A Projective Monte Carlo Method for Collisional Transport in Plasmas*

C. del-Castillo-Negrete†
Yale University
New Haven, CT, 06511

Particle-based Monte Carlo simulations are a powerful technique used to study collisional transport in plasmas. These methods suffer from two limitations. First, noise due to statistical sampling requires the use of a large amount of particles that can make the method very time consuming. Second, Monte Carlo methods are usually slow to converge to a steady state solution. In this study a novel projective Monte Carlo algorithm is proposed. The algorithm is based on tensor product decomposition techniques combined with projective integration to accelerate the convergence rate of a standard Monte Carlo simulation. An implementation of the proposed algorithm in Fortran is presented. This implementation was tested on a collisional transport problem of a homogeneous plasma with no electric field and the results demonstrate that the proposed projective Monte Carlo algorithm has the potential to accelerate traditional Monte Carlo simulations significantly.

I. INTRODUCTION

Particle-based Monte Carlo simulations are a powerful technique to study the behavior of many types of dynamic systems in physics. One area where they have seen a wide-spread use is in the study of collisional transport in plasmas. Collisional transport problems using the Fokker-Planck model can be reduced to a system of stochastic differential equations that can be solved independently for each particle. Therefore a Monte Carlo particle simulation reduces to just solving numerically each set of stochastic differential equations for each particle. The simplicity of this method makes it a very popular technique. The catch is that in order to reduce noise due to sampling, the model must be simulated with a very large amount of particles making these methods computationally expensive. Furthermore, Monte Carlo simulations tend to exhibit slow convergence towards the desired steady state solution. In particular, reaching a statistical equilibrium state in a Monte Carlo simulation usually requires very long (compared to the typical collision time scale) integration times. These limitations hinder the effectiveness of Monte Carlo simulations.

Previous work from Ref. [2] suggests a novel method of dealing with these issues in Monte Carlo methods. In particular, they show that low-rank matrix approximation techniques, commonly used for data compression in the field of Machine Learning, can be used to reduce the noise in simulation data produced by a Monte Carlo method. In this study, a new projective Monte Carlo method (PMC) that uses a low-rank tensor decomposition from Ref. [4] combined with a projective integration techniques from Ref. [3] is proposed. The goal of this novel algorithm is to accelerate the convergence of a standard Monte Carlo method towards a steady state solution. Furthermore, a library of Fortran routines was implemented to test the PMC method. The results of applying the PMC method to a steady state solution problem in collisional transport demonstrates that the PMC method is an efficient and quick way to find steady state solutions.

The remainder of this study is laid out as follows. First, a brief summary of the collisional transport model and the set of stochastic differential equations that it reduces to is provided. Next the steps of the proposed PMC algorithm are laid out. Finally, the results of applying an implementation of the PMC algorithm on a steady state solution problem are presented.

II. THE TRANSPORT MODEL

In the context of the kinetic description, the goal of collisional transport modeling is to obtain the single particle probability density function (pdf) $f(\mathbf{r}, \mathbf{v}, t)$, such that $f d\Omega$ gives the fraction of plasma particles in a phase space volume $d\Omega$. A plasma is an ionized gas and thus, in general, one needs to determined the pdf of electrons and ions, i.e. f_e and f_i. However, to a good approximation, it is customary to assume that the ions, being much heavier than the electrons, are in thermal equilibrium and thus assume that f_i is a given Maxwellian distribution. When this assumption holds, which is the case of interest in this work, the problem reduces to the computation of the electron distribution function f_e. Throughout this paper we will restrict attention to spatially homogeneous plasmas in which case the pdf depends only on the velocity, $f(\mathbf{v}, t)$, where to simplify the notation we have dropped the subindex e.

The equation determining the dynamics of $f(\mathbf{v}, t)$ due to external forces and collisions against a fixed Maxwellian distribution of ions, is the Fokker-Planck

*Undergraduate Thesis Presented to Yale Physics Department May 2015

†Electronic address: carlos.del-castillo-negrete@yale.edu, cdelcastillo21@gmail.com

equation [1]

$$\frac{\partial f}{\partial t} = \mathcal{E}[f] + \nu_D \mathcal{L}[f] + E[f]\,. \tag{1}$$

The first term on the right hand side of Eq. (1),

$$\mathcal{E}[f] = E_0\,\hat{e}_z \cdot \nabla_{\mathbf{v}} f = E_0 \left[\cos\theta \frac{\partial f}{\partial v} - \frac{\sin\theta}{v}\frac{\partial f}{\partial \theta}\right], \tag{2}$$

determines the change of f due to a constant electric field in the z-direction, where (v, θ, ϕ) denote spherical coordinates in the velocity space. The second term on the right hand side of Eq. (1),

$$\mathcal{L}[f] = \frac{1}{2}\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial f}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 f}{\partial\theta}\right], \tag{3}$$

which is the angular part of the Laplacian in spherical coordinates, is the Lorentz collision operator and describes diffusion on the surface of a sphere with radius v. The third term on the right hand side of Eq.(1) is the energy scattering operator,

$$E[f] = \frac{1}{v^2}\frac{\partial}{\partial v}\left[v^3\left(\alpha\nu_s f + \frac{1}{2}\nu_{||} v \frac{\partial f}{\partial v}\right)\right] \tag{4}$$

where α is the constant mass ratio of the electrons and ions, and

$$\nu_D = \frac{\mathrm{erf}(v) - G(v)}{v^3} \qquad \nu_s = 2G(v)/v$$
$$\nu_{||} = 2G(v)/v^3, \tag{5}$$

are the collision frequencies, where $G(x) = \mathrm{erf}(x) - x\mathrm{erf}'(x)/2x^2$ is the Chandrasekhar function and $\mathrm{erf} = (2/\sqrt{\pi})\int_0^x e^{-y^2}dy$. In writing Eq.(1)-(5) we have used non-dimensional variables. The time and velocity variables have been non-dimensionalized using the collision frequency, $\hat{\nu}$, and the thermal velocity of the background plasma, V_{th}, respectively.

The Monte Carlo simulation of the Fokker-Planck equation in (1) is based on a close mathematical connection between diffusion-type transport models and stochastic differential equations. A stochastic differential equation (restricting attention for the time being to a one-dimensional case) is an equation of the form,

$$dx = a(x,t)dt + b(x,t)\,dW(t)\,, \tag{6}$$

where the rate of change of the variable x in time, dx, is due to a deterministic part, $a(x,t)dt$, and a stochastic, random part $b(x,t)\,dW(t)$. In this equation, $\{W(t)\}$, denotes a Wiener process which is a stochastic process such that the probability distribution of its increments, $\Delta W = W(t+\Delta t) - W(t)$, is Gaussian, $p(\Delta W) = e^{-\Delta W^2/(2\Delta t)}/\sqrt{2\pi\Delta t}$, with zero mean and variance Δt. In the simplest case, $a = V$ =constant and $b = D/2$ =constant, it can be shown that the solution of Eq. (6) for a very large set of particles converges to the solution of the diffusion equation

$$\frac{\partial f}{\partial t} = -v\frac{\partial f}{\partial x} + D\frac{\partial^2 f}{\partial x^2}\,, \tag{7}$$

where f is the probability density function of these particles, V is the mean velocity, and D is the diffusivity. This connection is exactly what is behind the analogy between the diffusion equation and the random walk problem of Brownian motion.

The generalization of the connection between Eq. (6) and (7) to higher dimensions allows to write the following set of stochastic differential equation s

$$dv = \left[\mu v\nu_s - \frac{1}{2v^2}\frac{d}{dv}\left(v^4\nu_{||}\right) + E_0\cos\theta\right]dt + v\sqrt{\nu_{||}}\,dW_v \tag{8}$$

$$d\theta = \left[\frac{\nu_D}{2}\mathrm{ctg}\theta + E_0\frac{\sin\theta}{v}\right]dt + \sqrt{\nu_D}\,dW_\theta \tag{9}$$

$$d\phi = \sqrt{\frac{\nu_D}{\sin^2\theta}}\,dW_\phi \tag{10}$$

which are mathematically equivalent to the Fokker-Planck equation in (1). Further details of the connection between the Fokker-Planck equation and stochastic differential equations can be found in Ref. [5].

Because of the symmetry of the problem with respect to the electric field along the z-axis, it is convenient to use instead off the spherical coordinates in the velocity space (v, θ, ϕ) the parallel and perpendicular (with respect to the z-axis) components of the velocity

$$v_{\|} = v\,\cos\theta\,, \qquad v_{\perp} = v\,\sin\theta\,. \tag{11}$$

In terms of these variables, the stochastic differential equation model in Eq (9) becomes

$$d\mathbf{V} = \mathbf{A}dt + \mathcal{B}d\mathbf{W} \tag{12}$$

where $d\mathbf{V} = (dv_{||}, dv_\perp)^{\mathrm{T}}$, $d\mathbf{W} = (dW_v, dW_\theta)^{\mathrm{T}}$, and

$$\mathbf{A} = \begin{pmatrix} -(\mu\nu_s + \nu_D)v_{||} + \frac{v_{||}}{2v^3}\frac{d}{dv}(v^4\nu_{||}) - E_0 \\ -\left[\mu\nu + \frac{\nu_D}{2}\left(1 - \frac{v_{||}}{v_\perp}^2\right)\right]v_\perp + \frac{v_\perp}{2v^3}\frac{d}{dv}(v^4\nu_{||}) \end{pmatrix} \tag{13}$$

$$\mathcal{B} = \begin{pmatrix} \sqrt{\nu_{||}}v_{||} & -\sqrt{\nu_D}v_\perp \\ \sqrt{\nu_{||}}v_\perp & \sqrt{\nu_D}v_{||} \end{pmatrix} \tag{14}$$

It is important to realize that even in the absence of an external electric field, this transformation is valid and valuable because it reduces the number of velocity degrees of freedom from three to two. The goal of this project is to propose, implement, and test an efficient numerical method to solve the Fokker-Planck equation in Eq. (1) using a Monte Carlo method consisting of solving the equivalent set of stochastic differential equations in (12)-(14) for a very large ensemble of particles.

III. PROJECTIVE MONTE CARLO

This section presents an overview of the steps involved in the proposed projective Monte Carlo (PMC) algorithm. As illustrated in Fig.(1), the goal of this algorithm is to accelerate the convergence towards the final equilibrium state via repeated iterations of MC integrations steps followed by projective leaps.

A. Initializing Particle Data

The assumed starting point of this method is an analytical distribution function $f_0(v_{||}, v_\perp, t = 0)$ that describes the particle density in the $(v_{||}, v_\perp)$ phase space at time $t = 0$. To begin, the PMC algorithm initializes particle data with coordinates that match the initial configuration f_0. This is done by by sampling the initial distribution function over a given grid . Note that the number of particles N_p used in the sampling directly influences the amount of statistical error introduced in the Monte Carlo method.

B. Numerical Integration and Data Recording

For each particle p_i with coordinate $(v_{||}^{(i)}, v_\perp^{(i)})$, the stochastic differential equations in Eq.(12) are integrated from $t_0 = 0$ to some final time $t_{\rm SDE}$. Since the stochastic differential equations model the specific dynamics of each particle in velocity space as an independent random process, this task is computationally simple yet can take a very long time because of the potentially large number of particle positions that need to be updated at each time step. During the microscopic integration phase of the PMC method, histogram based estimations of the particle distribution function are computed at a given number of intermediate times and are collected into a 3-D tensor. Given an $N_x \times N_y$ spatial grid with nodes located at (X_i, Y_j) where $i = 1, 2, ...N_x$ and $j = 1, 2, ...N_y$ the histogram estimation of f is the third order tensor f_{lij} giving the fraction of particles with coordinates (x, y) such that $X_i \leqslant x \leq X_{i+1}$ and $Y_i \leqslant y \leq Y_{i+1}$. This 3-D data tensor can be also thought of as a 2D + 1 tensor that describes the evolution of the 2-D distribution function $f_0(v_{||}, v_\perp, t = 0)$ in time (the +1 dimension).

C. N+1 Tensor Product Decomposition

Following the microscopic integration, an optimal basis that captures the main dynamics of the distribution function estimations that were recorded in the data tensor f_{lij} is constructed. Considering the data set as a 2D+1 dimensional tensor, the +1 dimension being time, the PMC method seeks via tensor decomposition a time independent basis that captures the main variability in the whole collection of matrices. Mathematically, the PMC method constructs a decomposition of the form

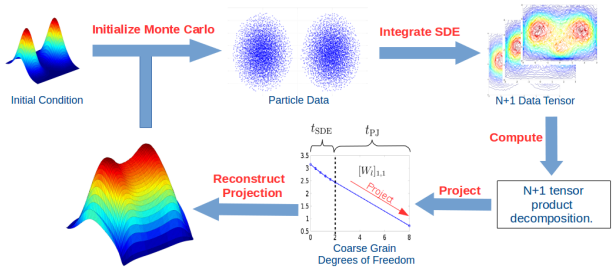

FIG. 1: Schematic diagram of the projective Monte Carlo method.

$$f_{lij}{}^{(r_1,r_2)} = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} [W_l]_{(k_1,k_2)} u_i{}^{(k_1)} v_j{}^{(k_2)} \tag{15}$$

where $u_i{}^{(k)}$ and $v_j{}^{(k)}$ are time independent orthonormal vectors, and all the time dependence is incorporated in the family of $r_1 \times r_2$ weight matrices W_l. The key issue is to use an optimal base that minimizes the reconstruction error, which in the time dependent case is defined as

$$e(r_1, r_2) = \sum_{l=1}^{N} ||F_l - U W_l V^{\rm T}||, \tag{16}$$

where $U_{ij} = u_i^{(j)}$ and $V_{ij} = v_i^{(j)}$. The construction of the optimal matrices U, V and W_l that minimize Eq.(16) was originally discussed in Ref. [4] in the context of video and image processing. Here it will be referred to as an N+1 tensor product decomposition to highlight the fact that the algorithm finds a time-independent decomposition.

D. Projecting Coarse Grained Degrees of Freedom

The key to the projective step is noticing that the set W_l of $r_1 \times r_2$ matrices define a new set of "coarse grained" degrees of freedom. Trends in the simulation data, and hence in the dynamics of the system, are represented most effectively by these degrees of freedom because they were constructed to capture maximum variability in the data set according to the reconstruction error in Eq.(16). Therefore in an effort to push the system into a future stable state, the PMC method can project the current trends exhibited by the coarse grained degrees of freedom in W_l. There are several options for a projection scheme. For simplicity the one used here is a simple linear extrapolation of the form

$$W_{k_1,k_2}(t + \Delta t) = W_{k_1,k_2}(t) + s\Delta t \tag{17}$$

applied to each coarse grained degree of freedom. In Eq.(17) Δt refers to the length of the projection and s is the slope of the line joining the last two values of W_{k_1,k_2} in the collection of matrices W_l. However by construction some elements W_{k_1,k_2} might exhibit minor fluctuations in the collection of matrices W_l and be relatively small value in comparison with other entries. Therefore a blind linear projection may artificially amplify behaviour along these coarse grained degrees of freedom from what was originally only noise in the data set. Therefore before projecting the coarse grained degrees of freedom, a filtering process is done on the coefficient matrices W_l based on an energy threshold defined as:

$$\frac{||W_{filtered}||^2}{||W||^2} \leqslant E_{th}. \tag{18}$$

$W_{filtered}$ is constructed by starting with an empty matrix and adding on the original elements of the original W matrix in decreasing order until the ratio of their norms exceeds E_{th}. Therefore the E_{th} parameter of the PMC method controls how many of the coarse grained degrees of freedom are actually projected.

E. Reconstructing and Re-initialize

Having projected along the coarse grain degrees of freedom, the PMC method can reconstruct the projected distribution function using:

$$f_{ij}{}^{(r_1,r_2)}(t = t_{SDE} + t_{PJ}) = \sum_{k_1=1}^{r_1}\sum_{k_2=1}^{r_2}[W_{pj}]_{(k_1,k_2)}u_i{}^{(k_1)}v_j{}^{(k_2)} \tag{19}$$

where W_{pj} is the weight matrix obtained from the the linear extrapolation performed in Eq.(17). Now the hope of this projective step is to push the system towards its steady state solution. In order to determine if such a steady state has been reached, the PMC method must re-initialize the standard Monte Carlo method by sampling the projected distribution function and generating a new set of particle coordinates. By then restarting the standard Monte Carlo solver, it can be determined if a steady state solution has been reached by analysing the net change in the estimations of the distribution function that are recorded in the 2D+1 data tensor. If the data still exhibits significant variability, the PMC method restarts by performing another projective step. Otherwise the PMC method has converged to a steady state.

IV. NUMERICAL RESULTS

In this section the numerical results obtained by the an implementation of the PMC algorithm are presented. The specific routines of each component of the PMC method were developed as a set of libraries in Fortran90 that build off one another. The goal in designing this library was to make the main solver routines that controlled the parameters of the method accessible and easy to use in external scripts by a general user.

A. Prolem Formulation

In order to study the performance of the PMC method, it was tested in the case where there is no external electric field in Eq.(1). The exact steady state solution of this system is known to be given by the Maxwellian distribution

$$f_\infty = f(v_{||}, v_\perp, t = 0) = \frac{2}{\sqrt{2}} v_\perp e^{-{v_\perp}^2 - {v_{||}}^2}. \tag{20}$$

In each test case, the system was initialized from the far from equilibrium "double hump" state given by the equation

$$f_0(v_{||}, v_\perp, t_0) = \frac{2}{\sqrt{2}} v_\perp e^{-(v_\perp - 2)^2}\left[e^{-(v_{||}-2)^2} + e^{-(v_{||}-2)^2}\right], \tag{21}$$

and the degree of convergence was measured by using the L_2 norm error

$$||f - f_\infty||^2 = \sum_{i,j=1} (f - f_\infty)_{i,j}{}^2, \tag{22}$$

between the estimated distribution function f and the expected Maxwellian distribution f_∞.

The parameters of the PMC method that were varied in this study are as follow:

1. The ratio of the projection time to the the total time during an iteration of the PMC method,

$$\mathrm{PJ}_{ratio} = \frac{t_{\mathrm{PJ}}}{t_{\mathrm{PJ}} + t_{\mathrm{SDE}}}, \tag{23}$$

   where t_{SDE} is the time spent integrating the stochastic differential Equations (12)-(14) and t_{PJ} is the size of the projective step.

2. The truncations ranks, (r_1, r_2), used in the N+1 tensor product decomposition used to construct the coarse grained degrees of freedom.

3. The filtering threshold, E_{th}, discussed in Eq. (18).

The parameters for the microscopic integration portion of each test case were kept fixed at: $\Delta t = 0.005$, $t_0 = 0.0$, $t_f = 40.0$, $N_p = 10^6$.

	Standard MC	Projective MC		
PJ Ratio	0.00	0.25	0.50	0.75
SMC Runtime (sec)	2391.0	1806.8	1139.9	713.85
PMC Runtime (sec)	0.00	236.29	142.39	89.03
TOTAL RUNTIME (sec)	2391.0	2043.1	1282.2	802.87
Speedup	−−	**1.170**	**1.865**	**2.978**

FIG. 2: Runtime table for PMC method using 3 different values of PJ_{ratio} and with $n_p = 10^6$, $(r_1, r_2) = (5, 5)$, $\mathrm{PJ}_{ratio} = 0.5$, $E_{th} = 0.9$ (standard Monte Carlo shown with $\mathrm{PJ}_{ratio} = 0.00$).

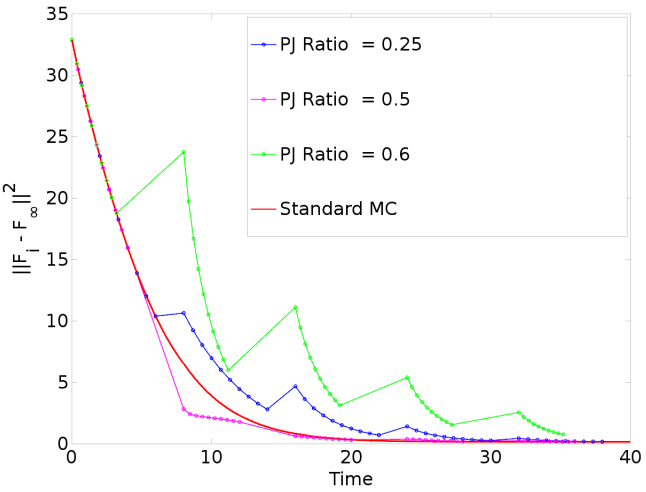

FIG. 3: Error plot for PMC method using 3 different values of PJ_{ratio} and with $n_p = 10^6$, $(r_1, r_2) = (5, 5)$, $E_{th} = 0.9$ (standard Monte Carlo shown in red). Note that all cases converge to the same final value regardless of the projection ratio used. However the quality of the projective step in pushing the solution towards convergence is highly dependent on the other parameters of the projection.

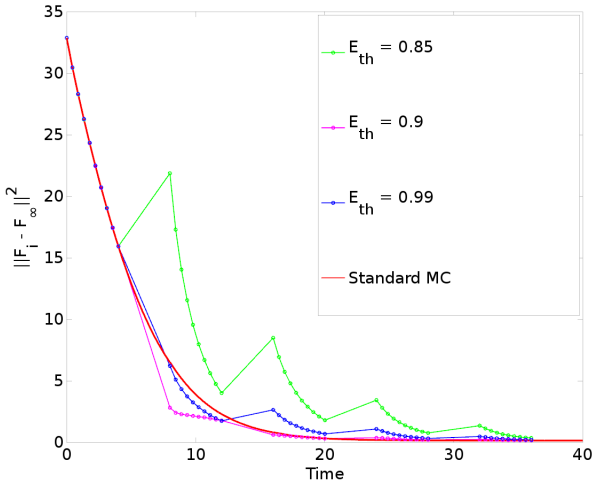

FIG. 4: Error plot for PMC method using different values of E_{th} and with $n_p = 10^6$, $(r_1, r_2) = (5, 5)$, and $\mathrm{PJ}_{ratio} = 0.5$ (standard Monte Carlo shown in red). Note there appears to be an optimal value for E_{th}. Too low of an E_{th} (green line) filters out too many of the $[W_l]_{k_1,k_2}$ terms and we loose useful information about the coarse-grain degrees of freedom during the projection step. Too high of an E_{th} (blue line) leads to not enough $[W_l]_{k_1,k_2}$ terms being filtered out, and these terms can be amplified erroneously by the projection step. An optimal E_{th} (magenta line) filters out just enough coefficients to capture all the variability of the data without amplifying noise.

B. Results

The projective Monte Carlo method was successful in converging to the same steady state distribution as the standard Monte Carlo method. Even if the projective Monte Carlo reaches this steady state after a couple of bad projective steps (see Fig.(3)), the fact that it reaches the steady state eventually is the only thing that matters. Furthermore the runtime table in Fig.(2) demonstrates that the projective Monte Carlo significantly out performs the standard Monte Carlo method. The projective step is essentially computationally free because it consists of only a linear projection. Therefore the possibilities of accelerating standard MC methods using the PMC method proposed are tremendous - the fastest PMC test case ran three times faster than the standard Monte Carlo.

Having said this, it is interesting to note that the relationship between PJ_{ratio} isn't as clear as it would seem. One would expect that projecting over smaller time intervals (that is having a lower PJ_{ratio}) would lead to more accurate projections since the state of the system will change less from the projected change over smaller time scales, and, on the other hand, projecting over longer time periods (higher PJ_{ratio}) would lead to less accurate predictions. However the relationship doesn't seem to be that linear. Holding all other parameters constant, a PJ_{ratio} of 0.5 (magenta line, Fig.(3)) produced the most efficient run while shorter and longer projection steps (blue/green lines) produced negative projection steps in each case. Nevertheless, other runs with shorter/longer projection intervals and different parameters (see for example Fig.(5)) produced high quality projections. Hence choosing the right parameters for a given projection ratio is critical for optimal performance.

Our results also indicate an intricate relationship between the parameters E_{th}, PJ_{ratio} and (r_1, r_2) of the projective method and the quality of the projection. Looking at Fig.(4) we see that there seems to be an optimal E_{th} parameter when holding all other parameters constant. This reflects the fact that there is an optimal filtering ratio that allows the projective step to project only those coarse grained degrees of freedom that are not noise and actually represent the main dynamics of the system over the projective step. Furthermore the optimal E_{th} varies depending on the length of the projective step. Comparing the quality of the projective step for $\mathrm{PJ}_{ratio} = 0.6$

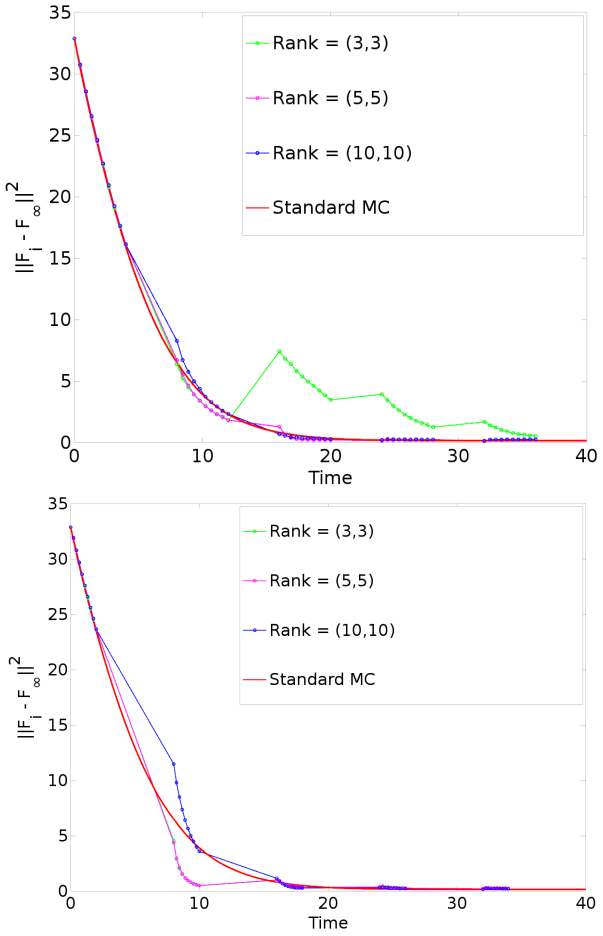

FIG. 5: Error plot for Projective Monte Carlo method using 3 different rank truncation values and a PJ_{ratio} of 0.5 (top) and 0.75 (bottom). Other parameters were set at $n_p = 10^5$ and $E_{th} = 0.99$. Note how using too low of a rank truncation may damage the quality of the projective step (green line, top).

using an E_{th} of 0.9 (see Fig.(3)) versus the quality of the projective step for $\mathrm{PJ}_{ratio} = 0.75$ using an E_{th} of 0.99 (See Fig.(5), right) , we see that the higher threshold improves the quality of the projection remarkably, even though the projective step was longer in the latter case. Clearly the extra coarse grained degrees of freedom that were not filtered out with a threshold of 0.99 but where with a threshold of 0.9 played in an important role in improving the quality of the projection over the longer time step.

The optimal E_{th} for a given problem is coupled directly with the rank truncations (r_1, r_2) used in the N+1 tensor decomposition. The rank truncation choice determines how many modes of complexity the coarse grained degrees of freedom can capture - using a high rank decomposition may be unnecessary since there may be no higher modes in the data, but using too low of a rank in the decomposition clearly can damage the quality of the projection if these higher modes exist and the tensor decomposition misses them (see Fig.(5)). Nevertheless, once an adequate choice of (r_1, r_2) has been made, choosing the right E_{th} is essentially to projecting those trends in the data that actually make significant contributions to the overall dynamics of the system.

In summary our results indicate that the PMC method proposed has huge potential for speed-up and can achieve a controlled and quick convergence with the proper tuning of the key parameters affecting the projection studied here, $E_{th}, \mathrm{PJ}_{ratio}$, and (r_1, r_2).

V. CONCLUSIONS

In this paper a novel machine learning based projective Monte Carlo method for solving problems related to collisional transport in plasmas was proposed. The concept of an N+1 tensor decomposition that seeks to find a compressed representation of dynamic state data over a preferred dimension (time) was introduced. Recognizing that this compressed representation defines a coarse-grain degrees of freedom that captures the major components of the dynamical system, a method to project these coarse-grain degrees of freedom to predict the state of the system at a later time was developed. The results of testing this projective integration scheme indicate that that the PMC method can be an effective way to accelerate traditional Monte Carlo particle simulations.

Future work will include developing a comprehensive set of heuristics to determine optimal parameters for the PMC method given a particular problem. Furthermore, the capabilities of the code to generate accurate results will be further tested by analysing different steady state problems. Mainly, in the context of collisional transport, the question becomes - what happens when the electric field is added to the simulation? The fast projective Monte Carlo solver routines developed here can provide valuable insight to the dynamics of this problem.

VI. ACKNOWLEDGMENTS

I would like to thank my senior thesis advisor at Yale University, Dr. Jun Korenaga, for his guidance throughout my project. Furthermore, I would like to thank Dr. Diego del-Castillo-Negrete for his help with this project during my visit to Oak Ridge National Laboratory in the Summer of 2014.

[1] P. Helander, and D.J. Sigmar, *Collisional Transport in Magnetized Plasmas*. (Cambridge University Press, New York, 2002.)
[2] D. del-Castillo-Negrete, D. A. Spong, and S. P. Hirshman, Phys. Plasmas **15**, 092308 (2008).
[3] I. G. Kevrekidis, C W. Gear and G. Hummer, AIChE Journal **50** 1346 -1355 (2004).
[4] Jieping Ye, Mach. Learn. **61** 167 (2005).
[5] C.W. Gardiner, *Handbook of Stochastic Methods*. (Springer-Verlag, Berlin Heidelberg, 2004).