.1080e-03

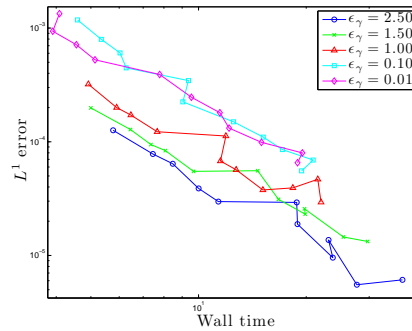
5.1.2 Choosing ε_γ according to the accuracy and computational effort trade-off

The parameter ε_γ does not have an immediately obvious effect on the accuracy and computation time of a numerical solution. Smaller values of ε_γ force a more accurate solution to the optimization problem, but require extra iterations to find that solution. On the other hand, larger values of ε_γ reduce the allowable time step Δt (see (5.1) and recall (2.20)), which leads more time steps (which means the optimization problem must be solved more times) to reach a given final time, but also improves the accuracy of the final solution.

We use the multipliers in (5.6) and set the tolerance on the norm of the gradient τ to 10^{-4} , a higher value so that (3.6b) is effectively the only stopping criterion. Then we ran several experiments changing ε_γ and N_x , the number of spatial cells. For each value of ε_γ , we then construct a curve parametrized by the the number of cells N_x . The vertical coordinate of the curve corresponds to the L^1 error between the numerical solution and the true solution, and the horizon-



(a) Total (serial) CPU time.



(b) Estimated wall time in fully parallel implementation. See (5.12).

Figure 5.2: L^1 errors and two computation time estimates (in seconds) for several values of ε_γ . Here $k_0 = 5$ and $\lambda = 1/3$.

tal coordinate of the curve corresponds to the CPU time needed to reach the final time $t_f = 1$. The results are in Figure 5.2. For both figures we choose $N_x \in \{200, 250, 300, 350, 400, 500, 600, 700, 800, 900, 1000\}$, and the computation time estimates do not include time spent calculating the source term for the manufactured solution.

Figure 5.2(a) uses the total CPU time of our current serial implementation. The results indicate that when high accuracy is desired or when lots of computation time is available, the largest choice $\varepsilon_\gamma = 2.5$ is best.

We are only able to consider values of ε_γ up to 2.5 for practical reasons: For larger values of ε_γ , it becomes difficult to maintain realizability of the numerical solution. This indicates that the assumption we make that $\|\bar{\alpha} - \hat{\alpha}\|_1 \leq 5\|\mathbf{d}(\bar{\alpha})\|_1$ (see (3.8)) no longer holds, a result which is not surprising since we only expect this inequality to hold when $\bar{\alpha}$ is near $\hat{\alpha}$.

In the future, the parallelizability of the moment system will be exploited, so the total CPU time is not the best measure of the wall time we expect to need. We make a rough guess of a lower bound on the time that a fully parallel implementation

would need as follows. First, let t^{CPU} indicate the total CPU time not including (again, not including the time spent calculating the source for the manufactured solution). Then we subtract the time spent in the optimization. This difference we divide by n , the number of spatial cells, indicating the assumption that the calculation is fully parallelized over space. Then, let t_{jk}^{opt} be the CPU time needed to perform the optimization at cell j at time step k . Next, we assume that the numerical PDE solver must wait for the last optimization problem to finish before proceeding. Therefore, our parallel estimate is given by

$$t^{\text{par}} := \frac{t^{\text{CPU}} - \sum_{j,k} t_{jk}^{\text{opt}}}{N_x} + \sum_k \max_{1 \leq j \leq N_x} t_{jk}^{\text{opt}} \quad (5.12)$$

This estimate does not attempt to include data communication costs. Figure 5.2(b) again shows the highest γ tolerance, $\varepsilon_\gamma = 2.5$, to be the best choice.

5.1.3 The effect of quadrature on accuracy

Finally, we performed a few experiments varying the fixed number of quadrature points n_Q per half-interval $\mu \in [-1, 0]$ and $\mu \in [0, 1]$ to see their effect on the accuracy of the final solution. We performed these tests for three different spatial resolutions and present the results in Table 5.11. The results show that increasing the n_Q does not improve accuracy.

Table 5.11: L^1 errors for n_Q tests.

N_x	n_Q		
	20	60	100
400	3.313e-06	4.075e-06	3.937e-06
800	1.931e-06	1.442e-06	1.307e-06
1200	8.794e-07	8.798e-07	8.874e-07

5.2 Standard Test problems

Now we use our algorithm on the two standard test problems used in [12]. Most of our tests here simulate the M_{15} model. We chose $N = 15$ for our tests because we want to test our algorithm on problems slightly harder than those likely to be faced by an end-user. Indeed, an end-user is likely to choose a smaller order to see substantial benefits of reducing the size of the velocity-space discretization.

While we present other results in [37], here we present what may be our most accurate results by using more conservative regularization parameters. We let $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}\}$ and only increase r after $k_0 = 40$ iterations. (In the aforementioned work, we include higher values of r and increase r more aggressively in order to decrease the total number of iterations taken.)

In this section, we use the smallest tolerance on γ that we can, $\varepsilon_\gamma = 0.01$. We found that smaller values of this tolerance are often prohibitively difficult to satisfy in double precision. While the results of Section 5.1.2 indicate that higher values of ε_γ are more efficient, this lower value of ε_γ makes the problem a more challenging test for our optimization algorithm.

5.2.1 Plane source

In this problem, we model an infinite domain with a purely scattering medium $\sigma_t = \sigma_s = 1$ and an impulse initial condition

$$\mathbf{u}(x, 0) = \delta(x) + F_{\text{floor}} \langle 1 \rangle , \quad (5.13)$$

where $F_{\text{floor}} = 0.5 \times 10^{-8}$ is used to keep moments away from the realizable boundary (at $\mathbf{u} = 0$). Although the problem is posed on an infinite domain, a finite domain is required for practical computation and boundary conditions must be specified. As in [12], we approximate the infinite domain by the interval $[x_L, x_R] = [-D/2, D/2]$, where $D := 2t_f + 0.2$ is chosen so that the boundary has negligible effects on the solution. At the right and left ends of the boundary, we enforce the boundary conditions

$$\mathbf{u}(x_L, t) = \mathbf{u}(x_R, t) = F_{\text{floor}} \langle 1 \rangle \quad (5.14)$$

for $t \geq 0$.

5.2.1.1 M_{15} simulation

Figure 5.3 presents a high-resolution ($N_x = 3000$) simulation of the M_{15} model of the plane source problem. The local particle concentration shown in Figure 5.3(a) shows several waves as observed in [12, 30], and Figures 5.3(b) and 5.3(d) confirm that the hardest problems occur around the characteristics $x = \pm t$ where particles enter a (near) vacuum. The maximum number of optimization iterations needed in

a single time step in one cell (off-scale in Figure 5.3(b)) was 218, though 98.76% of all problems were solved in three or fewer iterations, and the mean number of iterations is 1.88.

While we are unable to solve this problem without regularization, we only need to regularize about 0.008715% of the nontrivial optimization problems. Further, the histogram in Figure 5.3(f) shows that we we regularize, we use the smallest value of r , which here is 10^{-8} .

Figure 5.4 shows how the solution becomes closer to the realizable boundary as the spatial mesh is resolved. Here, $N_x \in \{200, 400, 600, \dots, 3000\}$. The decrease is consistent with our expectations, although $\min \rho_{\partial\mathcal{R}}(\mathbf{u}(x, t))$ is quite small—($O(10^{-11})$)—even for the coarsest mesh.

5.2.1.2 Convergence in N

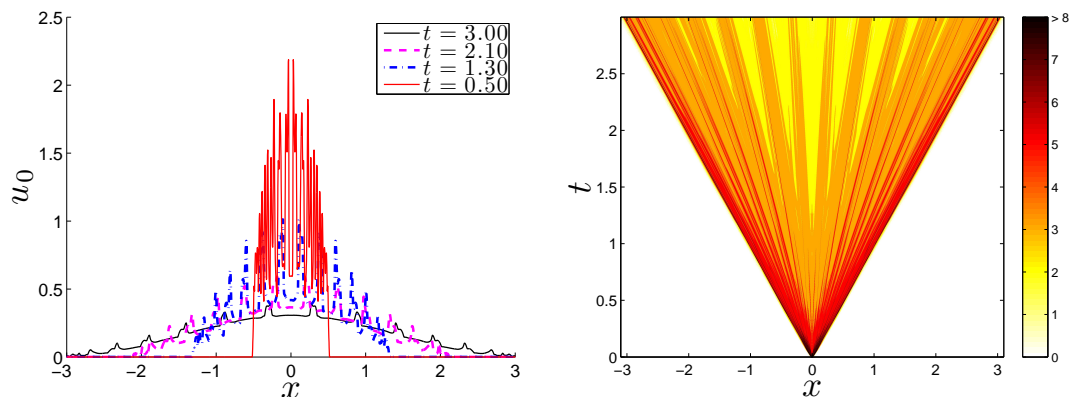
The robustness of our optimization algorithm allows us to compute solutions even for extremely high N . The highest N we have simulated is M_{199} with 600 cells. In this section, we display how the M_N model converges in N . We use $N_x = 600$ cells throughout, and choose $n_{\mathcal{Q}} = \max(20, N + 5)$ quadrature points per half interval so that the Hessian always has full rank.

Figure 5.5 shows that while the number of waves increases with N , their amplitude decreases. However, a small peak at $x = 0$ remains even in M_{199} .

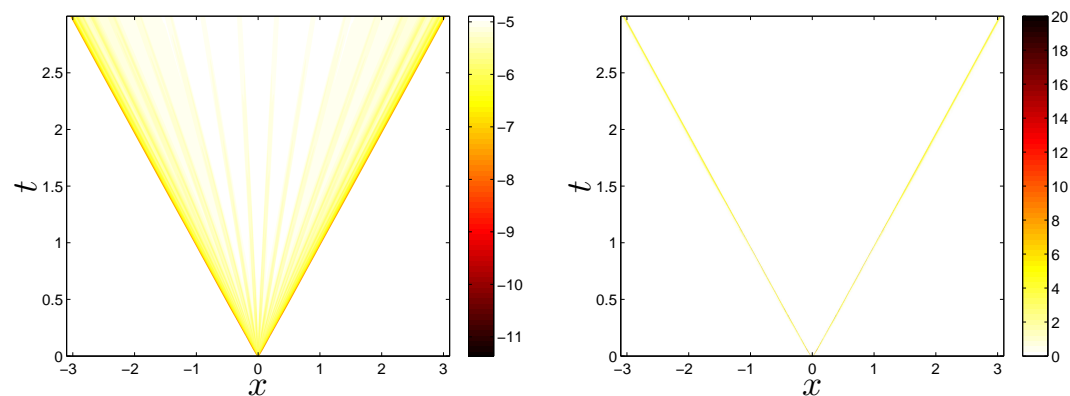
5.2.1.3 Effects of excessive regularization

The results with the manufactured solution in Section 5.1 suggest that regularization errors are quite small when regularization is not performed too frequently. In order to illustrate how the regularization affects the solution when it is performed too frequently, we decreased k_0 to 2 and augmented the sequence $\{r_\ell\}$ to $\{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$.

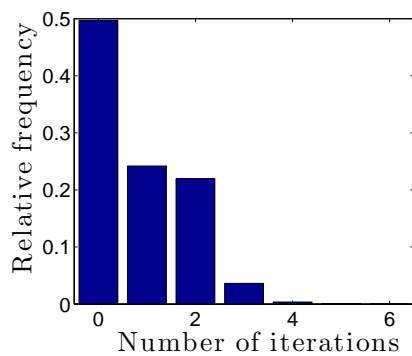
The results for the plane source problem are in Figure 5.6. Comparing Figure 5.6(a) to Figure 5.3(a), we can see that the regularization heavily slows down the movement of particles and smooths out the profile of the solution.



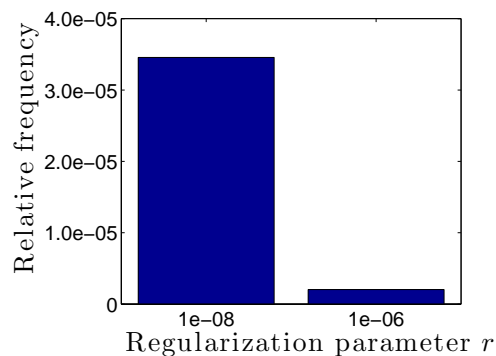
(a) Snapshots of the local particle concentration $u_0(x, t)$ of the solution. (b) The total number of optimization iterations at each mesh point.



(c) The measure of distance to the boundary, $\rho_{\partial\mathcal{R}}(\mathbf{u}(x, t))$ for the moments on the mesh. (d) The extra iterations needed to bring γ below the tolerance once the gradient tolerance had been satisfied.



(e) A histogram of the number of optimization iterations needed.



(f) A histogram of the values of r chosen.

Figure 5.3: A simulation of the M_{15} model of the plane source problem with $N_x = 3000$ cells.

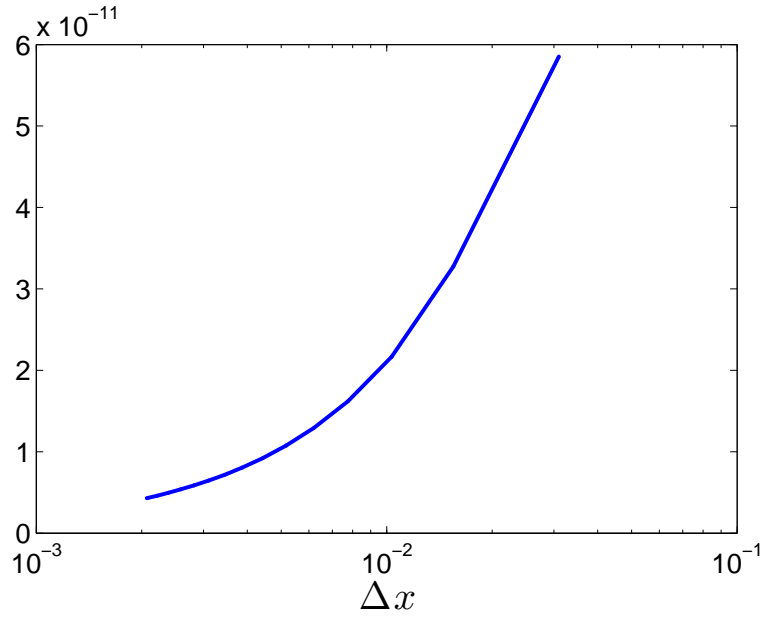


Figure 5.4: The minimum value of $\rho_{\partial\mathcal{R}}(\mathbf{u}(x, t))$ over the space-time mesh for $t \leq 3$ as the cell size is decreased.

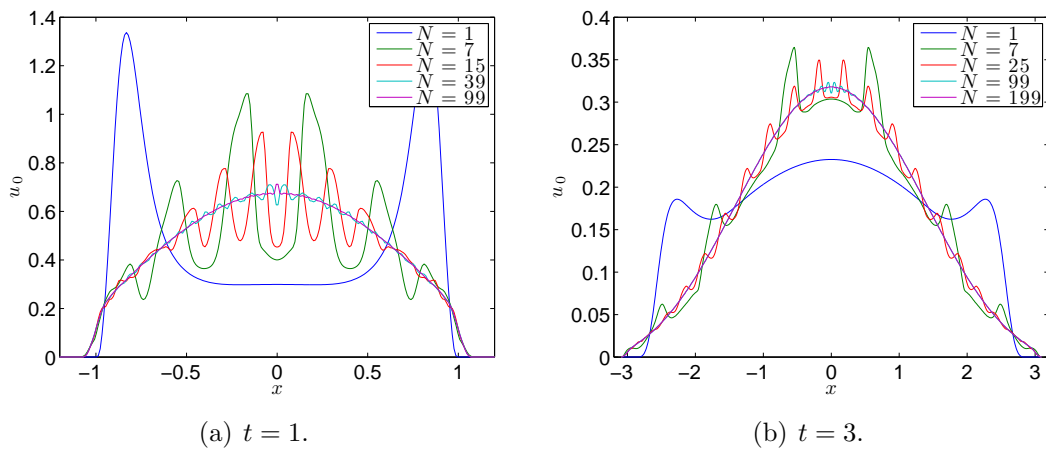
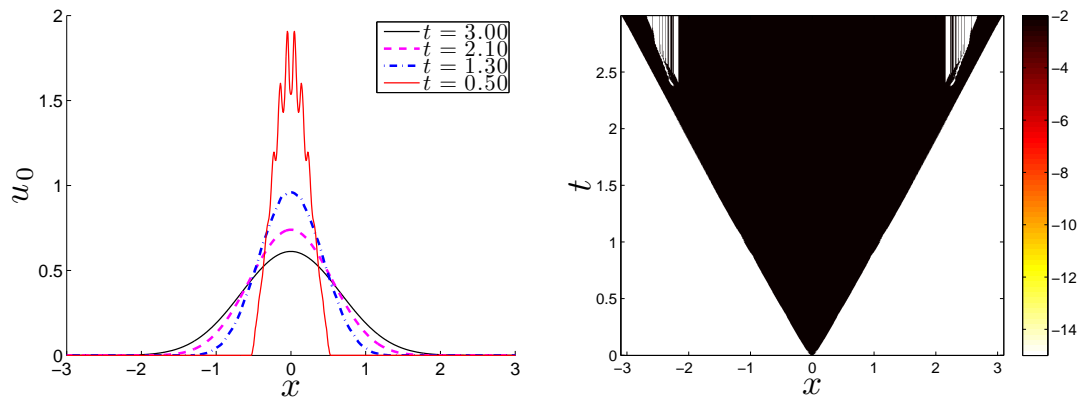


Figure 5.5: The convergence of the plane-source simulation as $N \rightarrow \infty$.



(a) Snapshots of the local particle concentration $u_0(x, t)$ of the solution. (b) The regularization values on the mesh.

Figure 5.6: The simulation with excessive regularization on the plane source problem. Here $k_0 = 2$, and $\lambda = 0$, and $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$.

5.2.2 Two-beam instability

In this problem, particles constantly stream into the domain from the left at $x_L = -0.5$ and the right at $x_R = 0.5$ into the initially (almost) vacuous interior. There is no scattering: $\sigma_s = 0$, while $\sigma_t = 2$. To model ‘forward-peaked’ boundary conditions of the kinetic equations,

$$F_L(\mu, t) = \exp(-10(\mu - 1)^2), \quad \mu > 0, t \geq 0, \quad (5.15a)$$

$$F_R(\mu, t) = \exp(-10(\mu + 1)^2), \quad \mu < 0, t \geq 0, \quad (5.15b)$$

$$F_0(x, \mu) \equiv F_{\text{floor}}, \quad \mu \in [-1, 1], x \in [x_L, x_R], \quad (5.15c)$$

we use the following boundary conditions for the moment system:

$$\mathbf{u}(x_L, t) = \langle \mathbf{m} \exp(-10(\mu - 1)^2) \rangle, \quad (5.16a)$$

$$\mathbf{u}(x_R, t) = \langle \mathbf{m} \exp(-10(\mu + 1)^2) \rangle. \quad (5.16b)$$

On the interior, the initial condition is isotropic with $u_0(x, 0) \equiv F_{\text{floor}} \langle 1 \rangle$.

With the boundary conditions (5.15), the kinetic equation tends toward a steady state given by

$$F_{\text{ss}} = \begin{cases} \exp(-\sigma_a(x - x_L)/\mu - 10(\mu - 1)^2) & \text{if } \mu > 0, \\ \exp(-\sigma_a(x - x_R)/\mu - 10(\mu + 1)^2) & \text{if } \mu < 0. \end{cases} \quad (5.17)$$

5.2.2.1 M_{15} simulation

Figure 5.7 demonstrates the results of a $N_x = 1000$ cell simulation. Small waves in the transient solutions shown in Figure 5.7(a) are noticeable, and Figures 5.7(b) and 5.7(d) indicate that these small oscillations lead to optimization problems which are nearly as difficult as those on the characteristics $x = 0.5 \mp t$ where the particles enter the initial (near) vacuum. The maximum number of optimization iterations needed in a single time step in one cell (off-scale in Figure 5.7(b)) was 306. Again, however, the mean number of iterations is quite small (1.53), and 97.22% of the optimization problems are solved in three iterations or fewer.

We are again unable to solve this problem without regularization. In this simulation we regularized 0.1158% of the nontrivial problems, slightly more regularization than the plane source problem. Again the histogram in Figure 5.3(f) shows that when we regularize, we use the smallest value of r , which here is 10^{-8} .

Figure 5.7(a) includes a green curve indicating the local particle concentration of the steady-state kinetic solution (5.17). It is indistinguishable from the moment solution, but (as hinted to by the iterations never quite going to zero in Figure 5.7(b)) the numerical moment solution never converges. In Figure 5.8, we plot an L^1 difference between time steps given by

$$\mathbf{e}^n = \Delta x \sum_{j=1}^{N_x} |\mathbf{u}_j^n - \mathbf{u}_j^{n-1}|, \quad n \geq 1, \quad (5.18)$$

where the absolute value is taken component-wise. In Figure 5.8, we plot the first five components of \mathbf{e} . The zeroth-component remains the largest, at just below 10^{-8} ,

and the following components are slightly smaller. This generally trend continues for the components up the fifteenth, though those components tend to have more variance.

5.2.2.2 Convergence in N

In this section as in Section 5.2.1.2, we again consider $N_x = 600$ cell simulations and let $n_{\mathcal{Q}} = \max(20, N + 5)$.

Figure 5.9 shows how transient solutions converge as N increases. At $t = 0.4$, the solutions are qualitatively difficult to distinguish for $N \geq 7$, though they do have distinct shapes which do become less pronounced as N increases. At $t = 0.7$ in Figure 5.9(b) after particles from the boundaries have crosses in the middle $x = 0$, we see the beginnings of the well-known nonphysical shock in the M_1 model. The solutions for higher N have oscillations in the middle which decrease as N increases.

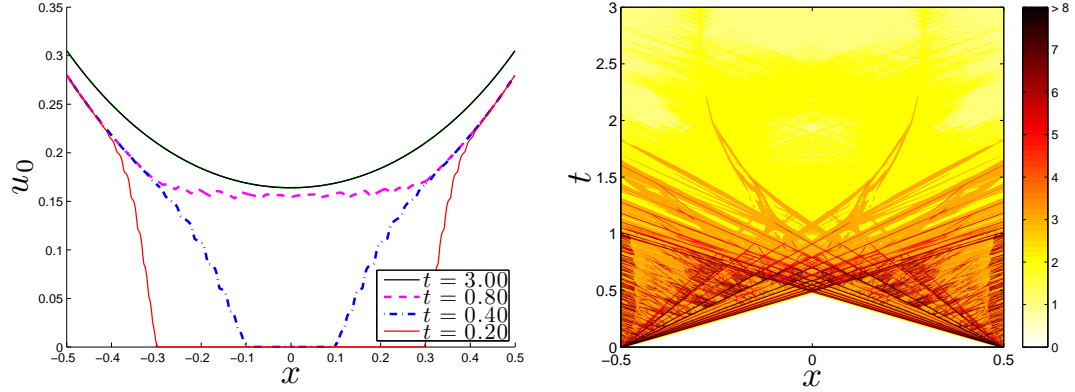
Figure 5.2.2.2 compares steady-state solutions as N increases. These results differ from those presented in [12], though there the author used isotropic boundary conditions. Here we see that the moment solution agrees with the steady-state kinetic solution quite well for $N \geq 5$.

5.2.2.3 Effects of excessive regularization

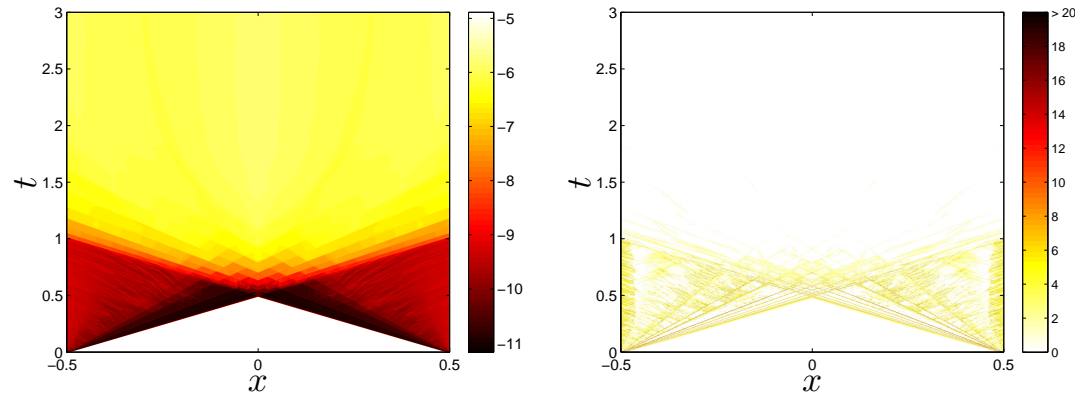
We repeat the parameter changes made in Section 5.2.1.3 in order to examine the effects of an overly aggressive regularization scheme. Figure 5.11 displays the results. Again, comparing Figure 5.11(a) to Figure 5.7(a), we see here that the

regularization significantly dampens the movement of particles across the domain, more strikingly than in the plane source problem. At $t = 3$, the system is not close to steady-state, and possibly is converging to an incorrect steady-state solution.

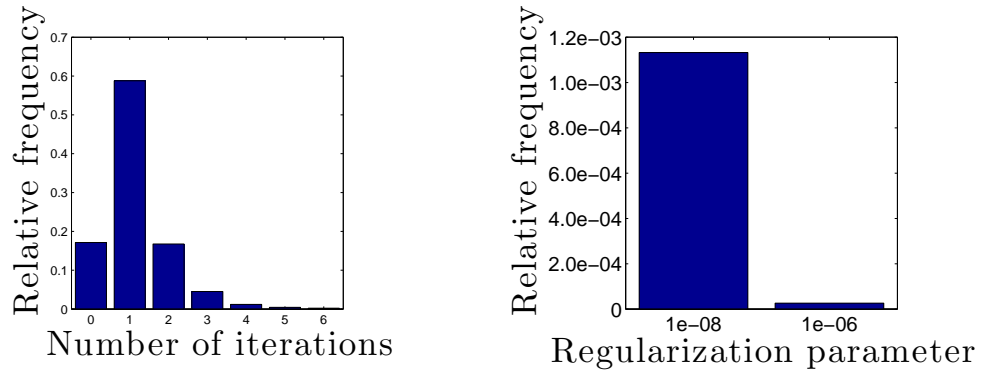
The results in this chapter show that the our final optimization algorithm, Algorithm 2, is robust. The experiments with manufactured solutions are not complete but promisingly indicate that the errors introduced by isotropic regularization are small. This method also allowed us to test a few of the many parameters in the overall solver. We showed with the test problems that the optimization works even for high spatial resolution or very high N with only small amounts of regularization. The low mean iteration counts for these test problems showed that M_N models may indeed become practical.



(a) Snapshots of the local particle concentration $u_0(x, t)$ of the solution. (b) The total number of optimization iterations at each mesh point.



(c) The measure of distance to the boundary, $\rho_{\partial\mathcal{R}}(\mathbf{u}(x, t))$ for the moments on the mesh. (d) The extra iterations needed to bring γ below the tolerance once the gradient tolerance had been satisfied.



(e) A histogram of the number of optimization iterations needed. (f) A histogram of the values of r chosen.

Figure 5.7: A simulation of the M_{15} model of the two-beam instability with $N_x = 1000$ cells.

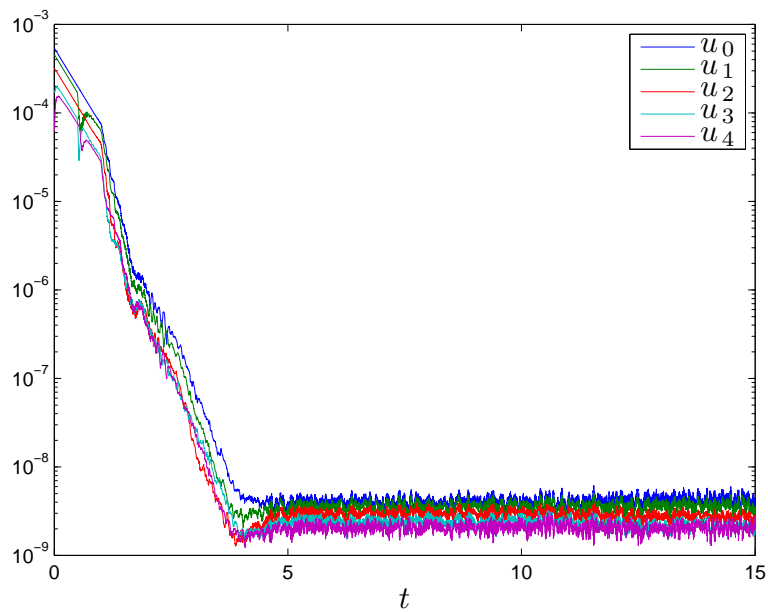


Figure 5.8: Convergence of individual moments in the M_{15} beams simulation.

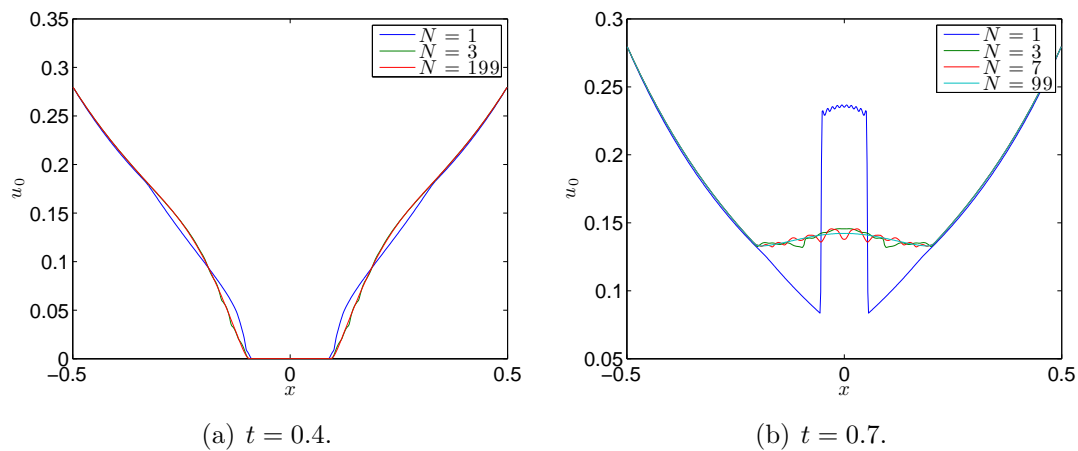


Figure 5.9: The convergence of transient profiles of the two-beam instability simulation as $N \rightarrow \infty$.

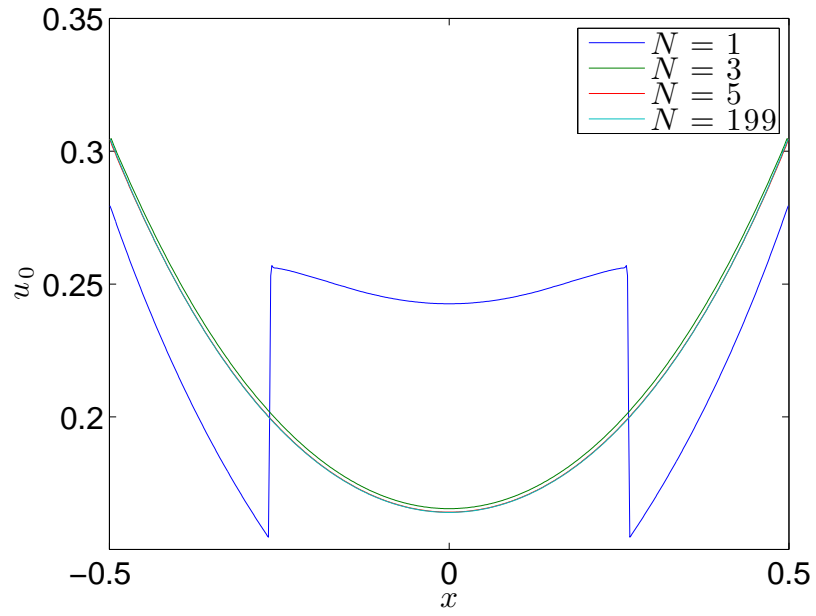
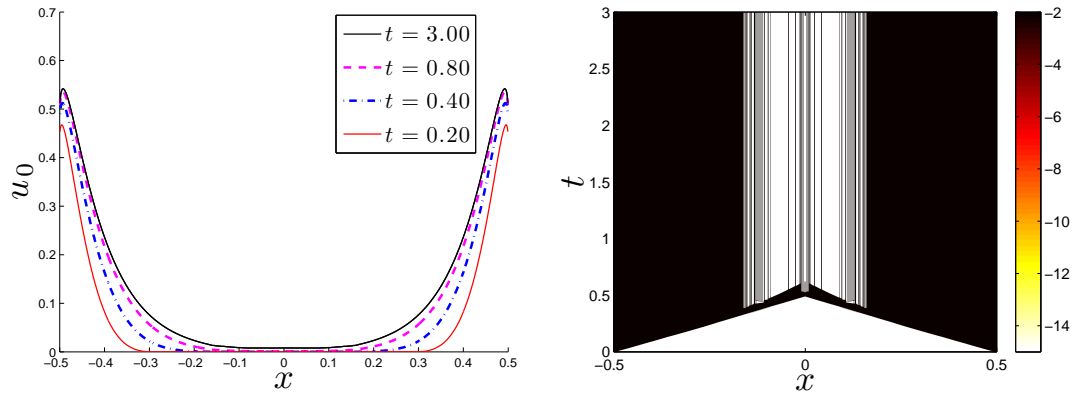


Figure 5.10: Steady-state M_N beams solutions.



(a) Snapshots of the local particle concentration $u_0(x, t)$ of the solution. (b) The regularization values on the mesh.

Figure 5.11: The simulation with excessive regularization on the two-beam instability. Here $k_0 = 2$, and $\lambda = 0$, and $\{r_\ell\} = \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$.

Chapter 6

Conclusions and Directions for Future Work

In this work, we directly used numerical optimization in simulations of entropy-based moment closures for a one-dimensional linear kinetic equation. Each optimization problem is often simple to solve with Newton's method, but strongly anisotropic moments which occur near the boundary of the set of realizable moments present difficulties for optimizer because the Hessian of the objective function near the solution approaches singularity. We exposed in detail how these anisotropies cause problems for the optimization.

We showed that when the angular integral is replaced by quadrature in the objective function, the set of moments for which the optimization has a solution is reduced since the quadrature cannot resolve moments generated by atomic distributions off the set of quadrature nodes. We characterized this reduced realizable set and observed that it is invariant in the numerical kinetic scheme when a fixed quadrature is used. A further conclusion here is that using an adaptive quadrature in the optimization and the kinetic scheme can, somewhat counter-intuitively, make the problem harder to solve.

We tested the adaptive-basis method proposed by Abramov in [18, 19] for problems on unbounded integration domains and modified it for our problem. We found that it can solve problems closer to the realizable boundary but cannot get

close enough to solve all the problems that show up in moment PDE.

Since the conditioning of some commonly occurring problems can be arbitrarily poor, for any given finite precision there are problems which cannot be solved. Therefore, any robust numerical method for this problem must make approximations near the realizable boundary. We proposed an isotropic regularization for these problems to introduce a small artificial numerical scattering to replace difficult problems with similar tractable ones. Our experiments have given strong evidence that this regularization quickly finds tractable problems, and the error introduced by the method is small.

Due to the lack of benchmark solutions to difficult test problems (particularly those with strong anisotropies), we experimented with the method of manufactured solutions. We outlined a way to use the method on this problem, namely by specifying the solution through the multipliers and using first-order models. Two attempts with this method were presented, but neither is sufficiently satisfactory yet. The solution specified by the first attempt varies by many orders of magnitude over the spatial domain. The solutions calculated according to the second attempt do not show second-order convergence. We remain confident, however, that an $O(1)$ solution that converges appropriately can be found soon.

The success we have had here, particularly in displaying that the mean number of optimization iterations is quite low (and consequently the percentage of computational effort spent on the optimization is reasonable), indicates that the obvious but important future directions are worth pursuing. These include nonlinear kinetic equations, higher-spatial dimensions, different entropy choices (for example

Bose-Einstein, Fermi-Dirac, and positive P_N), and parallel implementations.

But there is still room for improvement. While the numerical results with the isotropic regularization are promising, the technique needs to be analyzed. In particular, we would like theoretical assurance that the regularization indeed makes the optimization problem easier, at least when the moments \mathbf{u} are near the realizable boundary. Showing that the $\kappa(H(\hat{\boldsymbol{\alpha}}(\mathbf{v}(r))))$ decreases with r would be a helpful result. Additionally, the error introduced into the solution $\mathbf{u}(x, t)$ by regularization should be further understood.

The problems with bringing γ down discussed in Section 3.2.3 indicate that some regularization errors could be avoided if we had a different method for maintaining realizability of the numerical solution. The current method gives a strong sufficient condition for maintaining realizability, so it could be useful to investigate how far this condition is from necessity or to consider alternative numerical schemes that don't require the optimization to be solved with such high precision. However, it is certainly possible that such schemes would, in the process of accepting less accurate and less anisotropic solutions from the optimization, end up making the same errors as those introduced by the isotropic regularization.

Appendix A

Details and Properties of the Numerical Scheme

A.1 The numerical scheme

Let $\Delta x = (x_R - x_L)/N_x$ and $\Delta t > 0$ be given mesh parameters, and let $\{x_j\}_{j=-1}^{N_x+2} \times \{t^n\}_{n=0}^{N_t}$ be a uniform space-time mesh defined by $x_j := x_L + (j - 0.5)\Delta x$ and $t^n := n\Delta t$. The values x_j define the centers of contiguous spatial cells $I_j := (x_{j-1/2}, x_{j+1/2})$, where $x_{j\pm 1/2} := x_j \pm \Delta x/2$. The cells with indices $j \in \{-1, 0, N_x + 1, N_x + 2\}$ are ‘ghost cells’, which are not part of the physical domain but are used to implement boundary conditions.

We approximate \mathbf{u} numerically via its cell averages, letting

$$\mathbf{u}_j(t) \simeq \frac{1}{\Delta x} \int_{I_j} \mathbf{u}(x, t) dx, \quad j \in \{-1, \dots, N_x + 2\}. \quad (\text{A.1})$$

The semi-discrete, numerical scheme for (2.12) which defines \mathbf{u}_j on the interior of the domain is

$$\partial_t \mathbf{u}_j + \frac{\mathbf{f}_{j+1/2} - \mathbf{f}_{j-1/2}}{\Delta x} + \sigma_t \mathbf{u}_j = \sigma_s Q \mathbf{u}_j, \quad j \in \{1, \dots, N_x\}, \quad (\text{A.2})$$

the numerical flux $\mathbf{f}_{j+1/2}$ being given by

$$\mathbf{f}_{j+1/2} := \langle \mu \mathbf{m} \hat{G}_{j+1/2} \rangle, \quad j \in \{0, \dots, N_x\}, \quad (\text{A.3})$$

and $\hat{G}_{j+1/2}$ is an approximation of the entropy ansatz at the cell edge. These edge values are defined based on the sign of μ , via up-winding:

$$\hat{G}_{j+1/2}(\mu, t) := \begin{cases} \hat{G}_j(\mu, t) + \frac{\Delta x}{2} \hat{s}_j(\mu, t), & \mu > 0 \\ \hat{G}_{j+1}(\mu, t) - \frac{\Delta x}{2} \hat{s}_{j+1}(\mu, t), & \mu < 0 \end{cases}, \quad j \in \{0, \dots, N_x\}, \quad (\text{A.4})$$

where \hat{G}_j is the entropy ansatz associated to \mathbf{u}_j via (2.8):

$$\hat{G}_j(\mu, t) := G_{\hat{\alpha}(\mathbf{u}_j(t))}(\mu), \quad j \in \{-1, \dots, N_x + 2\}. \quad (\text{A.5})$$

For $j \in \{0, \dots, N_x + 1\}$, the quantity \hat{s}_j is an approximation of the spatial derivative of \hat{G} in cell I_j :

$$\hat{s}_j = \text{minmod} \left\{ \theta \frac{\hat{G}_j - \hat{G}_{j-1}}{\Delta x}, \frac{\hat{G}_{j+1} - \hat{G}_{j-1}}{2\Delta x}, \theta \frac{\hat{G}_{j+1} - \hat{G}_j}{\Delta x} \right\}, \quad (\text{A.6})$$

where $1 \leq \theta \leq 2$ [42, 43].⁽¹⁾ The minmod function selects the real number with smallest absolute value in the convex hull of its arguments. (Note that, in (A.6), \hat{G}_j is needed for $j \in \{-1, 0, \dots, N_x + 2\}$, which shows the need for four ghost cells, two on each side of (x_L, x_R) .)

As mentioned in Section 2.2, boundary conditions for moment systems remain an open question. In our implementation, we prescribe boundary conditions by specifying moments on the four ghost cells. For periodic boundaries used in some

¹Any value of $\theta \in [1, 2]$ will yield a second-order scheme and, roughly speaking, larger values of θ decrease numerical diffusion in the scheme. When $\theta = 1$, monotonic cell averages yield monotonic reconstructions $G_j(\mu, t) + s_j(\mu, t)(x - x_j)$. When $\theta = 2$, edge values are bounded by neighboring cell averages.

test problems, we simply set

$$\mathbf{u}_{-1}(t) = \mathbf{u}_{N_x-1}(t), \quad \mathbf{u}_0(t) = \mathbf{u}_{N_x}(t), \quad \mathbf{u}_{N_x+1}(t) = \mathbf{u}_1(t), \quad \mathbf{u}_{N_x+2}(t) = \mathbf{u}_2(t). \quad (\text{A.7})$$

For physical boundary conditions, moments in ghost cells are defined by extending the definitions of $F_L(\mu, t)$ and $F_R(\mu, t)$ to all μ and then taking moments:

$$\mathbf{u}_{-1}(t) = \mathbf{u}_0(t) = \langle \mathbf{m}F_L(\mu, t) \rangle, \quad \mathbf{u}_{N_x+1}(t) = \mathbf{u}_{N_x+2}(t) = \langle \mathbf{m}F_R(\mu, t) \rangle. \quad (\text{A.8})$$

While reasonable, this is clearly not the only option. Further discussion of this issue is given in [12, 44].

To integrate (A.2) in time, we use the optimal, second-order strong-stability-preserving Runge-Kutta (SSP-RK2) method [32], also known as Heun's method or the improved Euler method. We approximate $\mathbf{u}_j(t^n) \simeq \mathbf{u}_j^n$ and let \mathbf{u}^n denote the array containing $\{\mathbf{u}_j^n\}_{j=-1}^{N_x+1}$. For (A.2) with (A.7) or (A.8) in the abstract form $\partial_t \mathbf{u} = L(\mathbf{u})$, the SSP-RK2 method with initial stage $\mathbf{u}^{(0)} := \mathbf{u}^n$ is given by

$$\mathbf{u}^{(1)} := \mathbf{u}^{(0)} + \Delta t L(\mathbf{u}^n), \quad \mathbf{u}^{(2)} := \mathbf{u}^{(1)} + \Delta t L(\mathbf{u}^{(1)}), \quad \mathbf{u}^{n+1} := \frac{1}{2} (\mathbf{u}^{(0)} + \mathbf{u}^{(2)}) \quad (\text{A.9})$$

for all $n \in \{0, \dots, N_t - 1\}$.

As discussed in [12], kinetic scheme (A.2)–(A.9) is very inefficient in diffusive regimes, where σ_t is large. In such regimes, accuracy requirements dictate that the spatial and temporal mesh depends inversely on σ_t , even though the solution profile

varies on an $O(1)$ scale. However, for the test cases considered later in this paper, σ_t is an $O(1)$ quantity.

A.2 Proof of \mathcal{R} -invariance of the numerical scheme

Theorem 7. *Suppose that $\mathbf{u}_j^n \in \mathcal{R}$ for $j \in \{-1, \dots, N_x + 2\}$. If \mathbf{u}^{n+1} is defined via the kinetic scheme (A.2),(2.18),(A.4) –(A.9) with bars replacing hats in (23)–(25) and with time-step restriction*

$$\gamma_{\max} \frac{\Delta t}{\Delta x} \frac{2 + \theta}{2} + \sigma_t \Delta t < 1 \quad (\text{A.10})$$

and if the moments in the ghost cells are realizable at each stage of the Runge-Kutta scheme (A.9), then $\mathbf{u}_j^{n+1} \in \mathcal{R}$ for $j \in \{1, \dots, N_x\}$.

Proof. We show for $m \in \{1, 2\}$ that $\mathbf{u}_j^{(m-1)} \in \mathcal{R}$ for $j \in \{-1, \dots, N_x + 2\}$ implies $\mathbf{u}_j^{(m)} \in \mathcal{R}$ for $j \in \{1, \dots, N_x\}$. Realizability for the subvectors of \mathbf{u}^{n+1} then follows from (A.9) and Theorem 2 (convexity of \mathcal{R}). The key point is to observe that

$$\mathbf{u}_j^{(m)} = \langle \mathbf{m} \Phi_j^{(m)} \rangle, \quad j \in \{1, \dots, N_x\}, \quad m \in \{1, 2\}, \quad (\text{A.11})$$

where

$$\Phi_j^{(m)} := \hat{G}_j^{(m-1)} - \mu \frac{\Delta t}{\Delta x} \left(\bar{G}_{j+1/2}^{(m-1)} - \bar{G}_{j-1/2}^{(m-1)} \right) + \Delta t \left(-\sigma_t \hat{G}_j^{(m-1)} + \frac{\sigma_s}{2} \langle \hat{G}_j^{(m-1)} \rangle \right). \quad (\text{A.12})$$

Thus one need only show that $\Phi_j^{(m)} \geq 0$. Stripping away positive terms on the

right-hand side of (A.12) gives

$$\Phi_j^{(m)} \geq \hat{G}_j^{(m-1)} - \mu \frac{\Delta t}{\Delta x} \bar{G}_{j+1/2}^{(m-1)} - \Delta t \sigma_t \hat{G}_j^{(m-1)}. \quad (\text{A.13})$$

Assume $\mu \geq 0$. (The case $\mu < 0$ follows from an analogous argument.) If $\bar{s}_j > 0$ (so all arguments of the minmod in (A.6) are non-negative), we have (with bars instead of hats)

$$\bar{s}_j^{(m-1)} \leq \theta \frac{\bar{G}_j^{(m-1)} - \bar{G}_{j-1}^{(m-1)}}{\Delta x} \quad (\text{A.14})$$

so that, using (A.4),

$$\bar{G}_{j+1/2}^{(m-1)} \leq \left(1 + \frac{\theta}{2}\right) \bar{G}_j^{(m-1)} - \frac{\theta}{2} \bar{G}_{j-1}^{(m-1)} \leq \frac{2+\theta}{2} \bar{G}_j^{(m-1)}. \quad (\text{A.15})$$

Substituting (A.15) into (A.13) and invoking the definition of $\gamma^{j,(m-1)}$ from (2.19) gives

$$\Phi_j^{(m)} \geq \left(1 - \mu \gamma^{j,(m-1)} \frac{\Delta t}{\Delta x} \frac{2+\theta}{2} - \Delta t \sigma_t\right) \hat{G}_j^{(m-1)}. \quad (\text{A.16})$$

From (A.16), it is clear that (A.10) implies non-negativity of $\Phi_j^{(m)}$. On the other hand, if $\bar{s}_j \leq 0$, we obtain

$$\Phi_j^{(m)} \geq \left(1 - \mu \gamma^{j,(m-1)} \frac{\Delta t}{\Delta x} - \Delta t \sigma_t\right) \hat{G}_j^{(m-1)}. \quad (\text{A.17})$$

The positivity of the left-hand side of (A.17) is guaranteed by the condition

$$\mu \gamma^{j,(m-1)} \frac{\Delta t}{\Delta x} + \Delta t \sigma_t < 1, \quad (\text{A.18})$$

which is weaker than (A.10). This concludes the proof. \square

Remark 3. *The reader should note the following:*

1. *The proof of Theorem 4 does not depend on the specific form of \hat{G}_j or \bar{G}_j , only on the fact that they are positive. Thus the theorem applies to different types of closures and different types of numerical error, so long as positivity of the two approximations is maintained.*
2. *Setting $\gamma_{max} = 1$ recovers the time-step restriction for the case when there is no error in approximating the ansatz. If further $\sigma_t = 0$, then the corresponding time step restriction is exactly what is required to maintain positivity for a single Euler step applied to a semi-discrete MUSCL scheme [45] for a linear advection equation with speed one (the maximum value of $|\mu|$). This is not by accident; in this case, the kinetic scheme (A.2)–(A.9) is equivalent to the moments of a semi-discrete MUSCL scheme for the transport equation (2.1) with initial condition \hat{G} .*
3. *The quantity γ_{max} depends on the solution values at the intermediate Runge-Kutta stage $\mathbf{u}^{(1)}$. This leaves a user with two options: either (i) set an upper bound for γ_{max} to determine a suitable Δt and then require that the optimization error for every cell and every stage is below that bound or (ii) check the error at the intermediate stage and, if it is too high, exit the Runge-Kutta algorithm, go back to the previous time step, and choose a smaller value for Δt . In our implementation, we take the former approach.*

4. Equation (A.16) shows that the less conservative definition of γ_{\max} given by $\gamma'_{\max} := \max_{m,j,\mu} \{\mu \gamma^{j,(m)}(\mu)\}$ is sufficient to guarantee nonnegativity. This definition was not used in our implementation.

Bibliography

- [1] E. E. Lewis and Jr. W. F. Miller. *Computational Methods in Neutron Transport*. John Wiley and Sons, New York, 1984.
- [2] Cory D. Hauck and Ryan G. McClarren. Positive P_N closures. *SIAM Journal on Scientific Computing*, 32(5):2603–2626, 2010.
- [3] K. A. Mathews. On the propagation of rays in discrete ordinates. *Nucl. Sci. Eng.*, 132:155, 1999.
- [4] J. A. Fleck, Jr. and J. D. Cummings, Jr. An implicit monte carlo scheme for calculating time and frequency dependent nonlinear radiation transport. *J. Comput. Phys.*, 8(3):313–342, 1971.
- [5] A. M. Anile, S. Pennisi, and M. Sammartino. A thermodynamical approach to Eddington factors. *J. Math. Phys.*, 32:544 – 550, 1991.
- [6] E. T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106:620 – 630, 1957.
- [7] C. D. Levermore. Moment closure hierarchies for kinetic theory. *J. Stat. Phys.*, 83:1021–1065, 1996.
- [8] M. Junk. Domain of definition of Levermore’s five moment system. *J. Stat. Phys.*, 93(5-6):1143–1167, 1998.
- [9] Cory D. Hauck, C. David Levermore, and André L. Tits. Convex duality and entropy-based moment closures: Characterizing degenerate densities. *SIAM J. Control Optim.*, 47(4):1977–2015, 2008.
- [10] B. Dubroca and A. Klar. Half-moment closure for radiative transfer equations. *J. Comput. Phys.*, 180(2):584–596, 2002.
- [11] P. Monreal and M. Frank. Higher order minimum entropy approximations in radiative transfer. preprint.
- [12] C. D. Hauck. High-order entropy-based closures for linear transport in slab geometry. *Comm. Math. Sci.*, 9, 2011.
- [13] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Math. Statistics*, 43:1470 – 1480, 1972.
- [14] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 1 – 13, 1997.

- [15] K. Bandyopadhyay, A. K. Bhattacharya, P. Biswas, and D. A. Drabold. Maximum entropy and the problem of moments: A stable algorithm. *Phys. Rev. E*, 71:057701, May 2005.
- [16] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39 – 71, 1996.
- [17] Steven J. Phillips, Miroslav Dudík, and Robert E. Schapire. A maximum entropy approach to species distribution modeling. In *Proc. of the Twenty-First Int'l Conf. on Machine Learning*, pages 655 – 662, 2004.
- [18] R. V. Abramov. The multidimensional moment-constrained maximum entropy problem: A BFGS algorithm with constraint scaling. *Journal of Computational Physics*, 228:96–108, January 2009.
- [19] R. Abramov. The multidimensional maximum entropy moment problem: A review on numerical methods. *Communications in Mathematical Sciences*, 8:377–392, 2009.
- [20] G. C. Pomraning. Variational boundary conditions for the spherical harmonics approximation to the neutron transport equation. *Ann. Phys.*, 27:193–215, 1964.
- [21] E. W. Larsen and C. G. Pomraning. The P_N theory as an asymptotic limit of transport theory in planar geometry—I: Analysis. *Nucl. Sci. Eng.*, 109:49–75, 1991.
- [22] E. W. Larsen and C. G. Pomraning. The P_N theory as an asymptotic limit of transport theory in planar geometry—II: Numerical results. *Nucl. Sci. Eng.*, 109:76–85, 1991.
- [23] C. D. Levermore. Boundary conditions for moment closures. Presented at Institute for Pure and Applied Mathematics University of California, Los Angeles, CA on May 27, 2009.
- [24] James Alexander Shohat and Jacob David Tamarkin. *The Problem of Moments*. American Mathematical Society, New York, 1943.
- [25] M. Junk. Maximum entropy for reduced moment problems. *Math. Mod. Meth. Appl. S.*, 10(7):1001–1025, 2000.
- [26] J. Schneider. Entropic approximation in kinetic theory. *Math. Model. Numer. Anal.*, 38:541–561, 2004.
- [27] J. M. Borwein and A. S. Lewis. Duality relationships for entropy-like minimization problems. *SIAM J. Control Optim.*, 1:191–205, 1991.
- [28] L. R. Mead and N. Papanicolaou. Maximum entropy in the problem of moments. *Journal of Mathematical Physics*, 25:2404–2417, August 1984.

- [29] Raúl E. Curto and Lawrence A. Fialkow. Recursivness, positivity and truncated moment problems. *Houston Journal of Mathematics*, 4:603–635, 1991.
- [30] G. W. Alldredge, C. D. Hauck, and A. L. Tits. High-order entropy-based closures for linear transport in slab geometry II: A computational study of the optimization problem. *SIAM Journal on Scientific Computing*, 2012. To appear.
- [31] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, New York, 2009.
- [32] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):pp. 89–112, 2001.
- [33] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific J. Math*, 16(1):1–3, 1966.
- [34] C. D. Hauck. Private communication, 2012.
- [35] G. N. Minerbo. Maximum entropy Eddington factors. *J. Quant. Spectrosc. Radiat. Transfer*, 20:541–545, 1978.
- [36] Thomas A. Brunner and James Paul Holloway. One-dimensional Riemann solvers and the maximum entropy closure. *J. Quant Spect. and Radiative Trans*, 69(5):543 – 566, 2001.
- [37] G. W. Alldredge, C. D. Hauck, Dianne O’Leary, and A. L. Tits. Advanced optimization techniques in entropy-based moment closures in slab geometry. *In preparation*, 2013.
- [38] J. N. Lyness. When Not to Use an Automatic Quadrature Routine. *SIAM Review*, 25:63–88, 1983.
- [39] S.P. Boyd and L.Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- [40] Gordon L. Olson. Second-order time evolution of P_N equations for radiation transport. 228(8):3072–3083, May 2009.
- [41] Ryan G. McClarren and Cory D. Hauck. Robust and accurate filtered spherical harmonics expansions for radiative transfer. *Journal of Computational Physics*, 229(16):5597 – 5614, 2010.
- [42] Haim Nessyahu and Eitan Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of Computational Physics*, 87(2):408 – 463, 1990.
- [43] Bram Van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *Journal of Computational Physics*, 23(3):276 – 299, 1977.

- [44] Henning Struchtrup. Kinetic schemes and boundary conditions for moment equations. *Z. Angew. Math. Phys.*, 51(3):346–365, 2000.
- [45] S. Osher. Convergence of Generalized MUSCL Schemes. *SIAM Journal on Numerical Analysis*, 22:947–961, October 1985.