SANDIA REPORT
SAND2022-13436
Printed October 4, 2022

Sandia National Laboratories

Gaussian Mixture Model Solvers for the Boltzmann Equation

Thomas A. Gardiner and Allen C. Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

ABSTRACT

This report documents our experience constructing a numerical method for the collisional Boltzmann equation that is capable of accurately capturing the collisionless through strongly collisional limits. We explore three different functional representations and present a detailed account of a numerical method based on a spatially dependent Gaussian mixture model (GMM). The Kullback-Leibler divergence is used as a closeness measure and various expectation maximization (EM) solution algorithms are implemented to find a compact representation in velocity space for distribution functions that exhibit significant non-Maxwellian character. We discuss issues that appear with this representation over a range of Knudsen numbers for a prototypical test problem and demonstrate that the strongly collisional limit recovers a solution to Euler's equations. Looking forward, this approach is broadly applicable to the non-relativistic and relativistic collisional Vlasov equations.

ACKNOWLEDGMENTS

This work was supported by the Laboratory Directed Research and Development (LDRD) Program at Sandia National Laboratories with project title, "Hybrid Fluid-Kinetic Methods for Collisional Pulsed Power Plasma Modeling," and project number, 222323.

Scot Swan provided coding support for automating the analyses associated with some of the results presented here.

Amelia Hendriksen provided a detailed review of a draft of this report as well as helpful communications and references on concept drift in GMM machine learning.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This report describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the report do not necessarily represent the views of the U.S. Department of Energy or the U.S. Government.

....

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

....

1. INTRODUCTION

The Boltzmann equation provides a kinetic modeling representation of physical systems across a range of collisionality. The equations are very powerful and are only limited by the assumption of binary interactions, choice of effective collision operators and the large 7 dimensional (space, velocity and time) equations to be solved. The ratio of the mean free path of atomic collision relative to the physical length scale of interest is called the Knudsen number, K_n. For highly collisional systems when $K_n \ll 1$, five moment fluid (density, momentum, energy) equations derived from the Boltzmann equation are generally sufficient. However, across the large dynamic range of densities in plasma physics modeling, a more complicated view of the distribution function should be considered and supported. Additional moments beyond 5 may be constructed but even these do not give full access to arbitrary distribution functions. There is continuing interest in finding effective and efficient numerical methods for the Boltzmann equation that can effectively deal with the large dimension added by the additional velocity independent variables. For example, recent work has investigated compression of velocity space by low-rank tensor decomposition approaches [1, 11]. Adaptive discontinuous Petrov-Galerkin methods also show promise for reducing cost in the velocity dimension [28]. Machine learning techniques have been applied to approximate collision operators effectively while improving evaluation efficiency [22].

Given that physical collisions should smooth high frequencies in the velocity domain, it appears useful to investigate parameterized representations of the distribution function that are general, compact, and reduce to a Maxwellian distribution at low Knudsen numbers. We use Maxwellian distribution, fluid model and Gaussian distribution interchangeably. In this report we examine our experience investigating Gaussian mixture model (GMM) representations in velocity space. These compressed representations at the outset appear potentially useful since they may be highly effective at representing both the fluid regimes as well as smooth distributions with multiple features in velocity space.

In Section 2 we discuss our equations of interest and the representations to be examined. For simplicity we will assume a simple Bhatnagar-Gross-Krook (BGK) collision operator. In Section 3 we discuss the GMM mixture model and the classical expectation maximization (EM) algorithm for estimating GMM parameters from data and the Kullback-Leibler divergence as a means for determining the closeness of fit between two different distribution functions. In Section 3.3 we discuss an algorithm for solving the Boltzmann equation numerically given the GMM representation in velocity space. In Section 3.4 we present a prototypical problem and show how the representation can be effective but can also begin to illustrate limitations in the GMM representation that affect the distribution function evolution over a range of Knudsen numbers. We discuss our insights into the importance of the initial parameters assumed for the Gaussian when entering the Expectation Maximization (EM) algorithm compression step. We also discuss our experience with the effectiveness of the EM algorithm and a few acceleration schemes.

2. EQUATIONS AND REPRESENTATIONS

We are ultimately interested in a set of equations for the species plasma distribution function, $f_\alpha(\mathbf{x},\mathbf{u},t)$,

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{u}\cdot\nabla_{\mathbf{x}} f_\alpha + \frac{q_\alpha}{m_\alpha}(\mathbf{E}+\mathbf{u}\times\mathbf{B})\cdot\nabla_{\mathbf{u}} f_\alpha = \sum_\beta C_{\alpha\beta}(f_\alpha, f_\beta) \tag{1}$$

where $\mathbf{E}(\mathbf{x},t)$ and $\mathbf{B}(\mathbf{x},t)$ are the macroscopic electric field and magnetic flux density; q_α and m_α are species charge and mass respectively; $C_{\alpha\beta}$ is the inter-species collision operator, and $f_\alpha(\mathbf{x},\mathbf{u},t)$ is a seven-dimensional phase space of spatial coordinates, velocity, and time. These equations couple to Maxwell's equations

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla\times\mathbf{E} = 0 \tag{2}$$

$$\frac{\partial \varepsilon\mathbf{E}}{\partial t} + \mathbf{J} = \nabla\times(\mathbf{B}/\mu) \tag{3}$$

$$\nabla\cdot\mathbf{B} = 0 \tag{4}$$

$$\nabla\cdot\varepsilon\mathbf{E} = q \tag{5}$$

where

$$q = \sum_\alpha q_\alpha \int f_\alpha d\mathbf{u} \tag{6}$$

and

$$\mathbf{J} = \sum_\alpha q_\alpha \int \mathbf{u} f_\alpha d\mathbf{u}. \tag{7}$$

We see that the Boltzmann equation is connected to Maxwell's equations through the charge density, q, and the current density, $\mathbf{J}$. This coupled set of equations is termed a Vlasov-Boltzmann model. The equations can also be written in a divergence form

$$\frac{\partial f_\alpha}{\partial t} + \nabla_{\mathbf{x}}\cdot(\mathbf{u} f_\alpha) + \nabla_{\mathbf{u}}\cdot(\frac{q_\alpha}{m_\alpha}(\mathbf{E}+\mathbf{u}\times\mathbf{B}) f_\alpha) = \sum_\beta C_{\alpha\beta}(f_\alpha, f_\beta) \tag{8}$$

However, for simplicity in this report we discuss only a Boltzmann model associated with a force-free, single species reduced equation in one spatial dimension and one velocity dimension

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = C(f) \tag{9}$$

as this equation exemplifies the fundamental issues of concern. The left hand side says that the distribution function, f, is transported on $dx/dt = u$ characteristics. This transport is modified by the collision source term, $C(f)$. We will describe how a specific representation for f behaves across a range of response conditions where the effect of C is dominant to very small. We chose the simple Bhatnagar-Gross-Krook (BGK) operator,

$$C = -\frac{1}{\tau}(f - f_M) \tag{10}$$

where τ is the mean time between collisions and $1/\tau$ is the collision frequency [6]. We are interested in the numerical behavior of this equation for arbitrary $\tau > 0$. Here

$$f_M = \frac{n}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(u-\mu)^2/\sigma^2} = n\mathscr{N}(u;\mu,\sigma^2) \tag{11}$$

where f_M is the Maxwellian that matches the mass, momentum and energy moments of f, and $\mathscr{N}(u;\mu,\sigma^2)$ is the unit normal distribution with mean, μ, and variance, σ^2. In particular,

$$n = \langle f \rangle \tag{12}$$

$$\mu = \langle uf \rangle / n \tag{13}$$

$$\sigma^2 = \frac{\langle (u-\mu)^2 f \rangle}{n} \tag{14}$$

where $\langle \cdot \rangle$ denotes integration over velocity space. One can then show that

$$\langle C(f) \rangle = 0 \tag{15}$$

$$\langle uC(f) = 0 \tag{16}$$

$$\langle (u-\mu)^2 C(f) \rangle = 0 \tag{17}$$

and thus the collision term does not modify the first three moments.

If we non-dimensionalize equation 9 using a velocity scale, U, and a length scale, L, a non-dimensional $\hat{\tau} = \tau U/L$ appears inversely in front of the collision term. This is the Knudsen number

$$K_n = \hat{\tau} = \frac{\tau U}{L} = \frac{l}{L} \tag{18}$$

where $l = \tau U$ is the mean free path.

The last step is to choose a representation for the distribution function f. In the course of this project we explored a number of different representations, each with its own set of strengths and weaknesses. In what follows, we describe some of these representations and present examples of their behavior.

2.1. Asymmetric Hermite Expansion

The first representation explored was based on the asymmetric Hermite polynomial basis functions introduced in [16]. This basis has been applied successfully, e.g. [30, 12], to modeling of plasma dynamics. In a $(1x, 1u)$ domain, the distribution can be approximated as

$$\bar{f}(t,x,u) = \frac{1}{a} \sum_{j=0}^{N} c_j(t,x) \phi_j(s) \tag{19}$$

where w is a velocity shift, analogous to the fluid velocity, a is a velocity scale, analogous to the thermal speed, and the dimensionless velocity

$$s = \frac{u-w}{a}. \tag{20}$$

Here, one has a choice to either make (w,a) constant, piecewise constant over say a zone or time-step, or to allow it to vary locally.

If one chooses (w,a) as constants, then as shown in [16], substituting the expansion above into the Boltzmann equation one finds a linear system that can be diagonalized with real, distinct eigenvalues given by the zeros of Hermite polynomial $H_{N+1}(s)$. With this in hand, a variety of numerical methods can be readily constructed including high order, upwind Finite Volume (FV) or Discontinuous Galerkin (DG) methods with exact conservation properties.

Choosing (w,a) as piecewise constant one can still make use of many of these properties, develop exactly conservative, high order FV or DG methods, and one might expect the expansion to better represent the distribution locally, while adding the expense of additional projection operations relating the basis in one domain to that in another. This is the approach we sought to explore.

The final details to consider pertain to the ability of this representation to match a known distribution function. Qualitatively speaking, given a typical distribution function, what polynomial degree N should one expect to keep to match it with an Hermite expansion to say eye norm? To address this question, we consider a known, distribution function of two well separated Gaussians, characteristic of an asymmetric two-stream configuration.

$$f = \frac{1}{b}\phi_0(r_0) + \frac{2}{b}\phi_0(r_1) \tag{21}$$

where

$$r_0 = \frac{u+1}{b} \tag{22}$$

$$r_1 = \frac{u-1}{b} \tag{23}$$

$$b = 0.25 \tag{24}$$

and expand this in an asymmetric Hermite basis in the form of equation 19 where (w,a) are chosen to be the fluid velocity and thermal speed such that the lowest order term is the equilibrium Maxwell-Boltzmann. Truncating the expansion at degree N, the approximate distribution function $\bar{f}$ is plotted in figure 2-1.

There are a number of observations one can readily make from this figure. Most importantly, the distribution function is slow to converge, it converges in an oscillatory fashion with significant excursions to negative values and appears (in the eye norm) to match the input distribution at $N = 256$. The slow convergence of Hermite expansions is well documented in the literature and one suggestion [34] to improve this is to scale the velocity scale a by $\sqrt{N}$. Qualitatively, this has the behavior of keeping the zeros, and hence the high frequency behavior, of the Hermite polynomials within a range of velocities approximately given by the thermal velocity. Practically speaking, choosing the scaling of a is a delicate matter. Choose it too large and little improvement is observed. Choose it too small and the solution becomes increasingly oscillatory. In fact, scanning a range of velocity scale for this problem and one finds that the L2 error is oscillatory with multiple minima making an algorithmic search for the optimal decomposition numerically challenging.

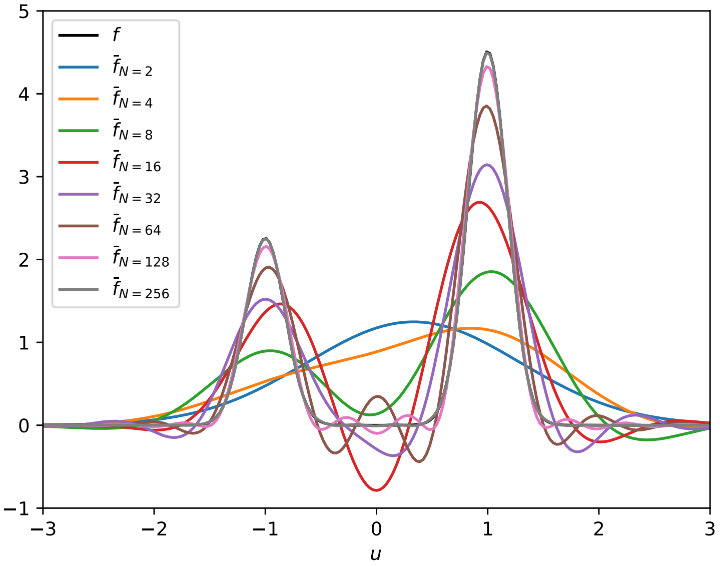

Figure 2-1 The known distribution function of two well separated, asymmetric Gaussians is approximated by an expansion in asymmetric Hermite polynomial basis functions and truncated at degree N.

2.2. Gaussian Positive Polynomial Expansion

A second representation explored in this project is one that we will refer to as a Gaussian Positive Polynomial (GPP) expansion. The goals here are to construct a representation that is essentially of the same form as an asymmetric Hermite expansion, i.e. a Gaussian times a polynomial, that is positive definite and has an optimal value for the velocity shift and scale parameters (w,a) introduced in §2.1. That is, we seek to approximate the distribution function as

$$\bar{f} = \frac{q}{a}\phi_0(s)P(s) \tag{25}$$

where q is an scale coefficient and $P(s)$ is a univariate polynomial of degree N.

A univariate polynomial $p(x) \in \mathbb{R}[x]$ of degree n has the form

$$p(x) = p_n x^n + p_{n-1}x^{n-1} + \cdots + p_1 x + p_0, \tag{26}$$

where the coefficients p_k are real. From the Fundamental Theorem of Algebra, every nonzero univariate polynomial of degree n has exactly n complex roots and as a result has the unique factorization

$$p(x) = p_n \prod_{k=1}^{n}(x - x_k), \tag{27}$$

where $x_k \in \mathbb{C}$ are the roots of $p(x)$. If all p_k are real and x_k is a root, then the complex conjugate x_k^* is also a root, so that complex roots appear in pairs. As a result, it follows that a univariate polynomial is non-negative if and only if it has the form

$$p(x) = p_n \prod_{j=1}^{n_j} (x - r_j)^2 \prod_{k=1}^{n_k} (x - a_k + ib_k)(x - a_k - ib_k)\,, \tag{28}$$

where r_j and $a_k \pm ib_k$ are real and complex roots respectively. This observation forms the basis of the GPP representation.

In order to capture the form of a general positive definite polynomial, yet not assume that the highest order coefficient is non-zero, we choose

$$P(s) = \prod_{i=0}^{N-1} \left[x_i^2 (s - y_i)^2 + z_i^2\right] \tag{29}$$

with real (x_i, y_i, z_i) for $P(s)$ in equation 25. To determine the coefficients (q, w, a, x_i, y_i, z_i), we follow the general procedure of minimizing the Kullback-Leibler (KL) divergence, as discussed in §3, subject to a number density conservation constraint. The resulting system of equations is

$$\int \left(\frac{f \bar{f}_{,j}}{\bar{f}}\right) s^n du = \int \bar{f}_{,j} s^n du \ \text{ for } \ n = (0, 1, 2) \tag{30}$$

where

$$\bar{f}_{,j} = \frac{\bar{f}}{x_j^2 (s - y_j)^2 + z_j^2} \tag{31}$$

and

$$\int f u^n du = \int \bar{f} u^n du \ \text{ for } \ n = (0, 1, 2). \tag{32}$$

Hence, the KL divergence minimizing solution $\bar{f}$ exactly conserves the first three moments of the distribution function f.

A numerical method to solve this system can be constructed. It is, unfortunately, not as simple and robust as a fixed-point method, yet a Newton method can be constructed. As an example of the behavior of this approximation, consider the test problem introduced in §2.1 of two well separated Gaussians, characteristic of an asymmetric two-stream configuration.

The KL divergence minimizing GPP approximation to this distribution is shown in figure 2-2 for a range of N, polynomial degree $2N$. By design, the GPP approximation is positive definite and converges to the known distribution function with increasing N. Note also that to the eye norm, this approximation recovers the input distribution at $N = 8$, a polynomial of degree 16. Comparing this to the asymmetric Hermite expansion presented in §2.1 one gets a feel for the relative impact of an optimal choice of the velocity shift and scale parameters (w, a) which are generated as part of the KL divergence minimizing GPP solution.

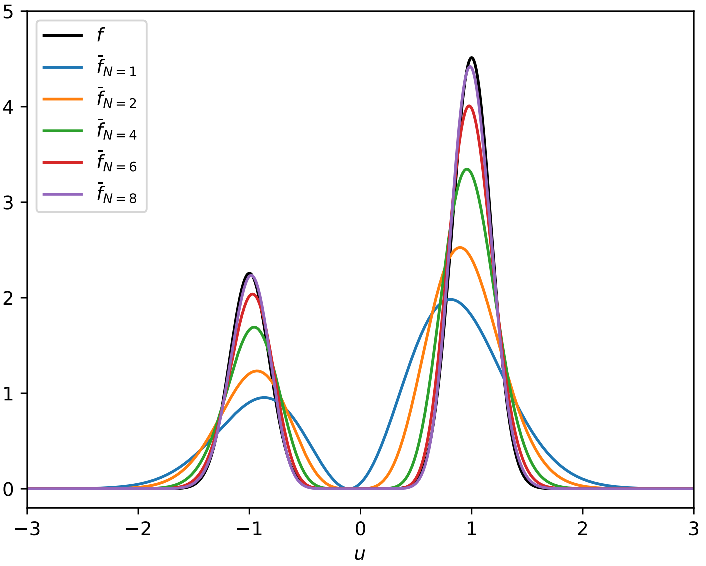

Figure 2-2 The known distribution function of two well separated, asymmetric Gaussians is approximated by a GPP for a range of N.

2.3. Gauss Mixture Model

The third, and most successful, representation explored is a Gauss Mixture Model (GMM). We propose to represent the distribution function f at each $\mathbf{x}$ position by a sum of M Gaussians, with positive weights $\pi_k(\mathbf{x})$. That is,

$$f \approx \bar{f}(u;\pi(x,t),\mu(x,t),\sigma(x,t)) = \sum_{k=1}^{M} \pi_k(x,t)\mathscr{N}(u;\mu_k(x,t),\sigma_k^2(x,t)) \tag{33}$$

where $\mathscr{N}$ is a unit normal distribution with mean μ_k and variance σ_k^2. The π_k "mixing" coefficients are required to sum to the number density, $n(x,t)$,

$$\langle \bar{f}(u;\pi(x,t),\mu(x,t),\sigma(x,t))\rangle = \sum_{k=1}^{M} \pi_k(x,t)\langle \mathscr{N}(u;\mu_k(x,t),\sigma_k^2(x,t))\rangle = \sum_{k=1}^{M} \pi_k(x,t) = n(x,t). \tag{34}$$

If $K_n \ll 1$ then collisions will dominate and f should be well approximated by a small number, M, of Gaussians. If $K_n \gg 1$ then collisions will be very infrequent and an arbitrary initial distribution function will tend to become highly complicated in phase space. We expect that at some point in the evolution of such a collisionless distribution function that any compressed representation will degrade. In this case any potential advantage of a compressed representation will then be lost. We intend to examine these questions in this report. For simplicity of notation we generally choose to drop explicit parametric dependence on x and t. This will be implicitly assumed in the mixture coefficients as well as each Gaussian mean and variance.

3. THE GMM REPRESENTATION, EXPECTATION MAXIMIZATION ALGORITHMS AND SOLUTIONS

We propose to represent the distribution function at each spatial position by a GMM along with constraints on the mixture coefficients to ensure mass conservation. The means and variances of each Gaussian are to be varied and chosen to match the actual distribution as closely as possible. The GMM is used in many fields of data science and in particular for cluster analysis. The standard popular algorithm for accomplishing this analysis is called Expectation Maximization (EM) and computes the mixture weights, means and distribution variances from statistical samples [21]. The Gaussian mixture concept has recently been used in the plasma physics context to compute a compressed representation of a particle-in-cell defined distribution function for the purposes of restarting very large computations [10]. This work demonstrates that such a compressed representation can be quite effective. In our work we examine the possibility that the GMM representation itself may be used as a basic internal representation and attempt to understand the ramifications of such an approach. We explore the Expectation Maximization (EM) algorithm in order to accomplish this goal. Instead of discussing the algorithm in terms of a sample set as is usually done, we will present the algorithm in terms of an assumed known distribution function, f. This is the “population” limit in the data science literature. We are not primarily concerned with statistical accuracy (sample complexity) but with the computational complexity of finding the best estimator for a given distribution.

We start with the need to approximate a distribution function with a sum of Gaussians. To do this we need some idea of how to "discriminate" between two different distributions. A standard way to do this is with the Kullback-Leibler (KL) divergence

$$D_{KL}(f||\bar{f}) = \langle \log(f/\bar{f}) f/n \rangle \tag{35}$$

Here f is the known distribution and $\bar{f}$ is the GMM approximation to f. The KL divergence always has a zero lower bound. This is seen using the probabilistic version of Jensen’s inequality for the concave down function

$$\log(\langle Y f/n \rangle) \geq \langle \log(Y) f/n \rangle \tag{36}$$

or

$$-\log(\langle Y f/n \rangle) \leq -\langle \log(Y) f/n \rangle \tag{37}$$

and thus

$$D_{KL}(f||\bar{f}) = \langle \log(f/\bar{f}) f/n \rangle = -\langle \log(\bar{f}/f) f/n \rangle \geq -\log(\langle (\bar{f}/f) f/n \rangle) = -\log(n/n) = 0. \tag{38}$$

Our desired approximation minimizes the KL divergence with respect to the vector of parameters $\Theta = (\pi, \mu, \sigma)$ and a Lagrange multiplier λ. We need to solve

$$\arg\min_{\Theta,\lambda} \left(D_{KL}(f||\bar{f}) + \lambda (\sum_{k=1}^{M} \pi_k - n) \right) \tag{39}$$

Necessary conditions for a minimizer can be found by differentiating with respect to Θ. Local minima should satisfy

$$\pi_k = \langle \gamma_k f \rangle \tag{40}$$

$$\mu_k = \langle u\gamma_k \rangle / \pi_k \tag{41}$$

$$\sigma_k^2 = \langle (u-\mu_k)^2 \gamma_k \rangle / \pi_k \tag{42}$$

$$\lambda = 1 \tag{43}$$

where

$$\gamma_k(u) = \frac{\pi_k \mathcal{N}(u; \mu_k, \sigma_k^2)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(u; \mu_j, \sigma_j^2)} \tag{44}$$

The Expectation Maximization algorithm is a fixed point iterative approach to find a solution to this system of equations:

- Step 0: Set M, ε, and an initial guess for the set Θ of parameters.

- Step 1: The first (Expectation) part of the EM algorithm computes an approximation, $\gamma_k(u)$, to the probability of a point in the sample space arising from the k-th Gaussian in the mixture.

$$\gamma_k(u) = \frac{\pi_k \mathcal{N}(u; \mu_k, \sigma_k^2)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(u; \mu_j, \sigma_j^2)} \tag{45}$$

- Step 2: The second (Maximization) part of the algorithm computes a new estimate for the parameters in the model based on the proposed γ_k functions. Thus

$$\hat{\pi}_k = \int \gamma_k(u) f(u) du \tag{46}$$

$$\hat{\mu}_k = \frac{\int \gamma_k(u) u f(u) du}{\int \gamma_k(u) f(u) du} = \frac{\int \gamma_k(u) u f(u) du}{\hat{\pi}_k} \tag{47}$$

$$\hat{\sigma}_k^2 = \frac{\int \gamma_k(u)(u-\hat{\mu}_k)^2 f(u) du}{\int \gamma_k(u) f(u) du} = \frac{\int \gamma_k(u)(u-\hat{\mu}_k)^2 f(u) du}{\hat{\pi}_k} \tag{48}$$

- Step 3: If $|D_{KL}| < \varepsilon$ or $|\delta D_{KL}| < \varepsilon$ or the maximum number of iterations is reached then Terminate otherwise Step 4.

- Step 4: Set π_k, μ_k, σ_k^2 to $\hat{\pi}_k, \hat{\mu}_k, \hat{\sigma}_k^2$ respectively and go to Step 1.

We also require a minimum number of iterations (default one (1) is used for the results in Section 3.4). The $|\delta D_{KL}| < \varepsilon$ check avoids additional iterations when no progress is being made. The distributional information in f comes from integral computations in Step 2. This algorithm has a significant advantage in that each iterate results in a state which exactly matches the first three moments of the distribution function as computed by the chosen numerical quadrature rule.

$$\sum_{k=1}^{M} \hat{\pi}_k = \int \sum_{k=1}^{M} \gamma_k f du = n \tag{49}$$

$$\sum_{k=1}^{M} \hat{\pi}_k \hat{\mu}_k = \int \sum_{k=1}^{M} \gamma_k u f du = \int u f du = \mu \tag{50}$$

$$\sum_{k=1}^{M} \hat{\pi}_k (\hat{\sigma}_k^2 + \hat{\mu}_k^2) = \int \sum_{k=1}^{M} \gamma_k u^2 f du = \int u^2 f du = \sigma^2 + \mu^2 \tag{51}$$

shows that the GMM retains the moments of the input distribution f. The moments are defined by the specific numerical quadrature scheme chosen for the integrals computed in Step 2. One must integrate over a sufficiently large domain in order to obtain robust results. This algorithm will eventually find a parameter that is a local minimum. There is no guarantee that the algorithm will find a global minimum. We can, however, track the quality of the approximation using the KL divergence.

3.1. Initial Value Algorithm

The standard EM algorithm is a simple, robust fixed point iteration. Although the literature contains numerous efforts to find ways to speed up the EM algorithm with limited success, simple, fast, robust, globally-minimizing algorithms remain elusive. The EM algorithm is well known to suffer from very slow rates of convergence especially for closely overlapping distributions. As an example, we found that using an initial starting mixture such that each Gaussian contains $1/M$ of the initial number density and equal velocities and variances set to the global mean and variance respectively – resulted in a complete failure. Recent theoretical work has attempted to understand the details of the observed rates of stochastic and algorithmic convergence [13, 3, 24]. In our case, we are interested in algorithmic convergence since we assume that we know the distribution function at each x position and desire to efficiently find a good parametrized representation of the function with a relatively small number of parameters. Generally one might assume that the most efficient results will depend on the closeness of the initial guess for the parameters and that an approach to solve the time stepping Boltzmann equation that uses a previously computed set of parameters will be the best starting point assumption. This is also incorrect. The EM algorithm is not efficient or effective in discovering the relatively wide ranging new changes to the distribution function generated by the Boltzmann equation when starting from a compact set of initial Gaussian states.

The best initialization algorithm we have found so far sets the first Gaussian of M Gaussians to a position and width associated with the actual mean and width of the distribution function. The remaining $M-1$ Gaussians then receive initial mixture positions and widths that are equidistributed across velocity space. The initial mixture coefficient values are set to the number density divided by the number of Gaussians, M. In particular, for the first Gaussian we set

$$\pi_0 = n/M \tag{52}$$

$$\mu_0 = \mu \tag{53}$$

$$\sigma_0 = \sigma \tag{54}$$

and for the remaining $M-1$ Gaussians ($k = 1, \cdots, M-1$) we set

$$\pi_k = n/M \tag{55}$$

$$\mu_k = \frac{u_{max} - u_{min}}{M-1}(k-1) + (u_{min} + \frac{u_{max} - u_{min}}{2(M-1)}) \tag{56}$$

$$\sigma_k = (u_{max} - u_{min})/(M-1) \tag{57}$$

This initial guess for the EM algorithm has been found to be quite beneficial and is significantly better at finding a quality solution than using the "last parameter set" as an initial condition for the EM algorithm. This is illustrated in the following pictures which show a distribution function plot of a Boltzmann equation simulation for a nearly collisionless case. Comparing Figures 3-1 and 3-2 we see that this initialization methodology (Figure 3-2) demonstrates significantly greater uniformity.

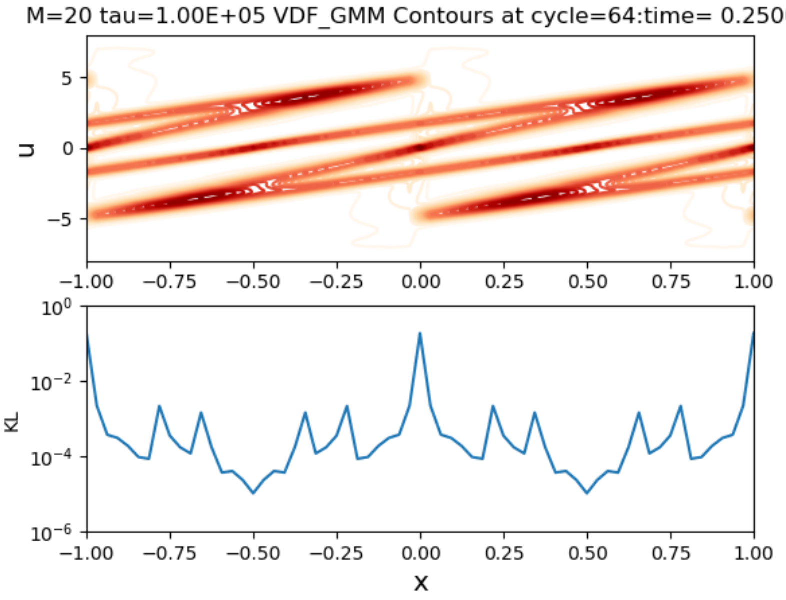

Figure 3-1 Solution for $\tau = 10^5$ and cycle 64 for $M = 20$ case using the "last parameter set" to start the EM algorithm for each compression step.

In general, the best solution to be found within given computational constraints depends heavily on the initial values of the parameters. As such, many different algorithms have been proposed and compared and sophisticated EM initialization strategies might indeed provide some additional benefit [7, 31]. Although the initial guess appears to be a key choice to develop an

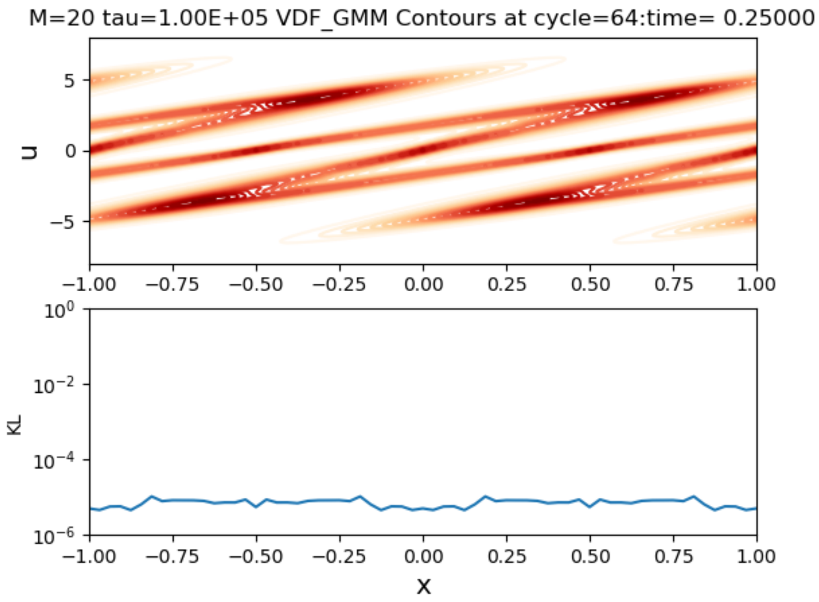

Figure 3-2 Solution for $\tau = 10^5$ and cycle 64 for $M = 20$ case using the initial mixture rule of Equations 54 and 57 to start the EM algorithm for each compression step.

effective Boltzmann equation algorithm, it is also of great importance to find ways to accelerate the algorithmic convergence of the EM algorithm as the algorithm tends to slow down as it gets closer to a viable solution. This will be examined next.

3.2. Acceleration Algorithms

A foundational requirement of our approach is the need to robustly represent the distribution functions uniformly at a specified level as measured by the KL divergence. It is clear that efficiently finding these quality representations as the distribution function evolves is a key issue. The literature on EM related algorithms is vast due to the importance of this model in the data analysis community. We have implemented several algorithms in our investigation of the GMM representation that showed promise to provide utility for our purposes.

We have implemented two algorithms proposed by Varadhan and Roland [35]. The first is a Steffensen type method for EM (STEM or S3) which is based on computing an estimate of the derivative from two sequential iterates. The second is a particular "squared" iterative method (SQUAREM or SqS3) which provides a new estimate that is quadratic in the derivative estimate. Both methods need limiting to achieve robustness. One possible algorithm implementation in the notation of Varadhan and Roland is as follows:

1. Θ_1=EM(Θ_0)
2. Θ_2=EM(Θ_1)

```
3. $r_0 = \Theta_1 - \Theta_0$
4. $r_1 = \Theta_2 - \Theta_1$
5. $v_0 = r_1 - r_0$
6. $\alpha = \min(-\|r_0\|/\|v_0\|, -1)$
7. S3: $\Theta = \Theta_0 - \alpha r_0$
8. SqS3: $\Theta = \Theta_0 - 2\alpha r_0 + \alpha^2 v_0$
9. $\Theta$=EM($\Theta$);
10. Check that all results are finite floating point numbers. If not then set $\alpha = (-1+\alpha)/2.$ and go to 7.
11. Feasible Parameter Limiting: if any $\pi_k < 0$ or any $\sigma_k < 0$, then set $\alpha = (-1+\alpha)/2.$ and go to 7.
12. Compute KL divergence
13. if $|D_{KL}| < \varepsilon$ or $|\Delta D_{KL}| < \varepsilon$ then stop else $\Theta_0 = \Theta$ go to 1.
```

A better implementation that avoids roundoff issues in the extrapolated parameters is to use

```
1. $\Theta_1$=EM($\Theta_0$)
2. $\Theta_2$=EM($\Theta_1$)
3. $r_0 = \Theta_1 - \Theta_0$
4. $r_1 = \Theta_2 - \Theta_1$
5. $v_0 = r_1 - r_0$
6. $\alpha = \min(-\|r_0\|/\|v_0\|, -1)$
7. $\beta = -1 - \alpha$
8. S3: $\Theta = \Theta_1 + \beta r_0$
9. SqS3: $\Theta = \Theta_2 + 2\beta r_1 + \beta^2 v_0$
10. $\Theta$=EM($\Theta$);
11. Check that all results are finite floating point numbers. If not then set $\beta = \beta/2$ and go to 8.
12. Feasible Parameter Limiting: if any $\pi_k < 0$ or any $\sigma_k < 0$, then set $\beta = \beta/2$ and go to 8.
13. Compute KL divergence
14. if $|D_{KL}| < \varepsilon$ or $|\Delta D_{KL}| < \varepsilon$ then stop else $\Theta_0 = \Theta$ go to 1.
```

We also implemented a "vector ε" algorithm that has been suggested as a useful approach for improving convergence of the EM algorithm [19, 36, 20, 8, 14]. Our algorithm takes two EM iterations and then uses the three available iterate vector values to compute a new estimate with a single acceleration step. The accelerated values are only accepted if all of the estimated mixture fractions are positive and all estimated standard deviations are positive. The vector inverse is defined by

$$[\Theta]^{-1} = \frac{\Theta}{||\Theta||^2} \tag{58}$$

and we have

1. Θ_1=EM(Θ_0)
2. Θ_2=EM(Θ_1)
3. $r_0 = \Theta_1 - \Theta_0$
4. $r_1 = \Theta_2 - \Theta_1$
5. EPSILON: $\Theta = \Theta_1 + [[r_1]^{-1} - [r_0]^{-1}]^{-1}$
6. Set Θ=EM(Θ)
7. Check results are finite floating point numbers with $\pi_k >= 0$ and $\sigma_k > 0$. If not, set $\Theta = \Theta_2$ and go to 8.
8. Compute KL divergence
9. if $|D_{KL}| < \varepsilon$ or $|\Delta D_{KL}| < \varepsilon$ then stop else $\Theta_0 = \Theta$ go to 1.

Other acceleration algorithms are certainly possible. For example, since the exact gradients of the KL divergence are available, a gradient line search algorithm is suggested [29]. Algorithms that include some sort of a global search might also be possible. The speed and efficacy of the algorithm is our primary concern with the need to balance a subtle interplay between initial parameter placement, the quality of the GMM representation achieved and the solution acceleration strategy.

As this project came to an end, we were excited to discover the existence of the parabolic EM algorithm proposed by Berlinet and Roland [4, 5]. They show geometrically how the squared algorithms of Varadhan and Roland and the "vector ε" are in fact related and demonstrate how these algorithms can be extended to include a line search using information found via point evaluations of the KL divergence. This algorithm appears extremely promising due to its simplicity, efficiency and robustness. Numerical experiments using the line search extension claim approximately a 20 fold speedup. This magnitude of solution acceleration would be highly significant for our Boltzmann equation application.

3.3. Numerical Method

We will now discuss details of numerical methods associated with the Boltzmann equation solver. Considering the advection portion of Equation 9 as an operator $L(f)$, we can formally write the equation as

$$\frac{\partial f}{\partial t} = L(f) + C(f) \tag{59}$$

This equation has a hyperbolic part, L, and a potentially stiff source term, C. We need numerical methods that can handle this equation appropriately across the range of feasible parameters. Robust methods have been proposed in the Implicit-Explicit (IMEX) Runge-Kutta numerical integration literature [27, 23]. We have

$$f^{(i)} = f^n + \Delta t \sum_{j=1}^{i-1} \tilde{A}_{ij} L(f^{(j)}) + \Delta t \sum_{j=1}^{i} A_{ij} C(f^{(j)}) \tag{60}$$

$$f^{n+1} = f^n + \Delta t \sum_{i=1}^{s} \tilde{w}_i L(f^{(i)}) + \Delta t \sum_{i=1}^{s} w_i C(f^{(i)}) \tag{61}$$

for stages $i = 1,...,s$ where s is order of the scheme. For the scheme to be IMEX we must have $\tilde{A}_{ij} = 0$ for $j \geq i$. To ensure a simple implicit representation, one assumes a diagonally implicit Runge-Kutta schemes (DIRK) where $A_{ij} = 0$ for $j > i$. The coefficients of such schemes can be displayed in a double Butcher tableau:

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{w}^T \end{array} \qquad \begin{array}{c|c} c & A \\ \hline & w^T \end{array} \tag{62}$$

The $\tilde{c}$ and c matrices provide the Δt time factors evaluated in the respective L and C terms if they are non-autonomous.

The first order scheme is a simple operator split starting with an implicit backward Euler step:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \tag{63}$$

The second order scheme is an IMEX-SSP(2,2,2) L-stable scheme

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array} \tag{64}$$

where $\gamma = 1 - 1/\sqrt{2}$. All the coefficients can be shown to be positive. The time step, Δt, is chosen as some fraction of the fastest characteristic speed of the explicit hyperbolic operator.

The IMEX algorithm is shown in pseudocode in Algorithm 1:

```
for $i = 1,...,s$ do
    $\hat{f} \leftarrow f^n + \Delta t \sum_{j=1}^{i-1} \tilde{A}_{ij} L_j + \Delta t \sum_{j=1}^{i-1} A_{ij} C_j$
    Evaluate $\hat{f}$ moments to form $\hat{C}$
    Implicitly solve $f^{(i)} = \hat{f} + \Delta t A_{ii} \hat{C}(f^{(i)})$ for $f^{(i)}$
    $C_i \leftarrow (f^{(i)} - \hat{f})/(A_{ii}\Delta t)$
    $L_i \leftarrow L(f^{(i)})$
end
$f^{n+1} \leftarrow f^n + \Delta t \sum_{j=1}^{s} \tilde{w}_j L_j + \Delta t \sum_{j=1}^{s} w_j C_j$
```

Algorithm 1: IMEX Algorithm Pseudocode Summary

The implicit solve step for $f^{(i)}$ and the explicit L_i step are now described in more detail.

We first consider how the BGK operator interacts with the computation of the moments in the numerical scheme. We evaluate $\hat{n}, \hat{\mu}, \hat{\sigma}$ from $\hat{f}$ to form $f_{\hat{M}}$ and thus obtain the $\hat{C}$ operator. We then solve

$$f^{(i)} = \hat{f} + \Delta t A_{ii} \hat{C}(f^{(i)}) \tag{65}$$

or

$$f^{(i)} = \hat{f} - \Delta t A_{ii} \frac{1}{\tau}(f^{(i)} - f_{\hat{M}}) \tag{66}$$

and thus for each u

$$f^{(i)} = \frac{\hat{f} + \Delta t A_{ii} \frac{1}{\tau} f_{\hat{M}}}{1 + \Delta t A_{ii} \frac{1}{\tau}} \tag{67}$$

From this last equation we see immediately that the moments of $f^{(i)}$ must be equal to the moments of $\hat{f}$. The discrete implicit step is thus an invariant moment operator and there is no discrepancy or inconsistency introduced via the computation of the moments as a first step before the implicit solve.

The discrete hyperbolic advection operator algorithm, L, must now be given a specific form. We evaluate the distribution function at any required x position, solve for the distribution function changes via a numerical transport method and then reparameterize the GMM to match the new distribution. Below a subscript to the distribution function represents the x index position. We implemented the first and second order Lax-Wendroff algorithms with $p = 1$ and $p = 2$ respectively:

$$f_j^{n+1} = f_j^n - \frac{1}{2}(\Delta t u/\Delta x)(f_{j+1}^n - f_{j-1}^n) + \frac{1}{2}(\Delta t |u|/\Delta x)^p (f_{j+1}^n - 2f_j^n + f_{j-1}^n) \tag{68}$$

Although the $p = 2$ algorithm is high order for smooth solutions, it is not monotone and may contain significant dispersive wiggles for sharp discontinuities. The method can be improved by adding a non-linear flux limiter. First define the flux form of these equations

$$f_j^{n+1} = f_j^n - (F_{j+1/2} - F_{j-1/2}) \tag{69}$$

where

$$F^p_{j+1/2} = \frac{1}{2}\frac{\Delta t}{\Delta x}u(f^n_{j+1} + f^n_j) - \frac{1}{2}\left|\frac{\Delta t}{\Delta x}u\right|^p (f^n_{j+1} - f^n_j) \tag{70}$$

Combining the two to obtain a limited flux gives

$$F_{j+1/2} = F^1_{j+1/2} + \phi(r_{j+1/2})(F^2_{j+1/2} - F^1_{j+1/2}) \tag{71}$$

with the Van Leer flux limiter function

$$\phi(r) = \frac{2r}{1+r}, \qquad r \geq 0 \tag{72}$$

$$= 0, \qquad r < 0 \tag{73}$$

and the upwind slope ratio defined as

$$r_{j+1/2} = \frac{f_j - f_{j-1}}{f_{j+1} - f_j} \qquad (f_j - f_{j-1})(f_{j+1} - f_j) > 0 \tag{74}$$

$$= 0 \qquad (f_j - f_{j-1})(f_{j+1} - f_j) \leq 0 \tag{75}$$

for $a > 0$ and

$$r_{j+1/2} = \frac{f_{j+2} - f_{j+1}}{f_{j+1} - f_j} \qquad (f_{j+2} - f_{j+1})(f_{j+1} - f_j) > 0 \tag{76}$$

$$= 0 \qquad (f_{j+2} - f_{j+1})(f_{j+1} - f_j) \leq 0 \tag{77}$$

for $a < 0$. For numerical stability it is necessary that $\max(|u|)\frac{\Delta t}{\Delta x} \leq 1$.

Unfortunately, even this algorithm does not ensure positivity, and a second order method that maintains positivity is desirable. To satisfy these requirements we implemented the MPDATA (multidimensional positive definite advection transport algorithm) scheme by Smolarkiewicz [32, 33]. This algorithm starts out with a first order linear upwind flux followed by a non-linear antidiffusive flux update. First define the flux function

$$F^1_{j+1/2}(f_j, f_{j+1}, u) = ((u + |u|)f_i + (u - |u|)f_{j+1})\frac{\Delta t}{2\Delta x} \tag{78}$$

In the first step compute

$$f^*_j = f^n_j - (F^1_{j+1/2}(f_j, f_{j+1}, u_{j+1/2}) - F^1_{j-1/2}(f_{j-1}, f_j, u_{j-1/2})) \tag{79}$$

followed by the update

$$f^{n+1}_j = f^*_j - (F^1_{j+1/2}(f^*_i, f^*_{i+1}, \tilde{u}_{j+1/2}) - F^1_{j-1/2}(f^*_{i-1}, f^*_i, \tilde{u}_{j-1/2})) \tag{80}$$

where

$$\tilde{u}_{j+1/2} = \frac{(|u_{j+1/2}|\Delta x - \Delta t u^2_{j+1/2})(f^*_{j+1} - f^*_j)}{(f^*_{j+1} + f^*_j + \delta)\Delta x} \tag{81}$$

where $\delta = 10^{-15}$ is a very small floating point number added solely to avoid division by zero. In our case, $u_{j+1/2}$ does not actually depend on the spatial index, j, but clearly the antidiffusive velocity, $\tilde{u}_{j+1/2}$, does. If the Courant stability condition $|u|\Delta t/\Delta x \leq 1$ is satisfied then f^* is positive if f^n is. The use of $\tilde{u}$ in the second upwind step affects neither stability or positivity [32, Equation 18]. Thus the MPDATA scheme satisfies all our requirements for a high order advection scheme.

The MPDATA advection operator and the collision operator are both positive schemes individually. We would like to show that the combined IMEX scheme is also positive if L and C are positive when applied separately. For the first order ($s = 1$) IMEX case we have

$$\hat{f} = f^n \tag{82}$$

$$f^{(1)} = \hat{f} + \Delta t \hat{C}(f^{(1)}) \tag{83}$$

$$f^{n+1} = f^n + \Delta t L(f^{(1)}) + \Delta t \hat{C}(f^{(1)}) \tag{84}$$

$$= f^n + \Delta t L(f^{(1)}) + f^{(1)} - f^n \tag{85}$$

$$= f^{(1)} + \Delta t L(f^{(1)}) \tag{86}$$

which implies that the first order IMEX scheme is positive under these assumptions.

In the second order case, we have

$$\hat{f} = f^n \tag{87}$$

$$f^{(1)} = \hat{f} + \Delta t \gamma \hat{C}(f^{(1)}) \tag{88}$$

$$C_1 = (f^{(1)} - \hat{f})/(\Delta t \gamma) \tag{89}$$

$$\hat{f} = f^n + \Delta t L(f^{(1)}) + \Delta t(1 - 2\gamma)C_1 \tag{90}$$

$$f^{(2)} = \hat{f} + \Delta t \gamma \hat{C}(f^{(2)}) \tag{91}$$

$$C_2 = (f^{(2)} - \hat{f})/(\Delta t \gamma) \tag{92}$$

$$f^{n+1} = f^n + \frac{\Delta t}{2} L(f^{(1)}) + \frac{\Delta t}{2} L(f^{(2)}) + \frac{\Delta t}{2} C_1 + \frac{\Delta t}{2} C_2 \tag{93}$$

Even though the second order in time IMEX algorithm is L-stable, it is not at all clear from the algorithmic form that each of the substeps and final step will maintain positivity. We implemented a limiting approach such that if any $\hat{f}$ or f^{n+1} of the IMEX algorithm gives a negative distribution function value at Equation 90 or 93, the function value is limited to be greater than or equal to zero. This algorithm fails for the more collisional cases and it became clear that the standard IMEX second order L-stable method is insufficient for this purpose. The literature reveals that there exists second order positivity preserving IMEX methods requiring three stages in the double Butcher tableau and a correction step with an additional CFL constraint are available [17]. Due to their computational complexity and perceived relative importance to our main concerns, we have not attempted to implement such high order in time schemes. For the purposes of this report, we give results with the first order in time positivity preserving IMEX scheme while testing both first and second order positive advection operators.

Before running over the spatial grid, the previous states are all kept in a compressed representation. In the iteration over spatial position, the required stencil values are uncompressed to a velocity grid, the stencil operations described above are executed and the final result for that spatial grid point is then compressed and stored. The compression step involves first computing initial GMM values as described in Section 3.1 and then executing one of the EM algorithms described above in order to find a new GMM representation for the updated velocity distribution function.

3.4. Numerical Results

The quality of the simulation depends intrinsically on whether or not the GMM provides an effective representation of the distribution function at each spatial point. This is measured by the KL divergence that is computed every time the distribution function is compressed to a GMM representation. We can track the maximal divergence over all space-time points in order to get an idea of the related error for a given collisionality.

We choose a particular sine wave initial condition that demonstrates the tradeoff between collisionless solutions and the collisional solutions:

$$\mu = -U\sigma \sin kx \tag{94}$$

$$f = \mathscr{N}((u-\mu)/\sigma)/\sigma \tag{95}$$

where $L = 2, U = 4$ and $\sigma = 1$ and $k = 4\pi/L$.

For the results in this section we used a uniform velocity grid containing 801 points in the interval [-10,10] for the uncompression step, the trapezoidal rule for all quadratures and $\varepsilon = 10^{-6}$ for the stopping tolerance value.

3.4.1. Collisionless results

The distribution function plots in the 25 Figures 3-3 to 3-27 show the exact collisionless distribution function solution at any point in time, since this can be computed exactly by the method of characteristics. Below the distribution function, we plot the KL divergence value achieved using our choice of initial parameters and using the standard EM iterative algorithm. There are 5 sets of plots utilizing $M = 1, 5, 10, 15$ and 20 and each with 5 plots with linearly increasing time. The collisionless distribution folds over on itself many times. Careful examination of the plots will show that smaller values of M achieve only relatively large KL divergence values. Larger values of M allow for smaller KL divergence values that hold for a longer time. Note how the average KL divergence increases as time moves forward. For this problem, for large enough time we will find that any fixed number of Gaussian terms will eventually fail to provide a good representation of the exact distribution function because of the many additional peaks that are generated as time progresses. We emphasize that the KL divergence values shown in these collisionless plots are a snapshot of the error in the GMM representation of the known exact solution at the given time as plotted across all x-values. The fact that a GMM is capable of matching reasonably well such complicated distributions is very encouraging.

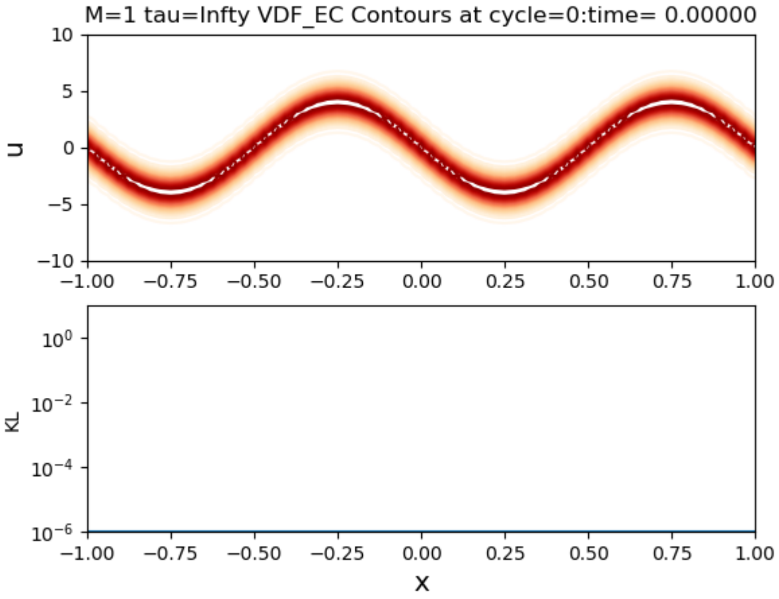

Figure 3-3 Exact collisionless solution for $M = 1$ at cycle 000.

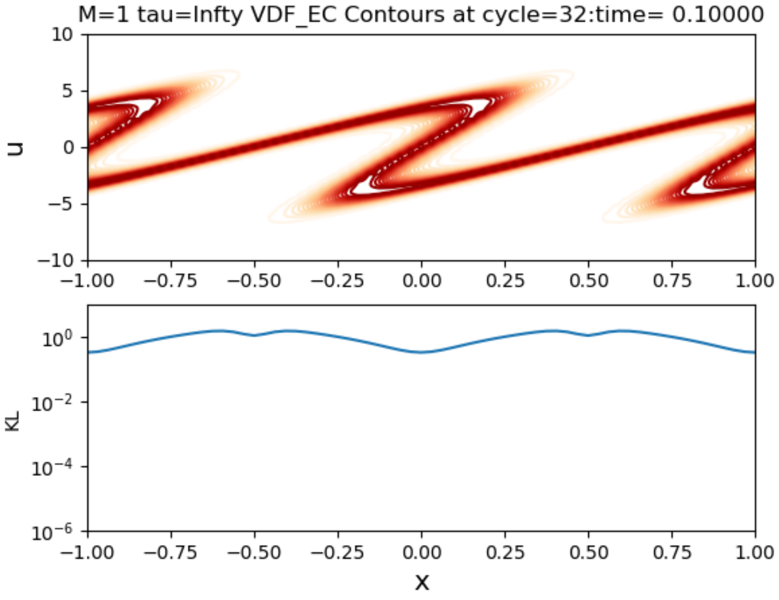

Figure 3-4 Exact collisionless solution for $M = 1$ at cycle 032.

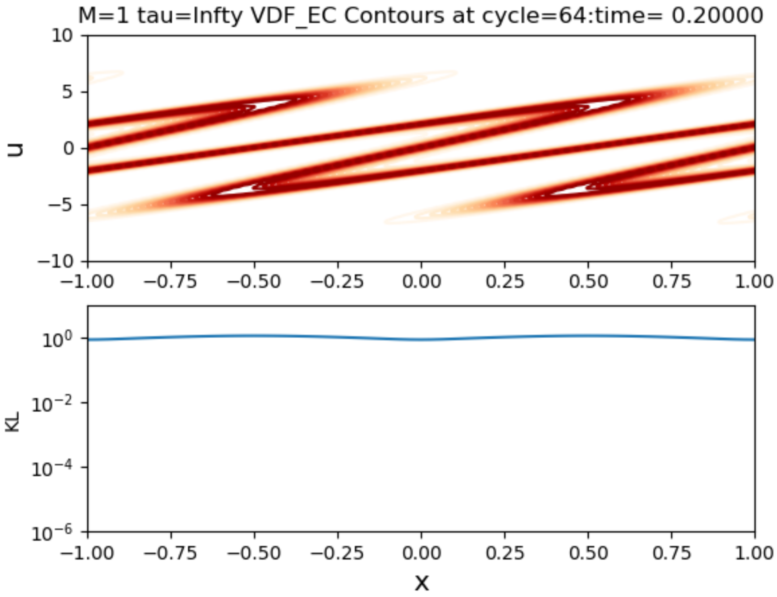

Figure 3-5 Exact collisionless solution for $M = 1$ at cycle 064.

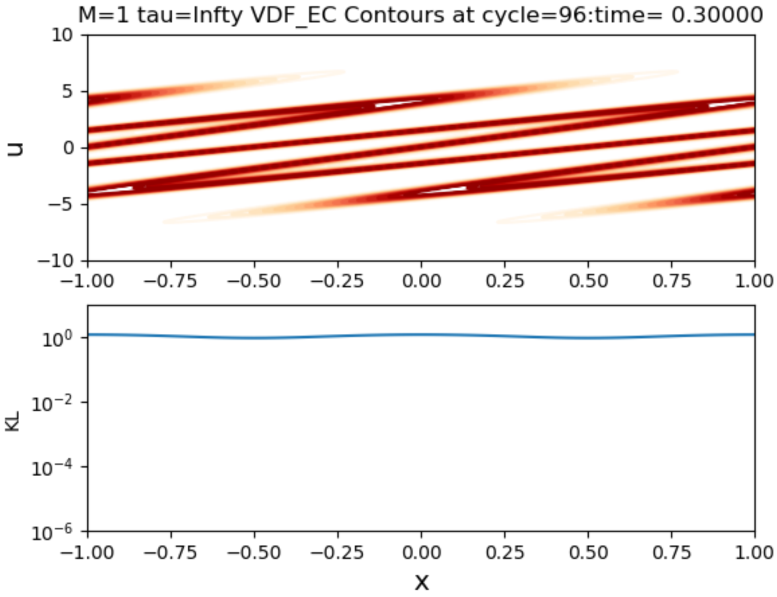

Figure 3-6 Exact collisionless solution for $M = 1$ at cycle 096.

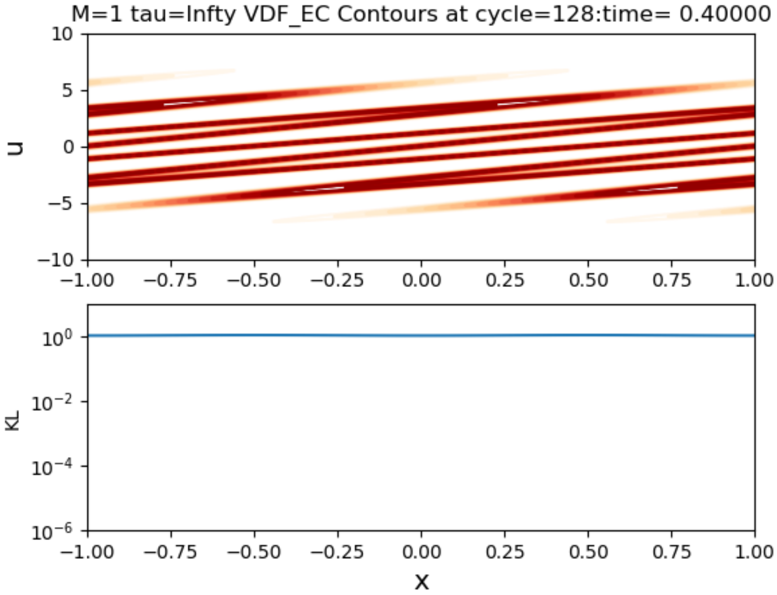

Figure 3-7 Exact collisionless solution for $M = 1$ at cycle 128.

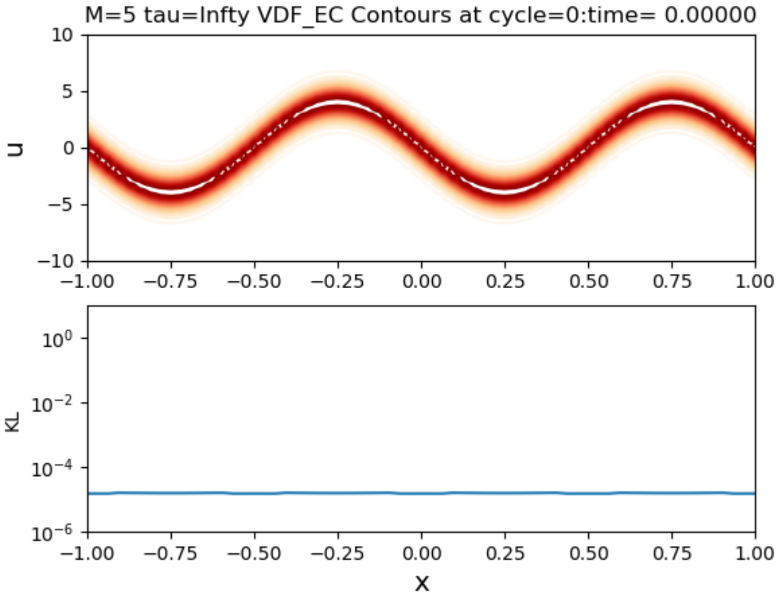

Figure 3-8 Exact collisionless solution for $M = 5$ at cycle 000.

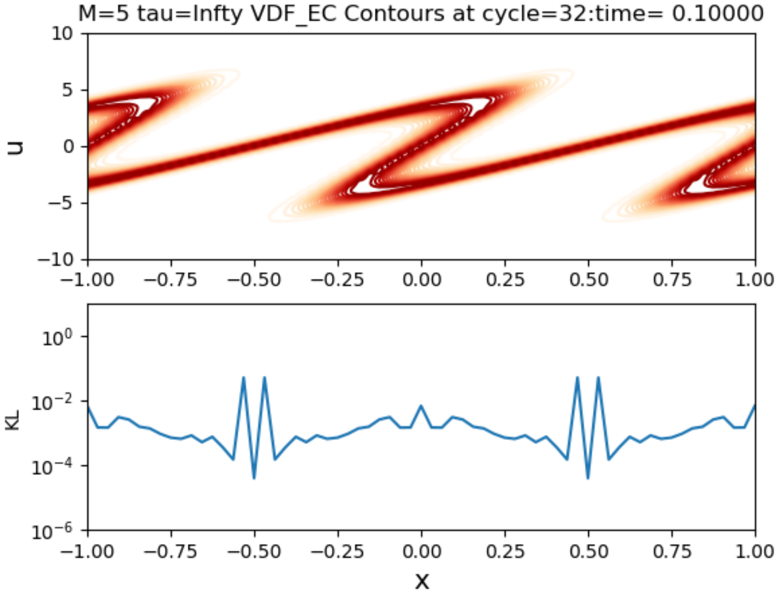

Figure 3-9 Exact collisionless solution for $M = 5$ at cycle 032.

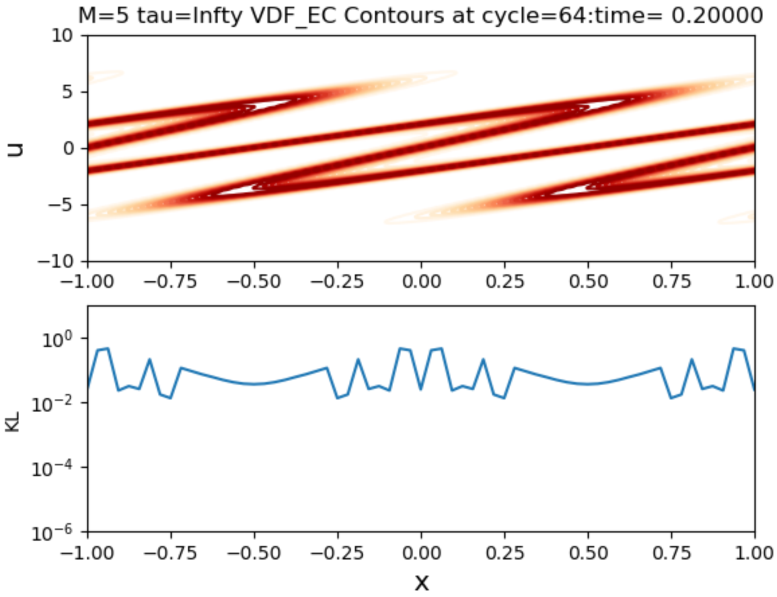

Figure 3-10 Exact collisionless solution for $M = 5$ at cycle 064.

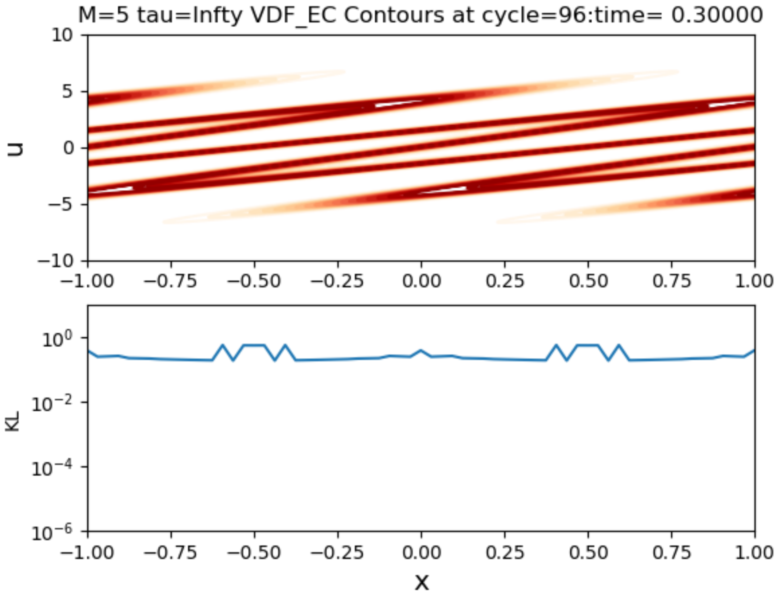

Figure 3-11 Exact collisionless solution for $M = 5$ at cycle 096.

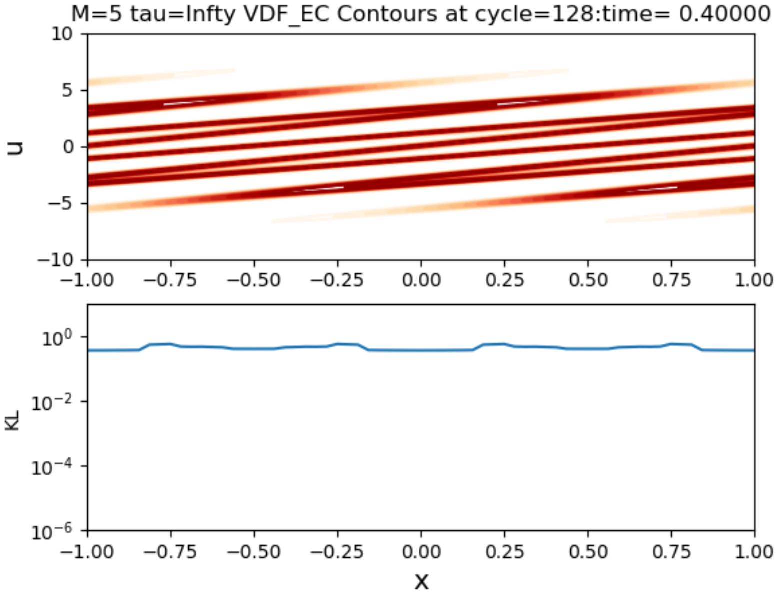

Figure 3-12 Exact collisionless solution for $M = 5$ at cycle 128.

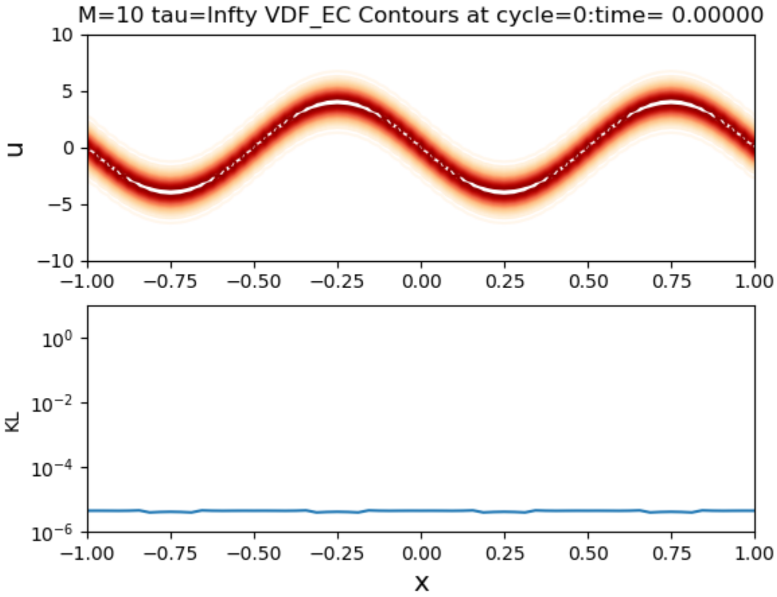

Figure 3-13 Exact collisionless solution for $M = 10$ **at cycle** 000**.**

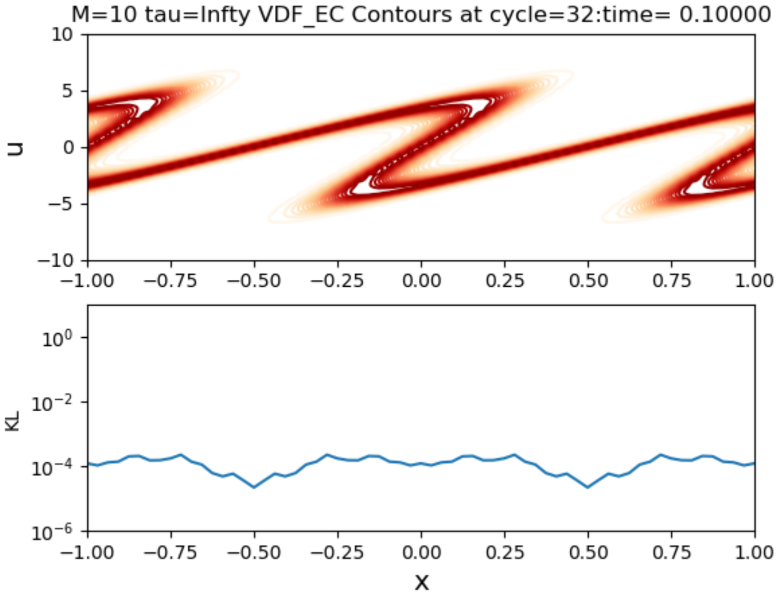

Figure 3-14 Exact collisionless solution for $M = 10$ **at cycle** 032**.**

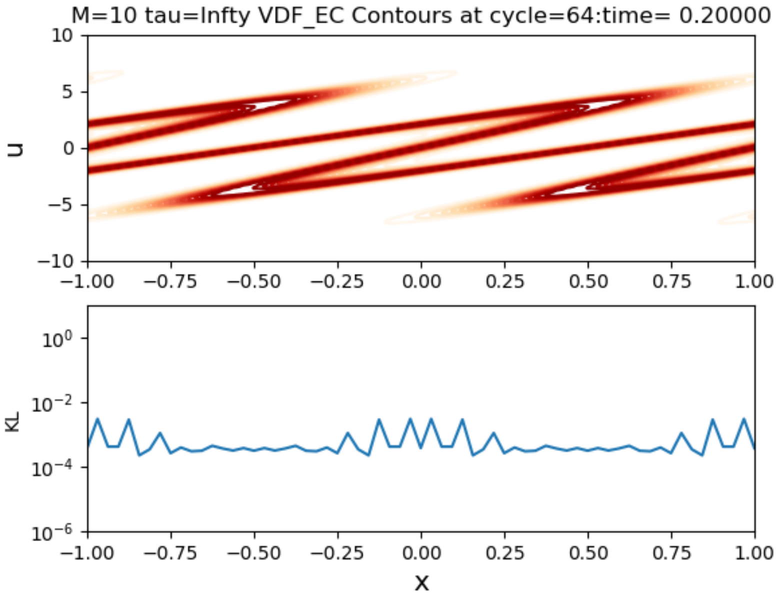

Figure 3-15 Exact collisionless solution for $M = 10$ at cycle 064.

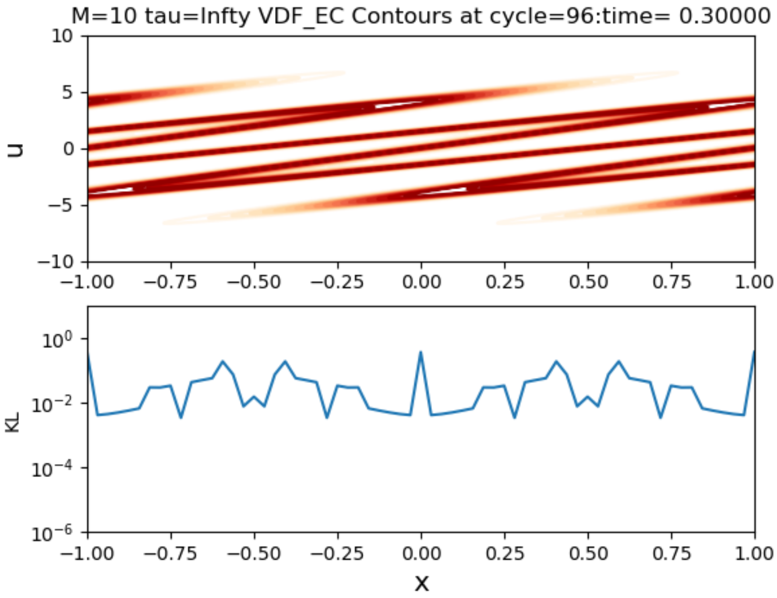

Figure 3-16 Exact collisionless solution for $M = 10$ at cycle 096.

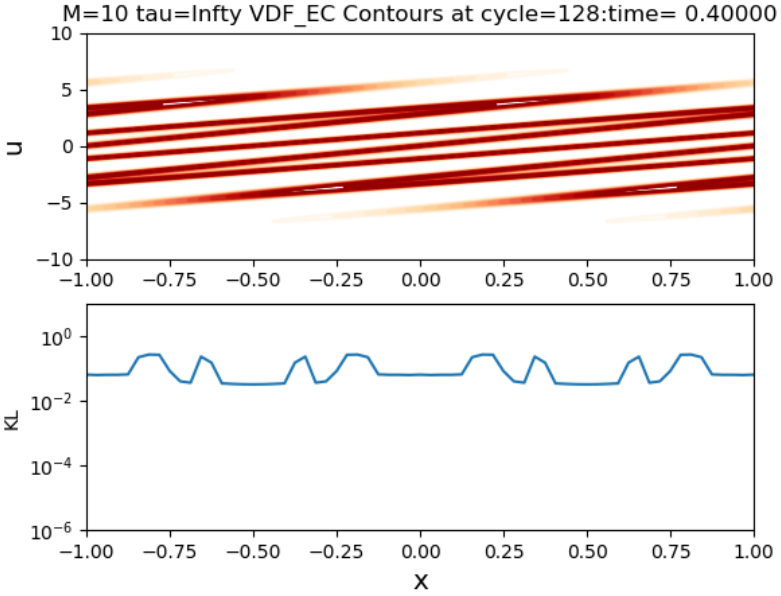

Figure 3-17 Exact collisionless solution for $M = 10$ at cycle 128.

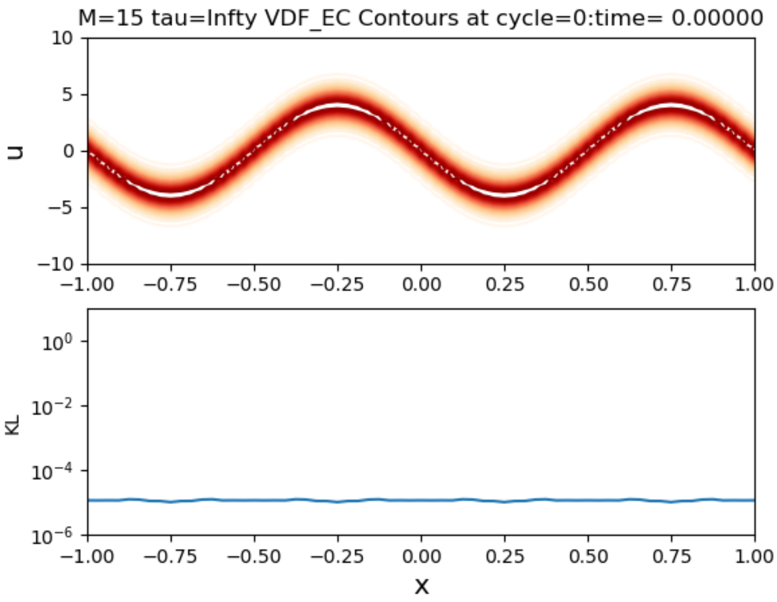

Figure 3-18 Exact collisionless solution for $M = 15$ at cycle 000.

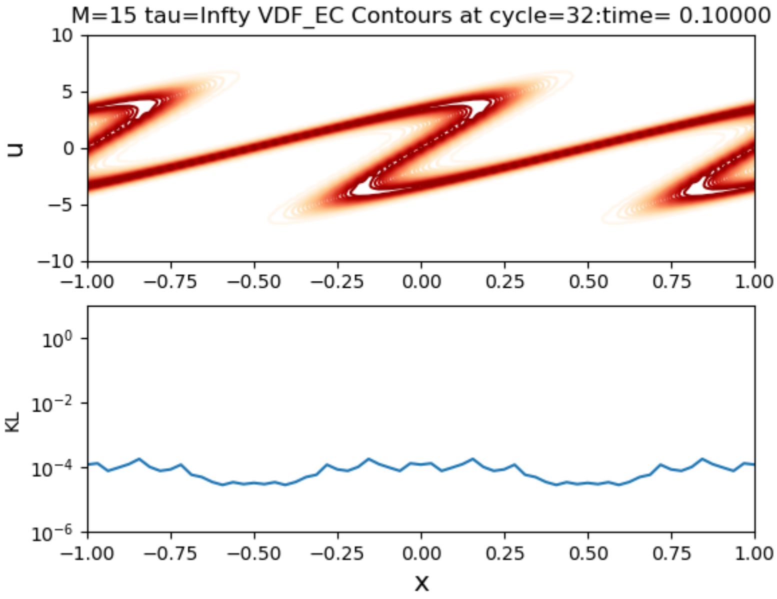

Figure 3-19 Exact collisionless solution for $M = 15$ at cycle 032.

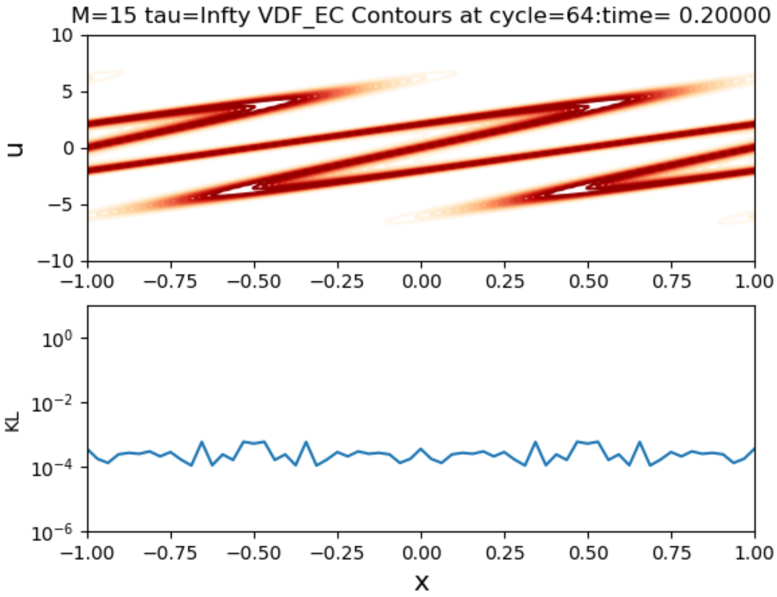

Figure 3-20 Exact collisionless solution for $M = 15$ at cycle 064.

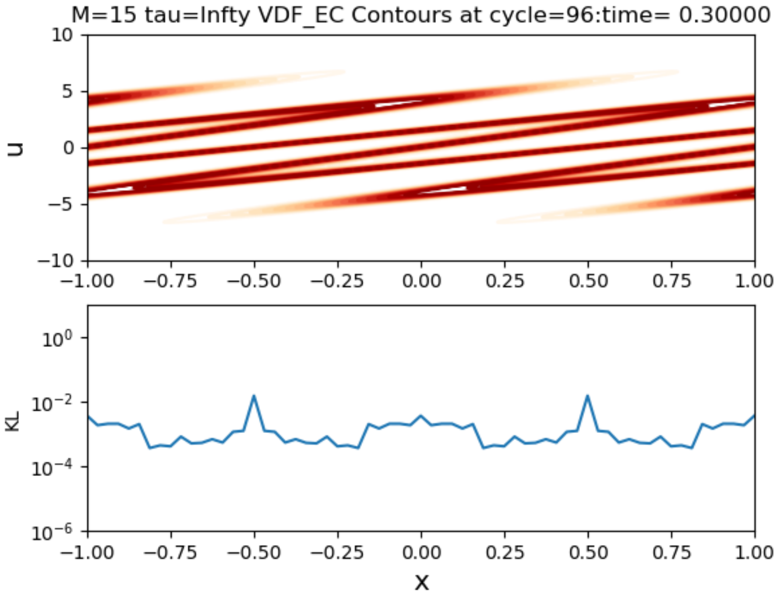

Figure 3-21 Exact collisionless solution for $M = 15$ at cycle 096.

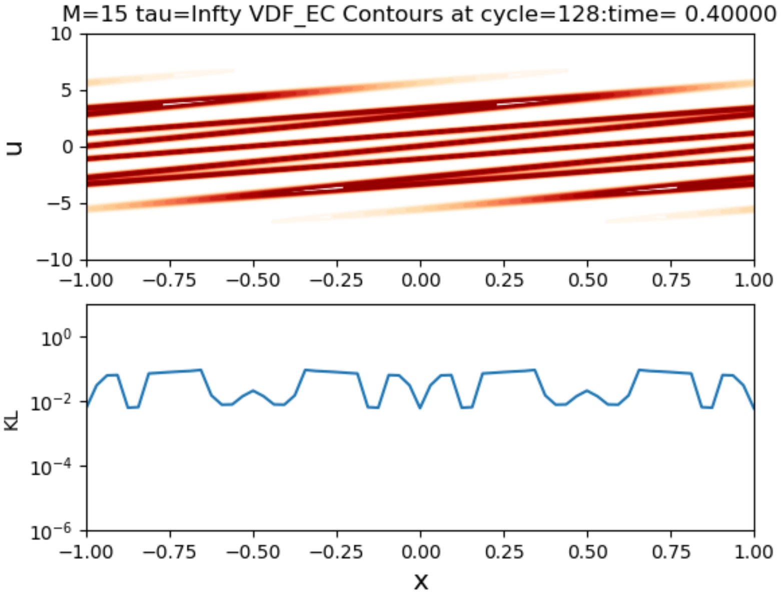

Figure 3-22 Exact collisionless solution for $M = 15$ at cycle 128.

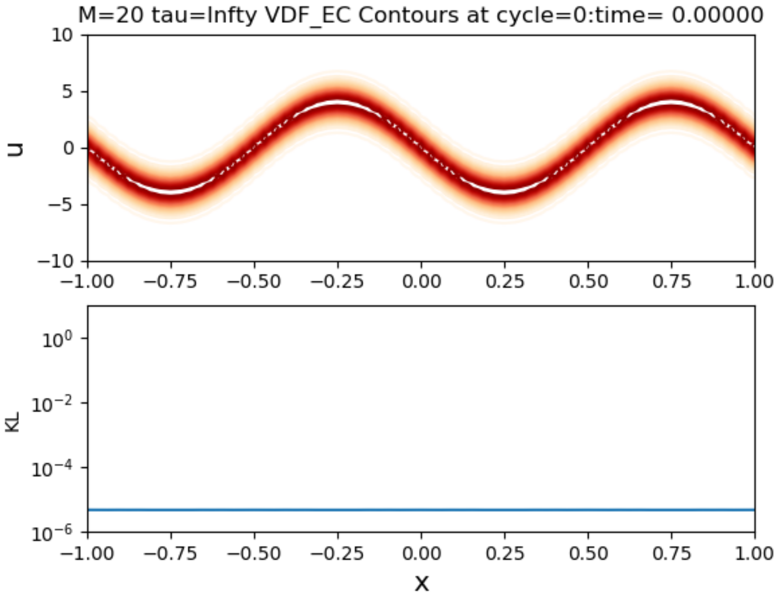

Figure 3-23 Exact collisionless solution for $M = 20$ at cycle 000.

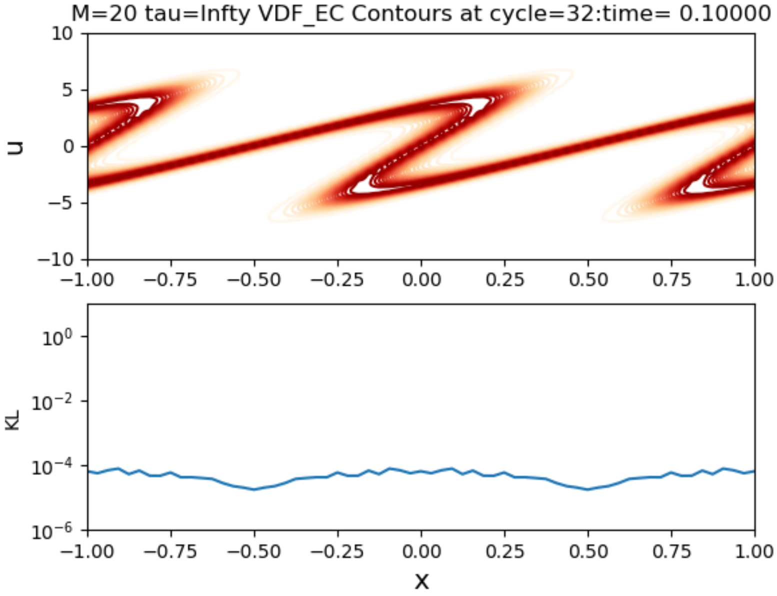

Figure 3-24 Exact collisionless solution for $M = 20$ at cycle 032.

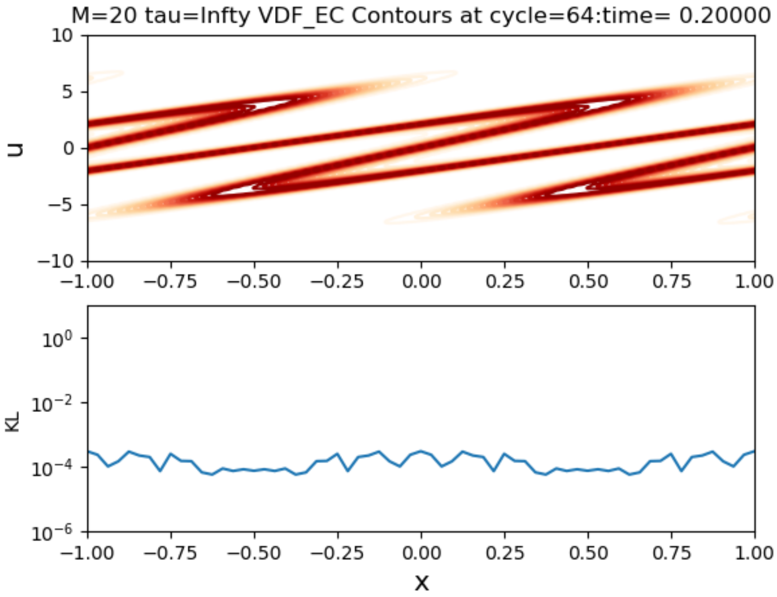

Figure 3-25 Exact collisionless solution for $M = 20$ at cycle 064.

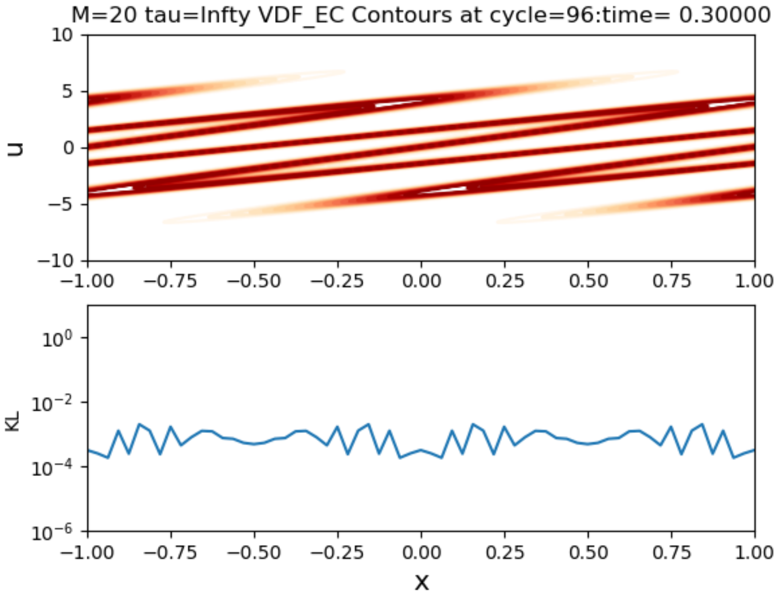

Figure 3-26 Exact collisionless solution for $M = 20$ at cycle 096.

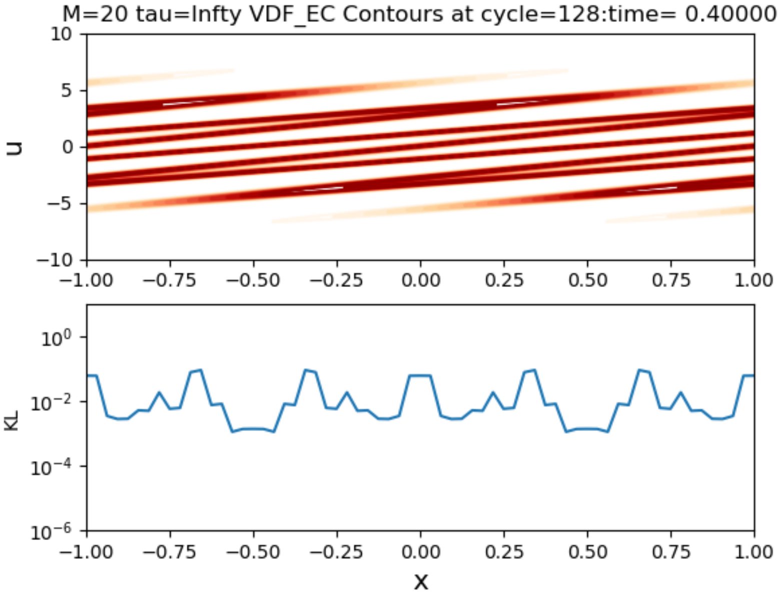

Figure 3-27 Exact collisionless solution for $M = 20$ at cycle 128.

3.4.2. *Collisional results*

The plots in the previous Section 3.4.1 demonstrate the ability of the GMM to match the exact collisionless solution. In contrast here in Section 3.4.2, we show results in which the distribution functions for a given spatial stencil are continually uncompressed, the distribution function moved forward in time via the local advection and collision update algorithm, and then recompressed to a new GMM state. With the addition of a collisional operator $C(f)$, we expect to eventually see the solution pushed toward a Maxwellian which should be matched well by a single term $M = 1$ GMM. We will need however sufficient resolution to pass through the transition from a sine wave distribution function attempting to evolve to a multi-peak distribution but balanced by collisions pushing the solution toward a single Gaussian.

Figures 3-28 to 3-32 shows the velocity distribution function at five linearly varying points in time for a strongly collisional operator with $\tau = 10^{-4}$ with $M = 20$, the EM algorithm and MPDATA advection. One observes the strongly collisional nature of the solution pushing toward a single Gaussian.

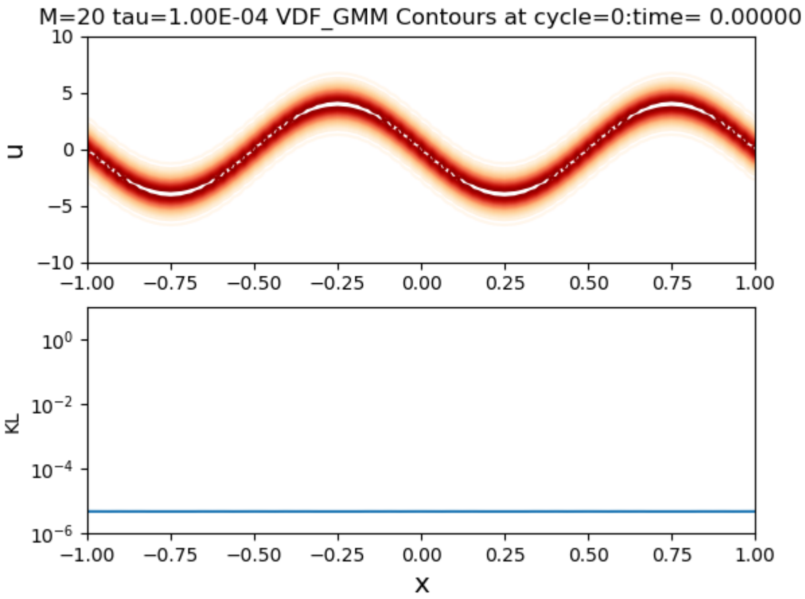

Figure 3-28 Collisional GMM $\tau = 10^{-4}$ solution at cycle 000

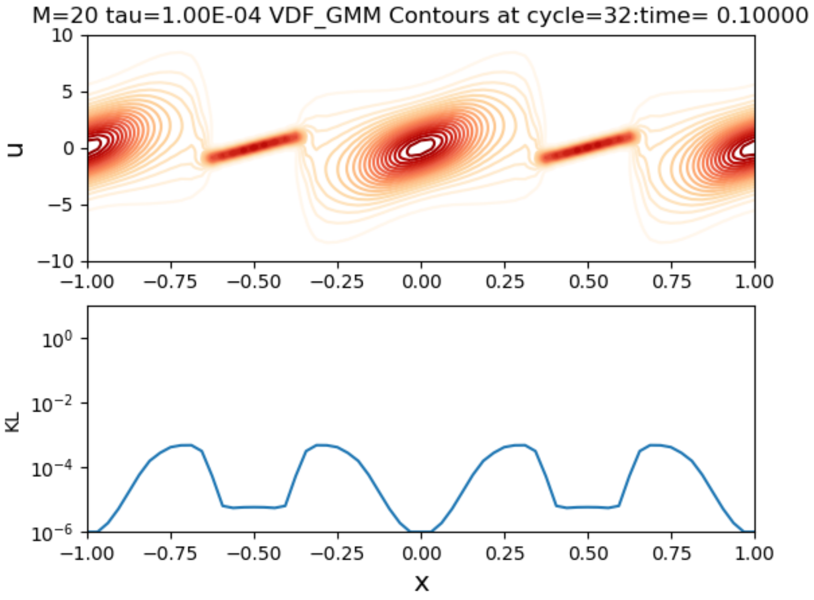

Figure 3-29 Collisional GMM $\tau = 10^{-4}$ solution at cycle 032

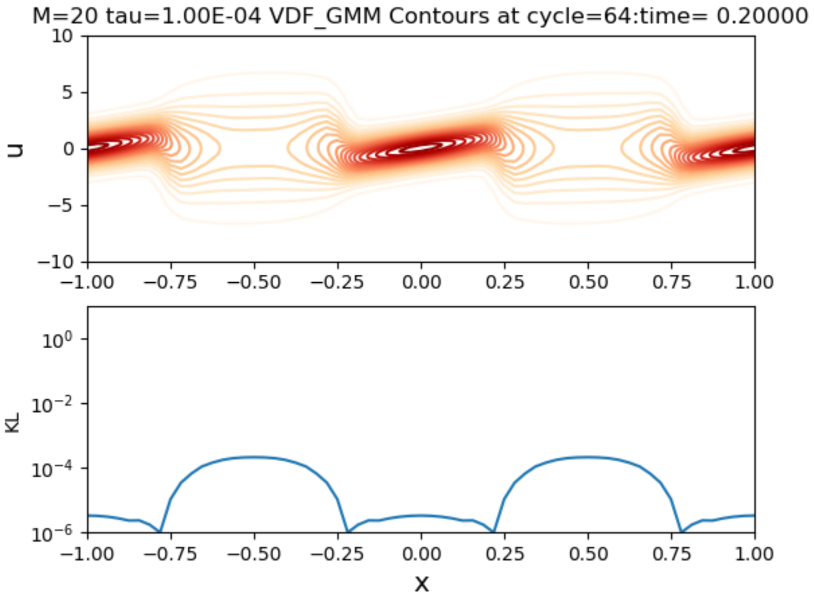

Figure 3-30 Collisional GMM $\tau = 10^{-4}$ solution at cycle 064

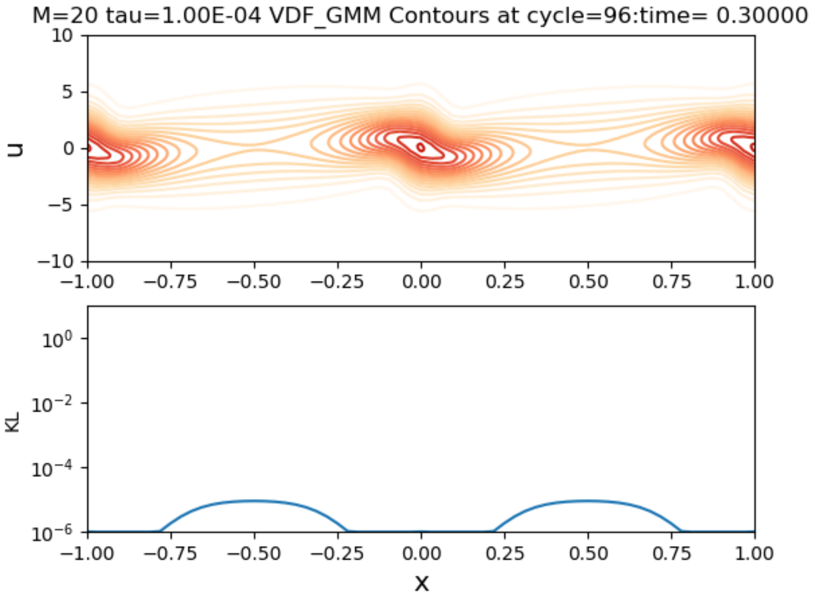

Figure 3-31 Collisional GMM $\tau = 10^{-4}$ solution at cycle 096

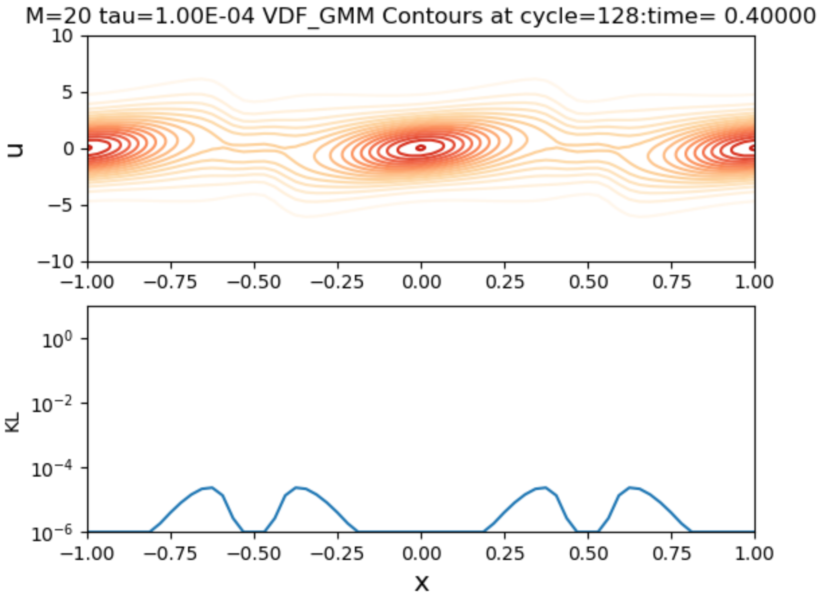

Figure 3-32 Collisional GMM $\tau = 10^{-4}$ solution at cycle 128

Figures 3-33 to 3-37 shows the velocity distribution function at 5 linearly varying points in time for $\tau = .01$ with $M = 20$, the EM algorithm and MPDATA advection. This is an intermediate collisional state.

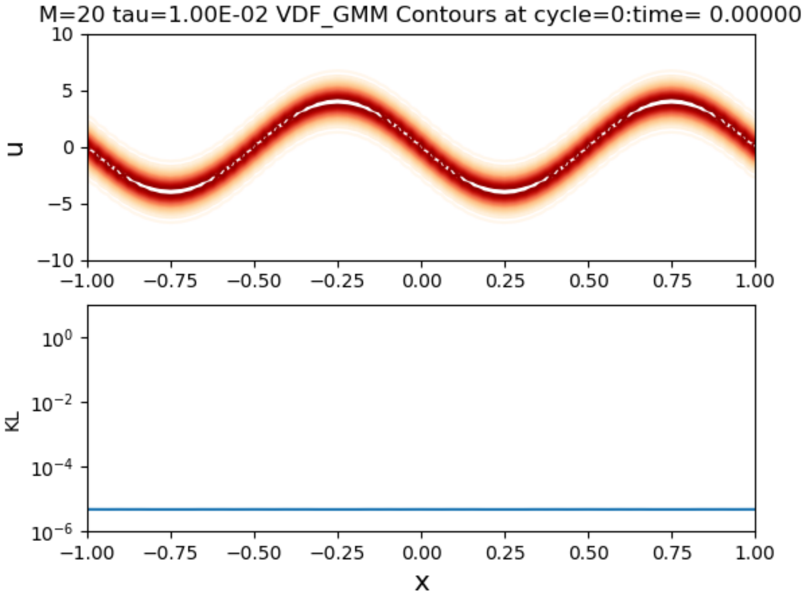

Figure 3-33 Collisional GMM $\tau = .01$ **solution at cycle** 000

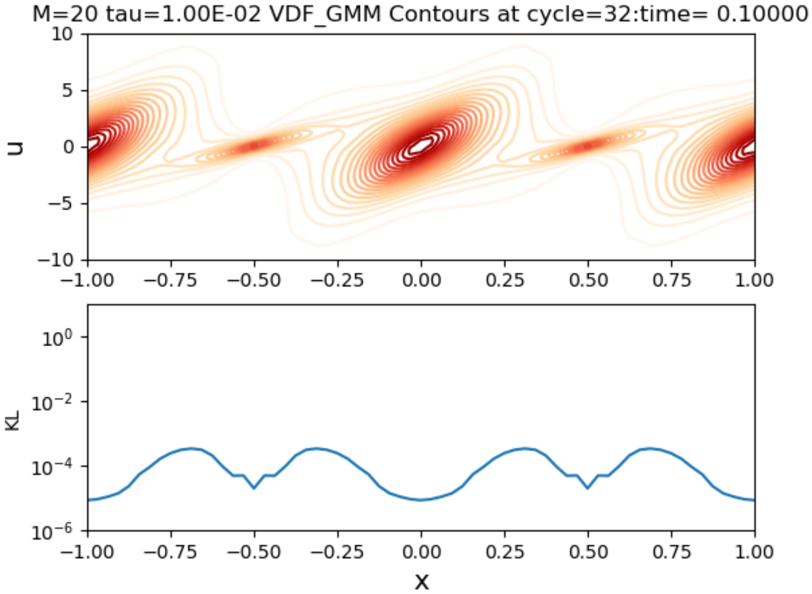

Figure 3-34 Collisional GMM $\tau = .01$ **solution at cycle** 032

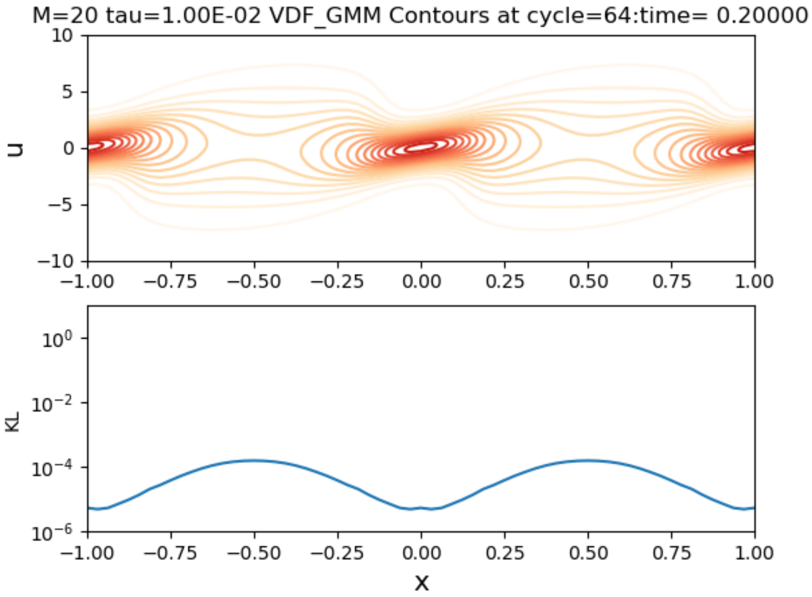

Figure 3-35 Collisional GMM $\tau = .01$ **solution at cycle** 064

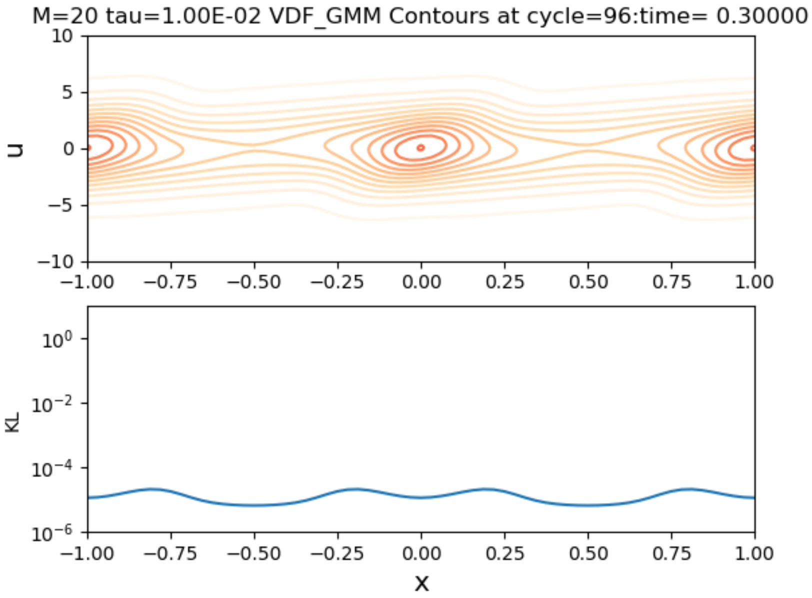

Figure 3-36 Collisional GMM $\tau = .01$ **solution at cycle** 096

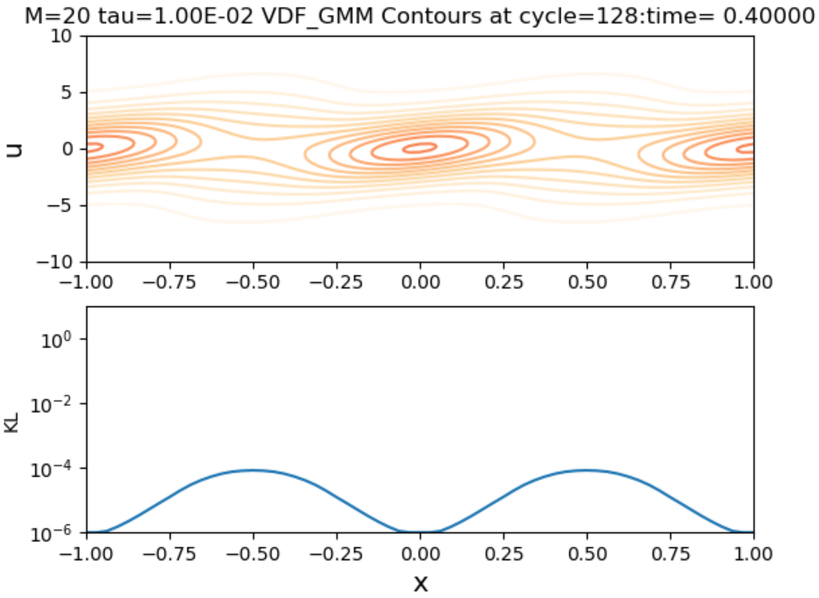

Figure 3-37 Collisional GMM $\tau = .01$ **solution at cycle** 128

Figures 3-38 to 3-42 shows the velocity distribution function at 5 linearly varying points in time for $\tau = 1$ with $M = 20$, the EM algorithm and MPDATA advection. This is a weakly collisional state.

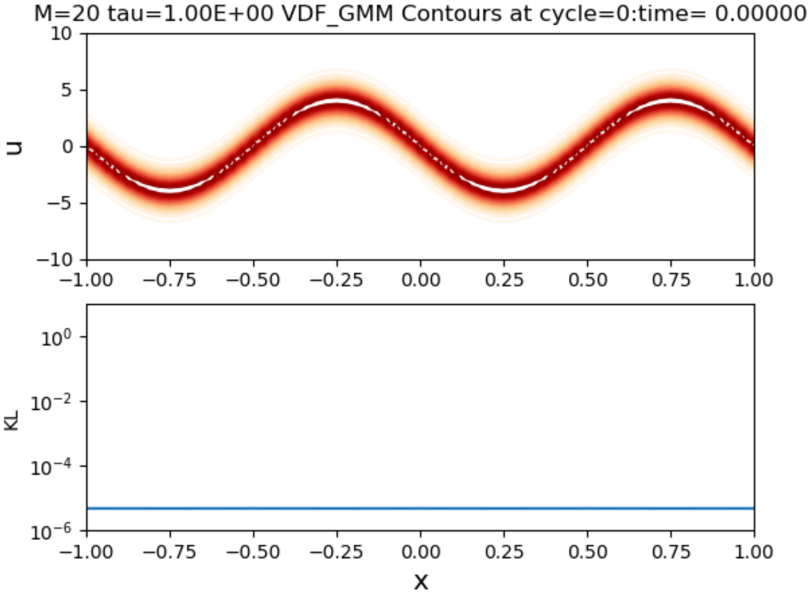

Figure 3-38 Collisional GMM $\tau = 1$ solution at cycle 000

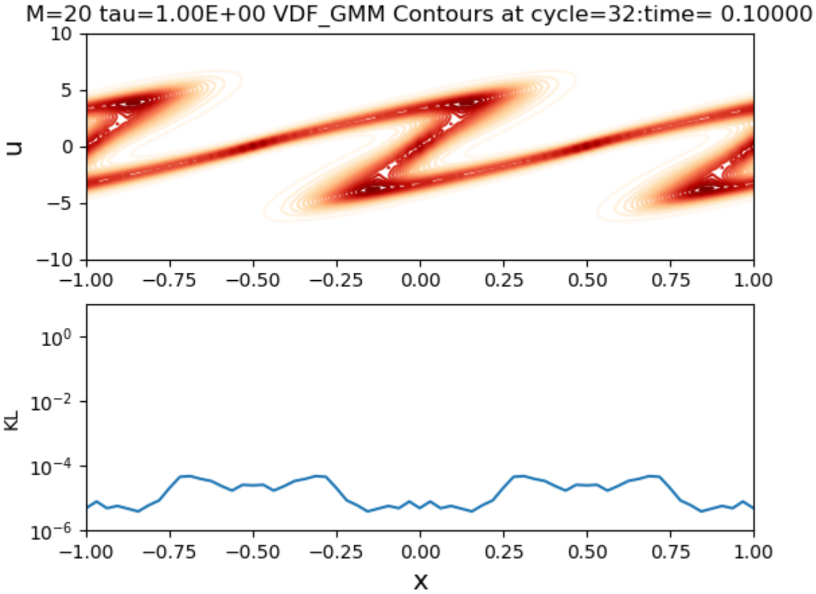

Figure 3-39 Collisional GMM $\tau = 1$ solution at cycle 032

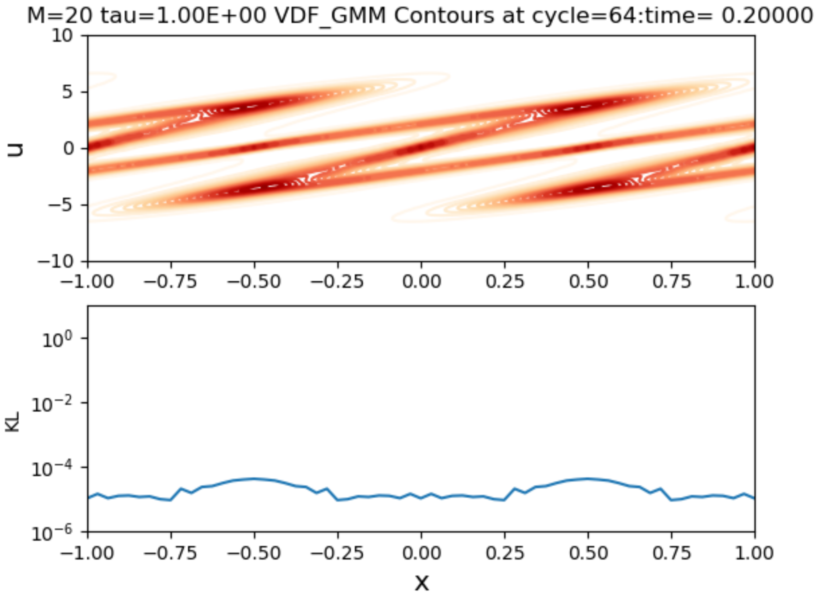

Figure 3-40 Collisional GMM $\tau = 1$ solution at cycle 064

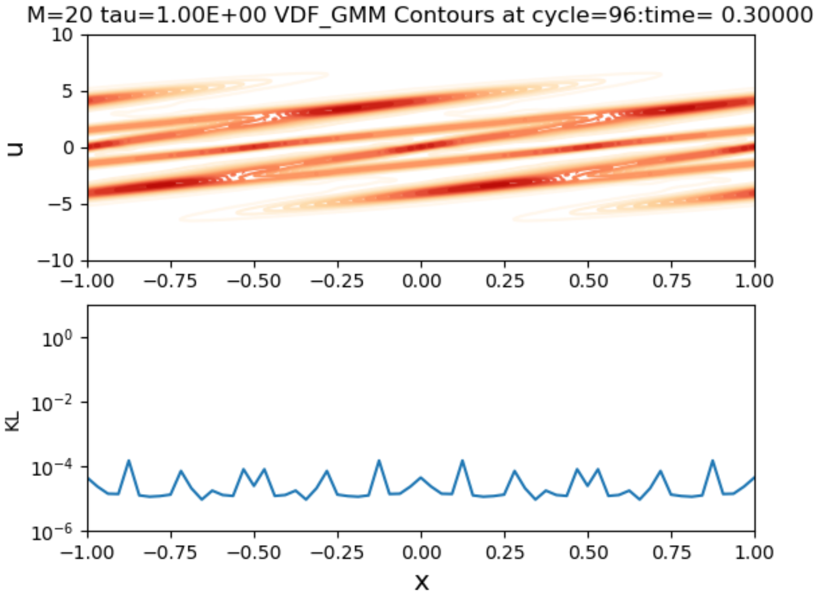

Figure 3-41 Collisional GMM $\tau = 1$ solution at cycle 096

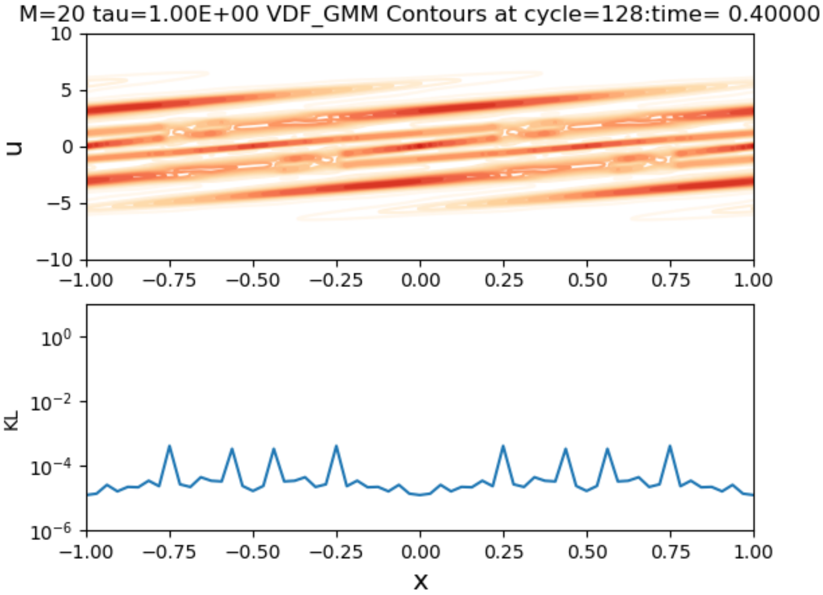

Figure 3-42 Collisional GMM $\tau = 1$ **solution at cycle** 128

Figures 3-43 to 3-47 shows the velocity distribution function at 5 linearly varying points in time for $\tau = 10^4$ problem with $M = 20$, the EM algorithm and the MPDATA advection. This is a nearly collisionless problem and one should observe the favorable comparison with Figures 3-23 through 3-27.

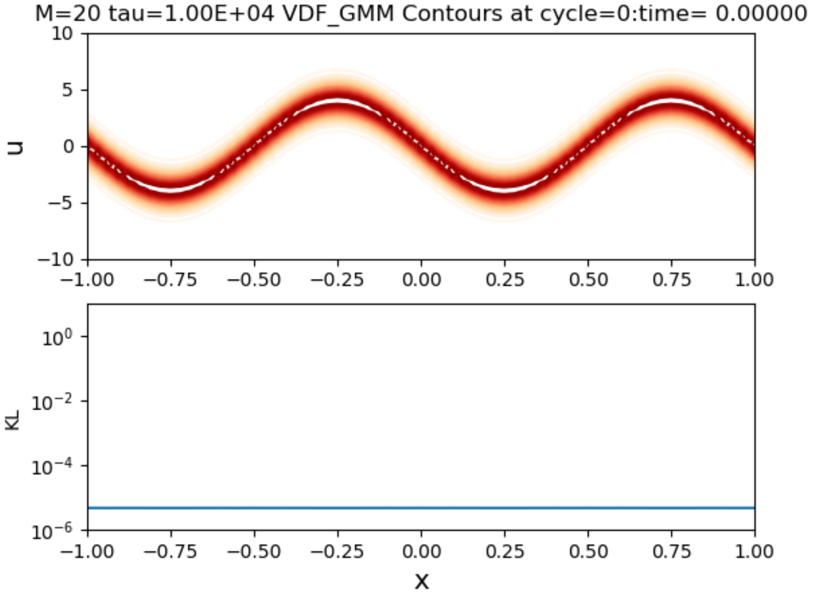

Figure 3-43 Collisional GMM $\tau = 10^4$ solution at cycle 000

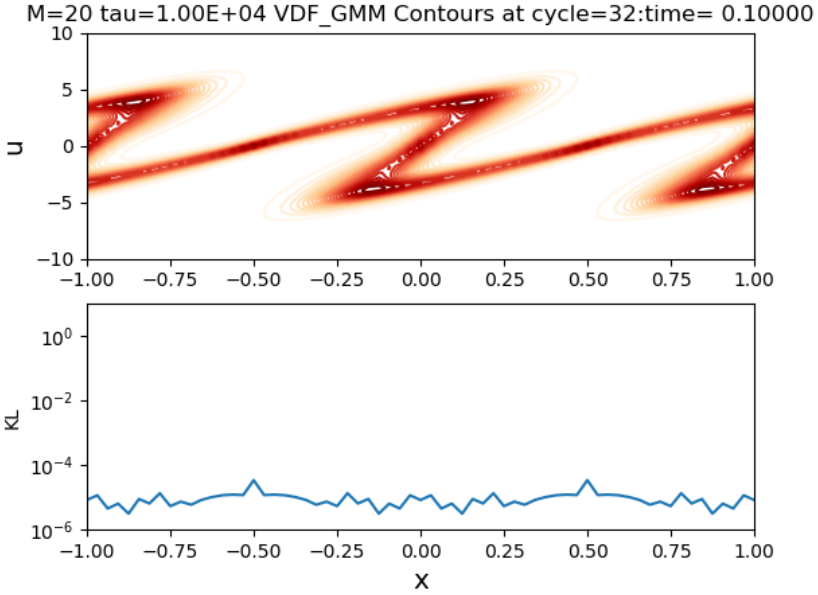

Figure 3-44 Collisional GMM $\tau = 10^4$ solution at cycle 032

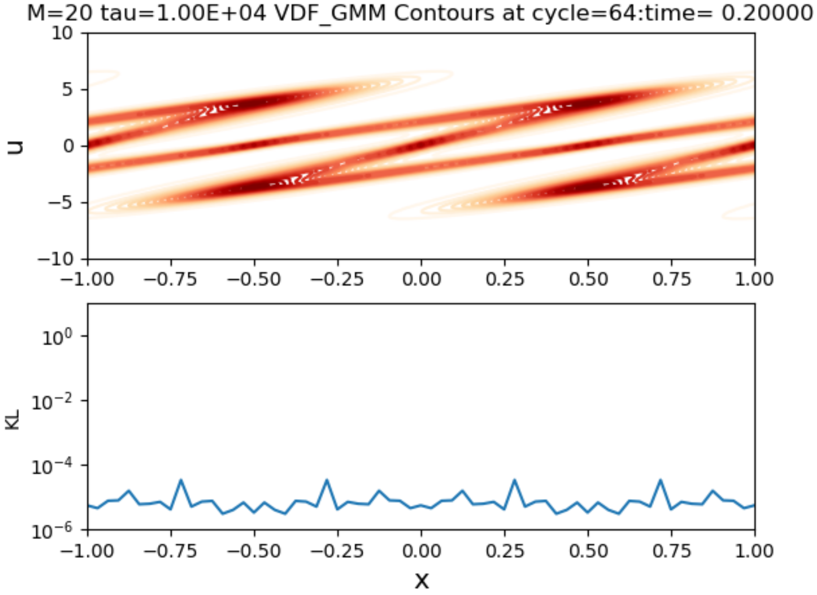

Figure 3-45 Collisional GMM $\tau = 10^4$ **solution at cycle** 064

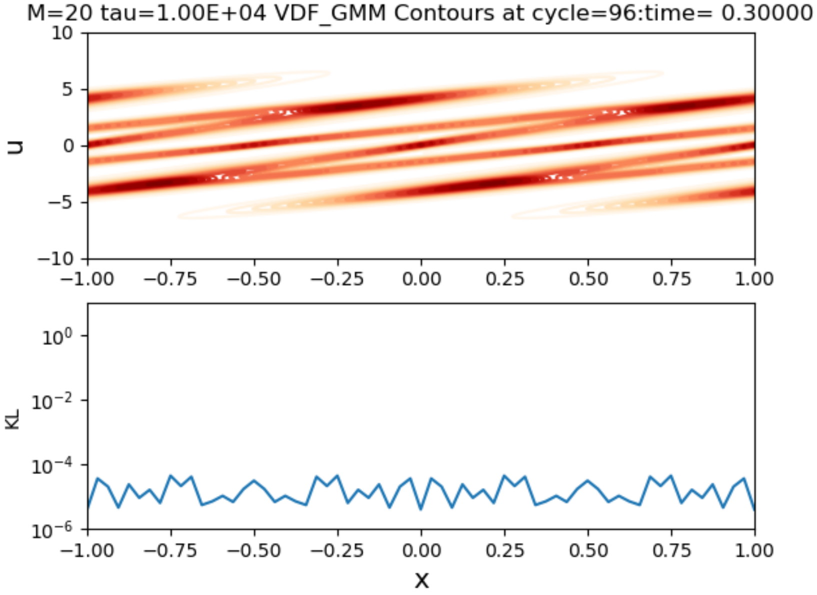

Figure 3-46 Collisional GMM $\tau = 10^4$ **solution at cycle** 096

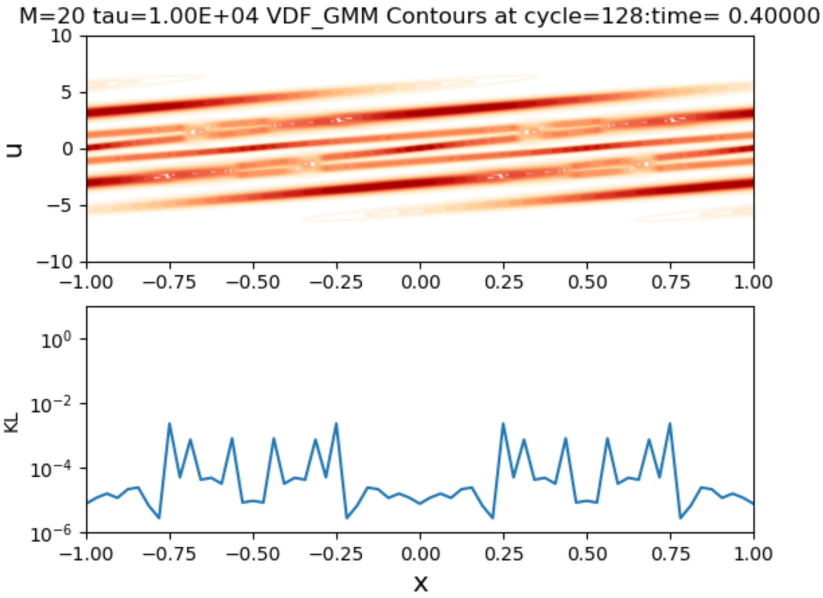

Figure 3-47 Collisional GMM $\tau = 10^4$ solution at cycle 128

We now attempt to gain a better picture of features of these computations that may give insight into efficacy and efficiency. In Figure 3-48 the maximal value of the KL divergence over all space-time points is plotted as a function of the number of available Gaussians and the collision time in the BGK model using a first order positive upwind advection scheme and the standard EM algorithm. The contours are derived from a [$\log_{10}\tau$] by [M] = [-4 2 0 2 4] by [1 5 10 15 20] grid of simulation runs. A large maximal KL divergence implies that the solution at some point in space or time became excessively complicated and the GMM solution quality suffered either because there were not enough degrees of freedom available or because the algorithm terminated and moved on using the current best available solution. Figure 3-48 shows that for high collisionality fewer Gaussians are needed to achieve a given KL divergence value. As collisions become less dominant we see that one quickly transitions into a state where many Gaussians are needed. Figure 3-49 shows the corresponding contours of maximum iterations.

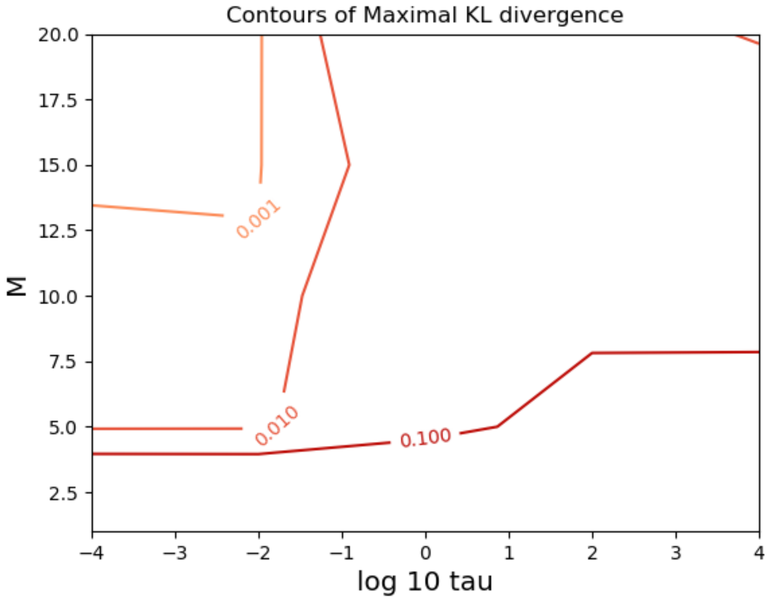

Figure 3-48 Maximum of KL divergence (over space and time) for low order upwind advection.

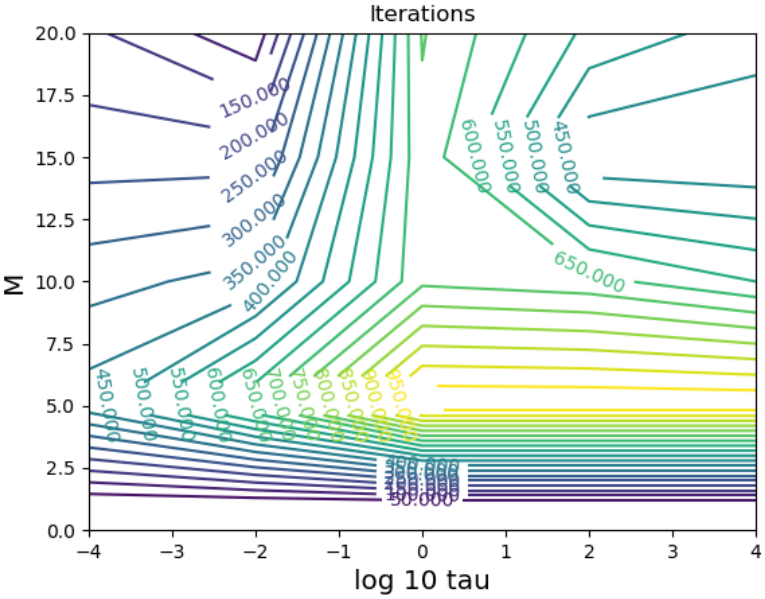

Figure 3-49 Maximum iterations (over space and time) for low order upwind advection.

For a comparison with high order advection, in Figure 3-50 the maximal value of the KL divergence over all space-time points is plotted as a function of the number of available Gaussians and the collision time in the BGK model using the MPDATA second order positive advection scheme and the standard EM iteration scheme. Figure 3-51 shows the corresponding maximal EM iterations. Again we see from Figure 3-50 that for high collisionality fewer Gaussian are

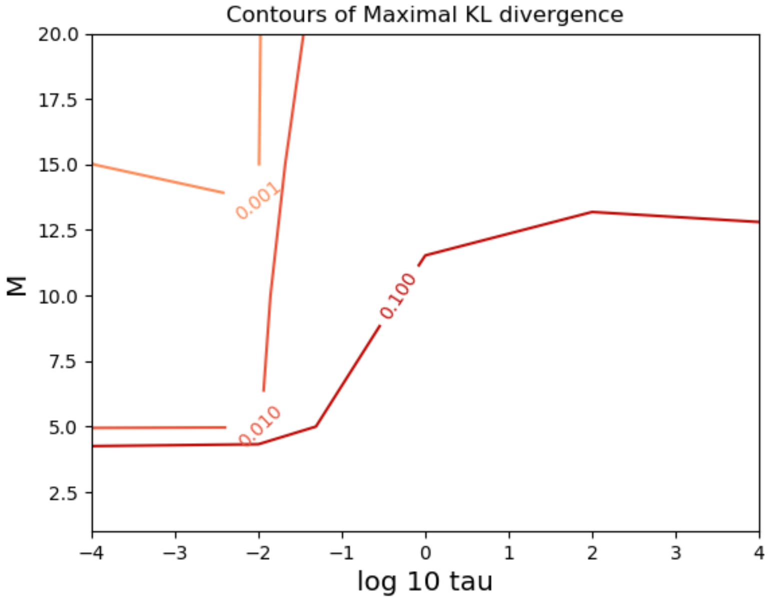

Figure 3-50 Maximum of KL divergence (over space and time) for high order MPDATA advection.

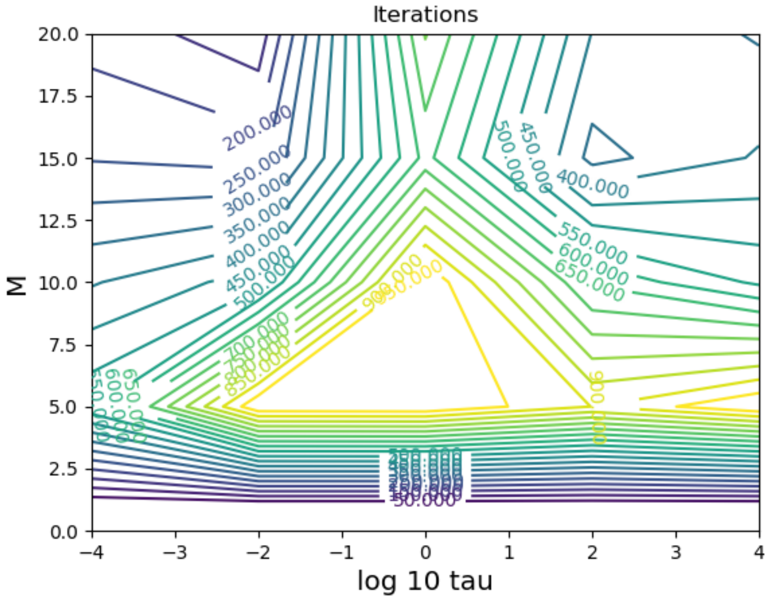

Figure 3-51 Maximum iterations (over space and time) for high order MPDATA advection.

needed. As collisions become less dominant we see that one quickly transitions into a state where many more Gaussians are required.

The above maps for the Upwind and MPDATA algorithm cases are very similar. Large numbers of iterations are generally required with cases of smaller collision times and larger values of M being slightly better. Very small values of M can converge relatively quickly but with very poor values of KL divergence.

We now show the same plots for the S3 algorithms for the MPDATA case. In Figure 3-52 the maximal value of the KL divergence is plotted as a function of the number of available Gaussians and the collision time for the MPDATA second order positive advection scheme. Figure 3-53 shows maximum space-time EM iterations.

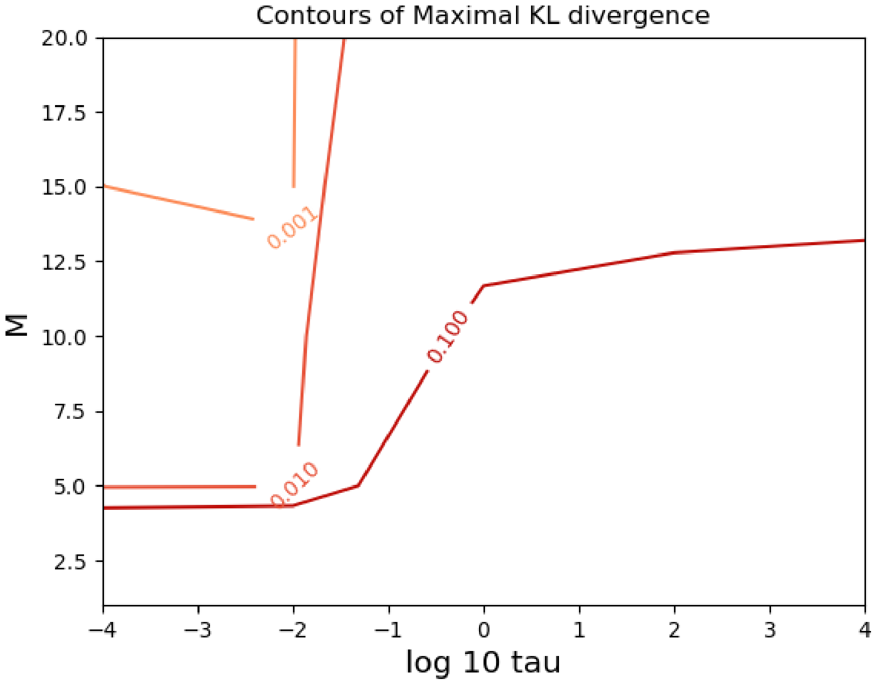

Figure 3-52 Maximum of KL divergence (over space and time) for MPDATA/S3 algorithm.

Next we show the plots for the SqS3 algorithm for the MPDATA case. In Figure 3-54 the maximal value of the KL divergence over all space-time points is plotted as a function of the number of available Gaussians and the collision time for the MPDATA second order positive advection scheme. Figure 3-55 shows maximum space-time EM iterations.

Finally we show the plots for the EPSILON algorithm for the MPDATA case. In Figure 3-56 the maximal value of the KL divergence is plotted as a function of the number of available Gaussians and the collision time in the BGK model for the MPDATA second order positive advection scheme. Figure 3-57 shows maximum space-time iterations.

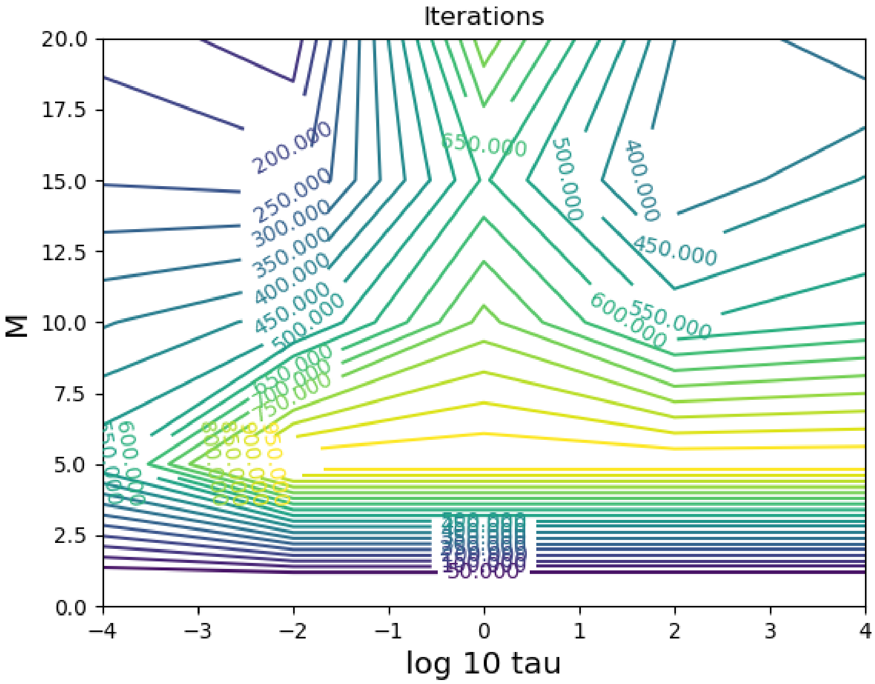

Figure 3-53 Maximum iterations (over space and time) for MPDATA/S3 algorithm.

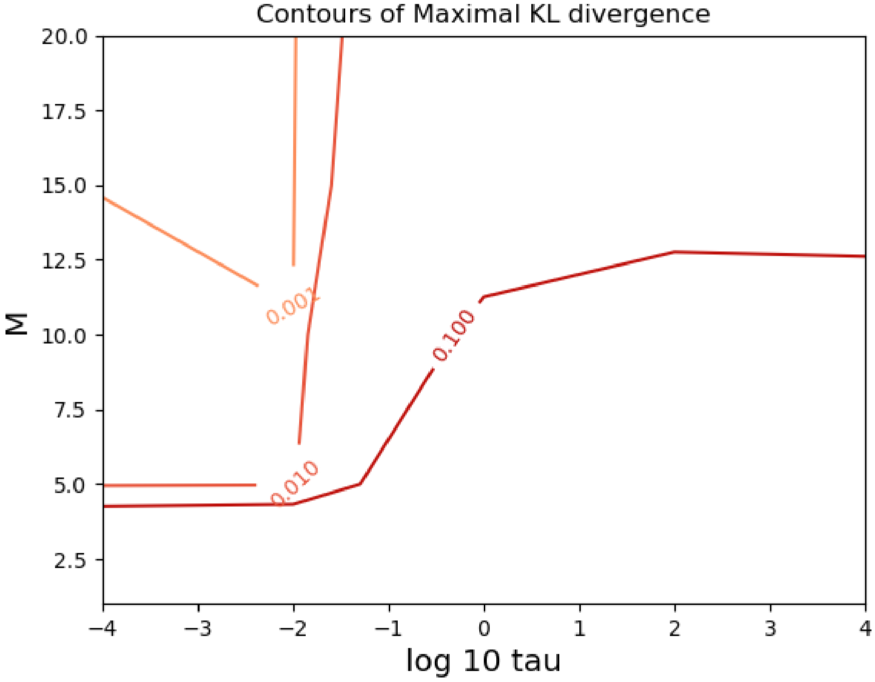

Figure 3-54 Maximum of KL divergence (over space and time) for MPDATA/SqS3 algorithm.

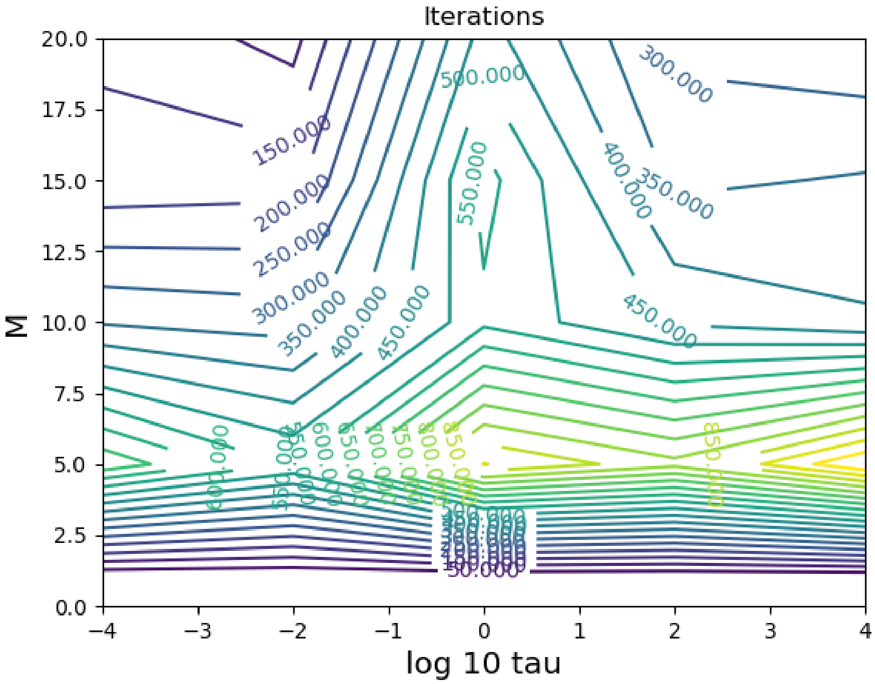

Figure 3-55 Maximum iterations (over space and time) for MPDATA/SqS3 algorithm.

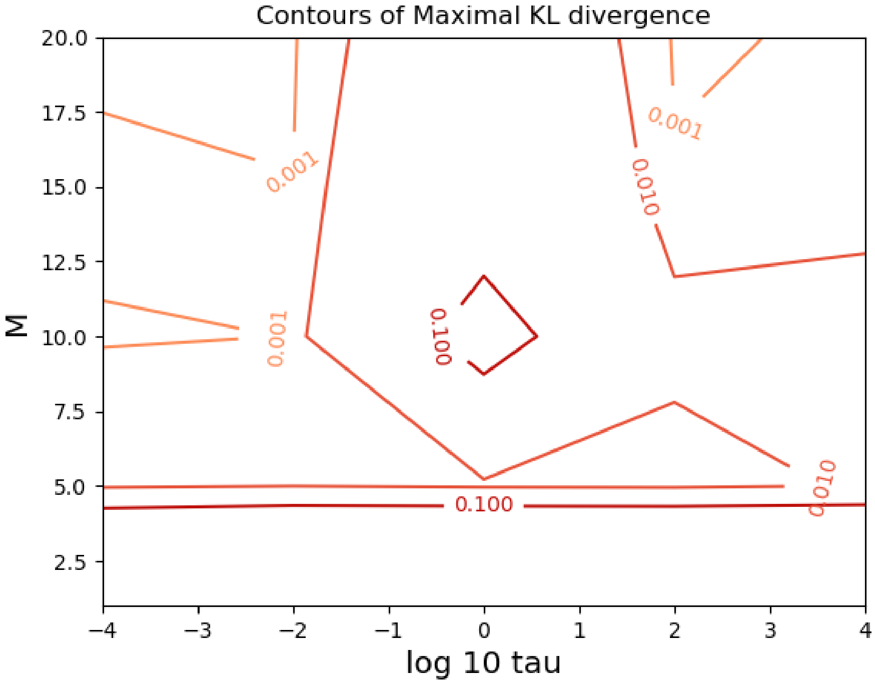

Figure 3-56 Maximum of KL divergence (over space and time) for MPDATA/EPSILON algorithm.

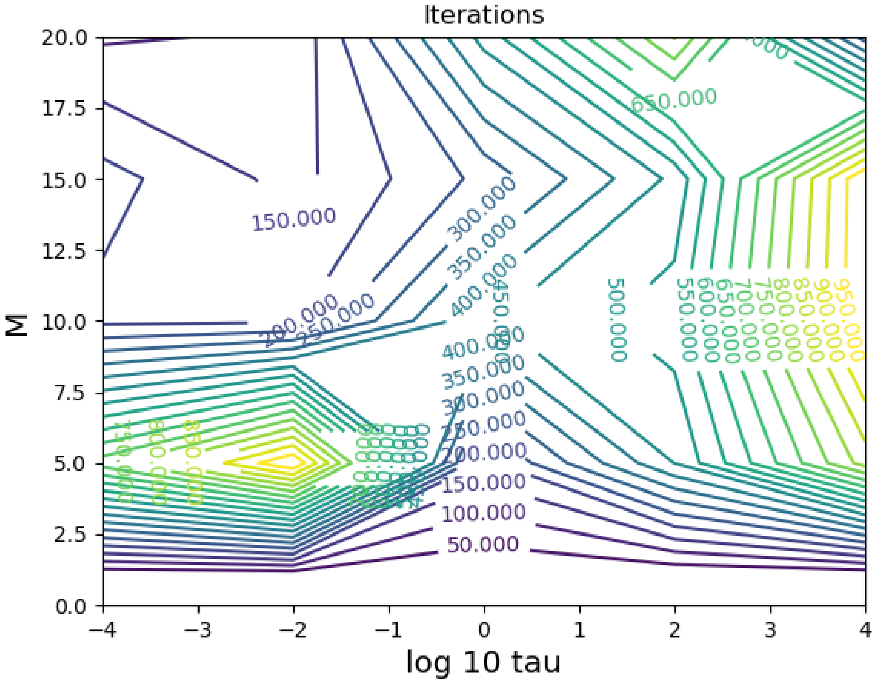

Figure 3-57 Maximum iterations (over space and time) for MPDATA/EPSILON algorithm.

In general all these results are consistent with our intuition. We observe lower values of the KL divergence for smaller collision times and higher numbers of Gaussians. We generally observe a large number of maximal iterations (order several hundred) in order to satisfy the stopping criterion at $\varepsilon = 10^{-6}$. In many cases the particular stopping criterion that is satisfied is the change in the KL divergence and not the KL divergence itself. However, it is clear that – given enough Gaussians in the GMM – a relatively small and reasonably uniform value of the KL divergence can often be obtained. The number of iterations shown is only indicative of cost since this number represents the outer iteration loop count and not the number of internal expectation maximum calls internal to each EM acceleration step. It is also the maximum value and thus gives no indication of the average cost. Table 3-1 provides one snapshot of a set of computing times displayed relative to the EM-MPDATA times. The computing time ratios shown include 25 runs that scan across the collisionality and number of M values ([$\log_{10}\tau$] by [M] = [-4 2 0 2 4] by [1 5 10 15 20]).

	EM	S3	SqS3	EPSILON
MPDATA	1.0	2.2	1.5	.67
Upwind	1.13	2.53	1.73	.68

Table 3-1 A single snapshot comparison of compute times scaled by the EM-MPDATA case for various algorithmic options.

It is interesting that the EPSILON algorithm seems to be slightly more efficient on average than the standard EM algorithm and that our implementation of the S3 and SqS3 algorithm are less efficient than the standard EM algorithm. The more dissipative Upwind algorithm also results in slightly higher computing times on average.

We believe that we have given sufficient evidence to show that a GMM model can be effective as a fundamental compressed form representation for arbitrary distributional states as part of a Boltzmann equation solver. Additional research toward an implementation of an EM algorithm containing both sufficient initialization coverage and an effective algorithmic speedup while reliably computing representations with sufficiently low KL divergence values is also seen as necessary.

3.5. Fluid Limit

One of the key aspects of this work is the ability for these methods to accurately and efficiently capture the strongly collisional limit of fluid dynamics. In Figures 3-58, 3-59 and 3-60 we present the mass density, velocity and pressure at a simulation time = 0.08. The line labeled as fluid is obtained from a high order Finite Volume solution of Euler's equations with a ratio of specific heats $\gamma = 3$, appropriate for a $(1x, 1u)$ kinetic system. As Euler's equations are consistent with the Boltzmann equation in the limit of vanishingly small collision time, or collisional mean free path, the expectation is that the kinetic solution should recover the solution for Euler's equations at sufficiently small τ. The other two lines plotted in these figures are obtained from the GMM method using $M = 9$, a BGK collision time $\tau = 10^{-4}$ and a collisionless result.

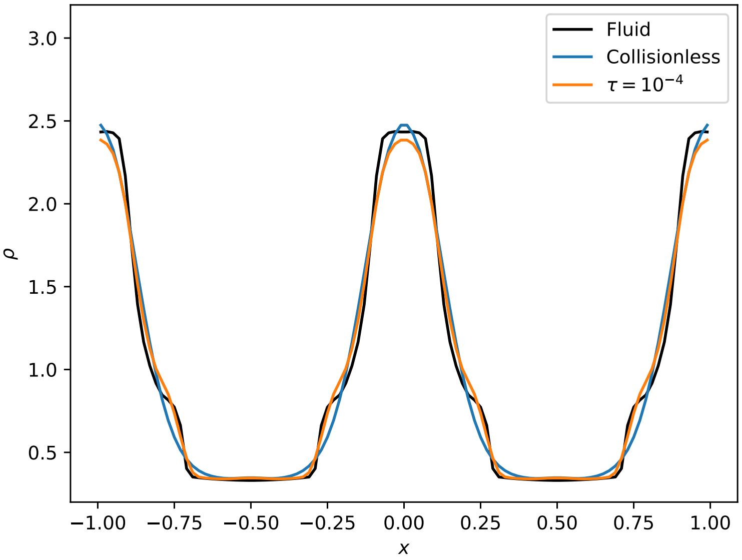

Figure 3-58 This plot shows the mass density obtained from a solution of Euler's equations and the GMM collisional kinetic solver for a collisionless case and BGK collision time $\tau = 10^{-4}$.

Comparing the three figures, it is immediately apparent that the GMM solution with $\tau = 10^{-4}$ captures the development of shock waves, the jumps in the velocity and pressure, the position of the shock waves, etc. This critical aspect of shock hydrodynamics is clearly apparent and the good agreement is likely dependent on the fact that both the fluid and kinetic solvers are conservative. It is also clearly apparent just how important collisions are in recovering fluid dynamics based on the comparison of the collisionless and collisional kinetic solutions. Perhaps the most surprising detail in these plots is the relatively close solutions for the density.

A detailed numerical study could (and should) repeat these calculations and quantitatively measure the convergence to the fluid solution with mesh resolution, number of Gaussians M, and collision time τ. Looking toward the future, these results are extremely encouraging. Extending this work to the Vlasov system and including collisions, one could, in principle, study a host of plasma physics that spans the limits from the weakly collisional kinetic limit through to single-fluid magnetohydrodynamics.

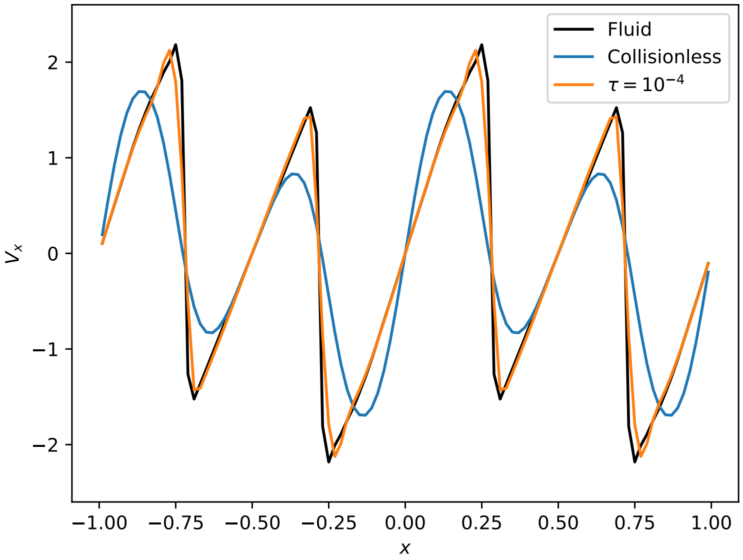

Figure 3-59 This plot shows the velocity obtained from a solution of Euler's equations and the GMM collisional kinetic solver for a collisionless case and BGK collision time $\tau = 10^{-4}$.

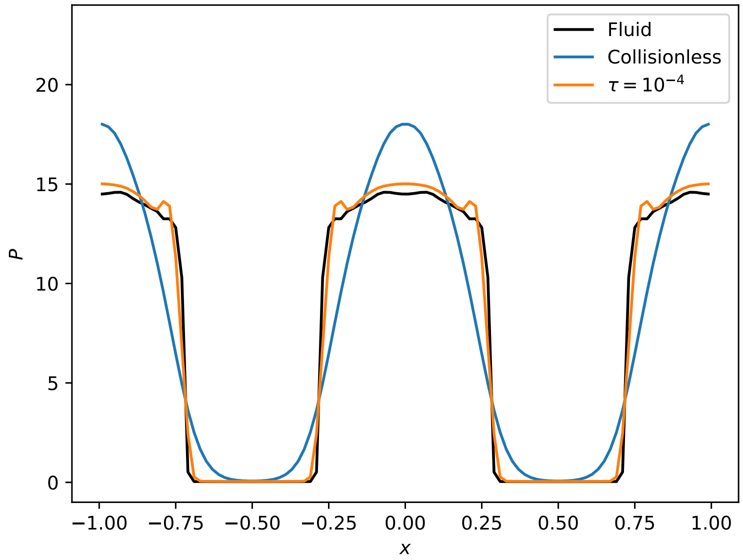

Figure 3-60 This plot shows the pressure obtained from a solution of Euler's equations and the GMM collisional kinetic solver for a collisionless case and BGK collision time $\tau = 10^{-4}$.

3.6. Dimensional Extensions, Memory Requirements and Boltzmann Equation Algorithms

We have provided results for the case of one spatial and one velocity dimension. The approach can be extended to the multi-variate D-dimensional case with

$$\mathcal{N}(\mathbf{u};\mu,\mathbf{C}) = \frac{1}{(2\pi)^{D/2}\sqrt{|C|}} \exp\left(-\frac{1}{2}(\mathbf{u}-\mu)^T\mathbf{C}^{-1}(\mathbf{u}-\mu)\right). \tag{96}$$

The EM algorithm proceeds in exactly the same way except that now there is a vector parameter μ of dimension D and a covariance matrix $\mathbf{C}$ with $D(D+1)/2$ independent parameters for a total of $M(D(D+1)/2+D+1)$ parameters. The advection operators must also be extended to the multidimensional case. We anticipate that similar results will be obtained in this multivariate setting.

Our current implementation of the EM algorithms described in Section 3 is in Python. The coding approach has an outer loop over spatial position that expands the Gaussian mixture model at each spatial stencil point to a relatively fine uniform velocity grid in order to implement the advection algorithms for all velocities and to apply NumPy vector operations. After the spatial point is updated, the EM compression algorithm is then applied to the updated distribution function. In this implementation we have additional memory requirements on the order of the velocity grid size. In this work we arbitrarily choose a fixed M value under the assumption that it will be sufficient to represent adequately the evolving distribution function.

However, the ideal and ultimate goal is to build a compressed velocity space representation that can grow or shrink M depending on the solution requirements. Also, utilizing a local velocity grid may not be practical or desirable. This suggests that what may be ultimately needed is a real-time, incremental GMM algorithm where the GMM parameters are continuously updated based on new distribution information as it is computed numerically while moving through velocity space. There is a base of promising literature to draw on in order to expand our research in this direction. Since in our application we are effectively replacing data as we walk through velocity space, this is equivalent to the problem of dynamic "concept drift" in the field of streaming machine learning. The work of Oliviera et. al. propose an approach for modeling large amounts of incoming data while distinguishing between "virtual" and "real" concept drift [25, 26]. In our application, we are not particularly concerned with the number of Gaussians in the mixture ("real concept drift") other than our desire to minimize this number under the constraint of maintaining a chosen level of distribution fidelity. Id and colleagues have proposed an algorithm for detecting and adapting a GMM model to concept drift [18]. The work of Arandjelovic and Cipollas indicates that an incremental GMM approach that locally updates parameters and can also build in splitting and merging (i.e. changes in M) is possible in the case of "temporally coherent" or "temporally correlated" data [2]. Such "temporal coherence" is to be expected when walking through velocity space updates. Chen, et. al. used a GPU-accelerated based incremental learning algorithm for GMM modeling [9]. Computer hardware and algorithms optimized for real-time GMM machine learning such as for real-time video analysis may be very effective [37, 15]. A GPU algorithm parallelized over space with robust GMM updating while traversing velocity space updates is potentially feasible. The memory required for such an algorithm would scale with the number of grid points times twice the number of required GMM parameters. Advances in the streaming GMM machine learning community either in the algorithmic or computer science arena will directly impact the viability of our proposed GMM approach. It thus appears possible that a truly effective GMM approach for Boltzmann equation modeling may most likely be achieved through a close collaboration with the real-time, streaming-data machine learning community.

4. CONCLUSION

In this report we explored a set of mathematical representations for the single particle distribution function including asymmetric Hermite basis, Gaussian Positive Polynomial (GPP) and Gaussian Mixture Model (GMM). We have shown that a GMM provides a useful and compact representation of the velocity distribution function for a one-space and one-velocity dimension Boltzmann equation. This was shown by computing the KL divergence of the GMM representation from the exact solution for a complicated collisionless test case. We also utilized the GMM representation for collisional cases with a first order IMEX numerical scheme with positivity preserving numerical advection operators and examined the quality of the solution approach by computing the maximal KL divergence computed at the end of each compression step. For best results, we found that a reinitialized initial guess is best for mapping the distribution function to the Gaussian mixture model. This allows for a better capture of variations in the distribution function as they arise from the advection operator acting on different points in velocity space. We achieved promising results across a range of collisionality that give evidence to the feasibility of using this classical approach from the machine learning community in a fundamental physics modeling role. One element of this approach that should be emphasized is the general applicability to both the non-relativistic and relativistic collisional Vlasov system. This stems from the fact that the GMM, and GPP, representation relies on non-negative, linearly independent basis functions rather than orthogonality. Hence for relativistic systems, the Maxwell-Boltzmann (Gaussian) basis can be replaced with a Maxwell-Jüttner (modified Bessel function of the second kind) basis.

Looking forward, one open question is whether a mixture model based on the GPP might offer a competitive advantage by capturing well separated peaks in the distribution function with a Gaussian and locally shaping the Gaussian with the polynomial. The major issue of concern needing future research is how to improve the computational speed and memory usage of the GMM compression step while at the same time maintaining a flexible and minimal GMM representation that provides robust uniform accuracy. We speculate that mutually benefiting progress may be most rapidly achieved by partnering with the real-time, streaming-data machine learning community.

REFERENCES

[1] F. Allmann-Rahn, R. Grauer, and K. Kormann. A parallel low-rank solver for the six-dimensional Vlasov–Maxwell equations. *Journal of Computational Physics*, page 111562, 2022.

[2] Ognjen Arandjelovic and R. Cipolla. Incremental learning of temporally-coherent Gaussian mixture models. In *BMVC 2005: Proceedings of the British Machine Conference 2005*, pages 59–1 (759) – 59–10 (768). BMVA Press, 2005.

[3] Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77 – 120, 2017.

[4] A.F. Berlinet and Ch. Roland. Parabolic acceleration of the EM algorithm. *Statistics and Computing*, 19:35–47, 2009.

[5] A.F. Berlinet and Ch. Roland. Acceleration of the EM algorithm: P-EM versus epsilon algorithm. *Computational Statistics & Data Analysis*, 56(12):4122–4137, 2012.

[6] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511–525, May 1954.

[7] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003. Recent Developments in Mixture Model.

[8] Claude Brezinski. Some results and applications about the vector ε-algorithm. *Rocky Mountain Journal of Mathematics*, 4(2):335 – 338, 1974.

[9] Chunlei Chen, Dejun Mu, Huixiang Zhang, and Bo Hong. A GPU-accelerated approximate algorithm for incremental learning of Gaussian mixture model. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pages 1937–1943, 2012.

[10] G. Chen, L. Chacón, and T.B. Nguyen. An unsupervised machine-learning checkpoint-restart algorithm using gaussian mixtures for particle-in-cell simulations. *Journal of Computational Physics*, 436:110185, 2021.

[11] Jack Coughlin and Jingwei Hu. Efficient dynamical low-rank approximation for the Vlasov-Ampère-Fokker-Planck system. *Journal of Computational Physics*, page 111590, 2022.

[12] G.L. Delzanno. Multi-dimensional, fully-implicit, spectral method for the Vlasov-Maxwell equations with exact conservation laws in discrete form. *Journal of Computational Physics*, 301:338 – 356, 2015.

[13] Raaz Dwivedi, Nhat Ho, Koulik Khamaru, Martin J. Wainwright, Michael I. Jordan, and Bin Yu. Singularity, misspecification and the convergence rate of EM. *The Annals of Statistics*, 48(6):3161 – 3182, 2020.

[14] E. Gekeler. On the solution of systems of equations by the epsilon algorithm of Wynn. *Mathematics of Computation*, 26(118):427–436, 1972.

[15] Conghui He, Haohuan Fu, Ce Guo, Wayne Luk, and Guangwen Yang. A fully-pipelined hardware design for Gaussian mixture models. *IEEE Transactions on Computers*, 66(11):1837–1850, 2017.

[16] James Paul Holloway. Spectral velocity discretizations for the Vlasov-Maxwell equations. *Transport Theory and Statistical Physics*, 25(1):1–32, 1996.

[17] Jingwei Hu, Ruiwen Shu, and Xiangxiong Zhang. Asymptotic-preserving and positivity-preserving implicit-explicit schemes for the stiff BGK equation. *SIAM Journal on Numerical Analysis*, 56(2):942–973, 2018.

[18] Ibnu Daqiqil Id, Masanobu Abe, and Sunao Hara. Concept drift adaptation for acoustic scene classifier based on Gaussian mixture model. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 450–455, 2020.

[19] Masahiro Kuroda, Zhi Geng, and Michio Sakakihara. Improving the vector ε acceleration for the EM algorithm using a re-starting procedure. *Computational Statistics*, 30:1051–1077, 2015.

[20] Masahiro Kuroda and Michio Sakakihara. Accelerating the convergence of the EM algorithm using the vector ε algorithm. *Computational Statistics & Data Analysis*, 51(3):1549–1561, 2006.

[21] Geoffrey J. McLachlan, Sharon X. Lee, and Suren I. Rathnayake. Finite mixture models. *Annual Review of Statistics and Its Application*, 6(1):355–378, 2019.

[22] Sean T. Miller, Nathan V. Roberts, Stephen D. Bond, and Eric C. Cyr. Neural-network based collision operators for the Boltzmann equation. *Journal of Computational Physics*, page 111541, 2022.

[23] S.T. Miller, E.C. Cyr, J.N. Shadid, R.M.J. Kramer, E.G. Phillips, S. Conde, and R.P. Pawlowski. IMEX and exact sequence discretization of the multi-fluid plasma model. *Journal of Computational Physics*, 397:108806, 2019.

[24] Iftekhar Naim and Daniel Gildea. Convergence of the EM algorithm for Gaussian mixtures with unbalanced mixing coefficients, 2012.

[25] Gustavo Oliveira, Leandro L. Minku, and Adriano L I Oliveira. Tackling virtual and real concept drifts: An adaptive Gaussian mixture model approach. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[26] Gustavo H. F. M. Oliveira, Leandro L. Minku, and Adriano L. I. Oliveira. GMM-VRD: A Gaussian mixture model for dealing with virtual and real concept drifts. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[27] Lorenzo Pareschi and Giovanni Russo. Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific Computing*, 25:129–155, 2005.

[28] Nathan V. Roberts, Sean T. Miller, Stephen D. Bond, and Eric C. Cyr. An implicit-in-time DPG formulation of the 1D1V Vlasov-Poisson equations. Draft in Progress, 2022.

[29] Ruslan Salakhutdinov, Sam Roweis, and Zoubin Gharramani. Optimization with EM and Expectation-Conjugate-Gradient. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*. AAAI, 2003.

[30] Joseph W. Schumer and James Paul Holloway. Vlasov simulations using velocity-scaled Hermite representations. *Journal of Computational Physics*, 144(2):626 – 661, 1998.

[31] Emilie Shireman, Douglas Steinley, and Michael J. Brusco. Examining the effect of initialization strategies on the performance of Gaussian mixture modeling. *BEHAVIOR RESEARCH METHODS*, 49(1):282–293, FEB 2017.

[32] Piotr K. Smolarkiewicz. A simple positive definite advection scheme with small implicit diffusion. *Monthly Weather Review*, 111(3):479 – 486, 1983.

[33] Piotr K. Smolarkiewicz and Len G. Margolin. Mpdata: A finite-difference solver for geophysical flows. *Journal of Computational Physics*, 140(2):459–480, 1998.

[34] Tao Tang. The Hermite spectral method for Gaussian-type functions. *SIAM Journal on Scientific Computing*, 14(3):594–606, 1993.

[35] Ravi Varadhan and Christophe Roland. Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scandinavian Journal of Statistics*, 35(2):335–353, 2008.

[36] Mingfeng Wang, Masahiro Kuroda, Michio Sakakihara, and Zhi Geng. Acceleration of the EM algorithm using the vector ε algorithm. *Computational Statistics*, 23(3):469–486, 2008.

[37] Yuanfan Xu, Zhaoliang Zhang, Jincheng Yu, Jianfei Cao, Haolin Dong, Zhengfeng Huang, Yu Wang, and Huazhong Yang. GAME: Gaussian mixture model mapping and navigation engine on embedded FPGA. In *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 60–68, 2021.

....

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Thomas Mattsson	1640	trmatts@sandia.gov
Luke N. Shulenburger	1641	lshulen@sandia.gov
Thomas A. Gardiner	1641	tagardi@sandia.gov
Matthew R. Martin	1641	mrmarti@sandia.gov
Nathaniel D. Hamlin	1641	ndhamli@sandia.gov
Kristian Beckwith	1684	kbeckwi@sandia.gov
Stephen D. Bond	1442	sdbond@sandia.gov
Eric C. Cyr	1442	eccyr@sandia.gov
Nathan V. Roberts	1442	nvrober@sandia.gov
Steven W. Bova	1443	swbova@sandia.gov
Allen C. Robinson	1443	acrobin@sandia.gov
Thomas E. Voth	1443	tevoth@sandia.gov
M. Scott Swan	1446	mswan@sandia.gov
Amelia Henriksen	1462	aahenri@sandia.gov
William J. Rider	1544	wjrider@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.