Inverse transform sampling with Chebyshev polynomial approximation

Xin An,[1, a)] Anton Artemyev,[1] Vassilis Angelopoulos,[1] San Lu,[2] Philip Pritchett,[3] and Viktor Decyk[3]

[1)] *Department of Earth, Space and Planetary Sciences, University of California, Los Angeles, CA, 90095, USA*

[2)] *School of Earth and Space Sciences, University of Science and Technology of China, Hefei, China.*

[3)] *Department of Physics and Astronomy, University of California, Los Angeles, CA, 90095, USA*

(Dated: 4 October 2021)

`Chebsampling`, a computationally efficient, robust tool to sample general distribution functions in one and two dimensions was developed. This tool is based on inverse transform sampling with function approximation by Chebyshev polynomials. We demonstrate practical uses of `Chebsampling` through numerical examples from space plasmas.

[a)] xinan@epss.ucla.edu

I. INTRODUCTION

Non-Gaussian distribution functions are commonly observed in space plasma systems, in which the extremely low frequency of particle collisions allows velocity distributions quite different from the Maxwell (Gaussian) solutions of the Boltzmann equation[1]. Except for planetary and solar atmospheres, the entire heliosphere, a region filled with plasma of solar origin, can usually be considered as a weakly collisional medium where charged particle velocity distributions may arbitrarily deviate from the Gaussian distribution. Such velocity distributions include multicomponent distributions consisting of localized peaks in 6D phase space of velocities and coordinates, power-law distributions in which particles have a significant probability of achieving a velocity very different from the mean velocity, and distributions resulting from collisionless relaxation of plasma instabilities. Although many of these distributions may be obtained in numerical simulations, this approach is computationally expensive and quite unstable. Thus, we need an approach that allows us to set any required non-Gaussian distribution as an initial condition for further investigation of its driven and undriven dynamics. For our application, we need an algorithm to sample arbitrary, non-Gaussian distributions in either configuration or velocity space to load particles in particle-in-cell simulations.

More broadly, generating pseudo-random samples from a prescribed distribution is a procedure important to computational plasma physics as well as other branches of computational physics. Particle-in-cell simulations, Monte Carlo simulations, molecular dynamics simulations, and gravitational simulations, for example, all use certain sampling algorithms to initialize various distribution functions. One of these algorithms, inverse transform sampling, is a simple method of generating samples X from any probability distribution function (PDF) by inverting its cumulative distribution function (CDF) as $F_X^{-1}(U)$, where U is uniformly distributed on $[0, 1]$ and F_X denotes the CDF. Such sampling has been thought to be unappealing, however, because it requires either a closed form of F_X^{-1} or a complete approximation to F_X regardless of the desired sample size, it does not generalize to multiple dimensions, and it is less efficient than other approaches[2–4]. Instead, rejection sampling[2] is usually used for low-dimensional distributions, and the Metropolis-Hastings algorithm[5] is used for high-dimensional distributions.

The development of Chebyshev technology[6,7] (see `chebfun.org`) has enabled a complete

approximation to a smooth function using Chebyshev polynomials. Furthermore, approximation by Chebyshev polynomials to an analytic function converges geometrically fast. For this reason, in inverse transform sampling the CDF can be well approximated to a high precision by the Chebyshev projection and can be evaluated efficiently. The use of Chebyshev projection in sampling has been only recently explored by Olver and Townsend[8], who showed that inverse transform sampling with Chebyshev polynomial approximation is computationally efficient and robust in one dimension. They extended the approach to two dimensions but required that the CDF be well approximated by a low-rank function.

Here we develop a numerical tool to apply inverse transform sampling with Chebyshev polynomial approximation to distribution functions in one and two dimensions. In Section II, we describe the algorithm and the implementation of our numerical tool `Chebsampling`. In Section III, we demonstrate the accuracy and efficiency of our algorithm by sampling representative distribution functions (in either the configuration space or in the velocity space) in space plasmas. In Section IV, we summarize the results and discuss the pros and cons of our method.

II. METHODOLOGY

A. Inverse transform sampling

We briefly recap the inverse transform sampling method with one and two variables. In one dimension (1D), let $f(x)$ be a PDF defined on the interval $[a, b]$. Its CDF $F_X(x)$ is a strictly increasing function. To generate N samples $x_1, x_2, \cdots, x_N$ that are distributed according to $f(x)$, we invert the corresponding CDF, i.e.,

$$x_j = F_X^{-1}(u_j) \quad (j = 1, 2, \cdots, N), \tag{1}$$

where u_j is uniform on $[0, 1]$. This is inverse transform sampling. In practice, we find x_j by finding the root of $F_X(x_j) = u_j$, because the inverse transform F_X^{-1} often cannot be easily obtained. Thus, generating N samples requires solving N root-finding problems.

In two dimensions (2D), let $f(x, y)$ be a joint PDF defined on the rectangular domain $[a, b] \times [c, d]$. This joint distribution may be written as

$$f(x, y) = f_Y(y) \cdot f_{X|Y}(x|y), \tag{2}$$

where f_Y is the marginal distribution in the y direction, and $f_{X|Y}$ is the conditional distribution in the x direction for a given value of y. We do not require that $f(x, y)$ be approximated by a low-rank function as in Ref.[8], because this approximation is not always valid in our applications. Let F_Y and $F_{X|Y}$ be the CDFs of f_Y and $f_{X|Y}$, respectively. First, N_y samples $y_1, y_2, \cdots, y_{N_y}$ are generated by solving the root-finding problem

$$F_Y(y_k) = u_k, \ \ (k = 1, 2, \cdots, N_y), \tag{3}$$

where u_k is uniform on $[0, 1]$. Second, for each y_k in Equation (3), N_x samples $x_{1k}, x_{2k}, \cdots, x_{N_x k}$ are generated by finding the root for

$$F_{X|Y}(x_{jk}|y_k) = u_j, \quad (j = 1, 2, \cdots, N_x), \tag{4}$$

where u_k is uniform on $[0, 1]$. Thus, sampling of a joint 2D PDF is reduced to sampling of two 1D PDFs. As indicated by Equations (3) and (4), generation of $N_x \cdot N_y$ samples requires solving $(N_x + 1) \cdot N_y$ root-finding problems. In the special case of the separable distribution function (i.e., $f(x, y) = f_Y(y) \cdot f_X(x)$), only $N_x + N_y$ root-finding problems need to be solved to generate $N_x \cdot N_y$ samples.

B. Chebyshev polynomial approximation

The efficiency of inverse transform sampling depends on the computational cost of root finding, so we adopt the bisection method for root finding. Because the CDFs increase monotonically, this method is guaranteed to converge to a high precision. We should note that it is possible to speed up the root finding further by using a hybrid bisection combined with a Newton method (since calculating derivatives with Chebyshev polynomials in fast), but it is not necessary to do so in our application, and the performance of the bisection method is acceptable. Most of the computing time in root finding is spent on evaluation of the functions (i.e., CDFs). Fortunately, most of these functions can be accurately approximated by Chebyshev polynomials, and there are well-developed fast algorithms to evaluate them. Below we describe representation of a function by Chebyshev polynomials and rapid evaluation of this function at an arbitrary point in the domain.

Chebyshev polynomials are defined on the interval $[-1, 1]$ to which other interval $[a, b]$

can be scaled. We consider the Chebyshev points

$$x_k = \cos\left(\frac{k\pi}{n}\right) \qquad (k = 0, 1, \cdots, n), \tag{5}$$

which are extrema of the nth Chebyshev polynomial $T_n(x) = \cos(n \cdot \arccos x)$. The Chebyshev points are clustered near the two ends of the interval, -1 and 1. Unlike polynomial interpolation at equispaced points[6,9], which is associated with a well-known numerical instability (the Runge phenomenon), polynomial interpolation at the Chebyshev points is numerically stable. The Chebyshev polynomials $T_0(x)$, $T_1(x)$, $\cdots$, $T_n(x)$ on these points are orthogonal to each other[10], i.e.,

$$\sum_{k=0}^{n}{}'' T_i(x_k)T_j(x_k) = \begin{cases} 0, & (0 \leqslant i, j \leqslant n;\ i \neq j), \\ \frac{n}{2}, & (0 < i = j < n), \\ n, & (i = j = 0 \text{ or } n), \end{cases} \tag{6}$$

where the double dash in $\sum''$ denotes the first and last terms in the sum are to be halved. This discrete orthogonality property leads us to a very efficient interpolation formula.

We approximate f by the nth degree polynomial

$$p_n(x) = \sum_{k=0}^{n}{}'' c_k T_k(x), \tag{7}$$

which interpolates f at the Chebyshev points, i.e., $p_n(x_j) = f(x_j)$ with $x_j = \cos(j\pi/n)$. The interpolation coefficients c_k are given by

$$c_k = \frac{2}{n}\sum_{j=0}^{n}{}'' f(x_j)T_k(x_j) \qquad (k = 0, 1, \cdots, n). \tag{8}$$

The evaluation of c_k can be done in $\mathcal{O}(n \log n)$ operations by using the Fast Fourier Transform (FFT), which is detailed in A.

To determine the degree n of the polynomial that is sufficient to approximate f, we adopt an adaptive procedure introduced in the Chebfun software system[7]. In this procedure, we progressively select n to be $2^4 = 16$, $2^5 = 32$, $2^6 = 64$ and so on. For a given n, the f data at the $n + 1$ Chebyshev points is converted to $n + 1$ Chebyshev coefficients. If the tail of these coefficients falls below a relative level of prescribed precision, then the Chebyshev points are judged to be fine enough. We truncate the tail and keep only the non-negligible

terms. The complex engineering details of truncating a Chebyshev series are given by Ref.[11] (see the function "standardChop" in Chebfun).

Once the Chebyshev coefficients c_k have been obtained, the original data f can be discarded. These Chebyshev coefficients are then repetitively used to efficiently evaluate f (also $\int f \, \mathrm{d}x$ and $\mathrm{d}f/\mathrm{d}x$) for arbitrary points in the domain. One way of achieving this is to use the Clenshaw algorithm[12] (details of this algorithm are described in B).

C. Implementation

With the above considerations, we implemented `Chebsampling` in Fortran 90 with parallelization using MPI. The logical flows of 1D and 2D inverse transform sampling programs are summarized in Algorithms 1 and 2, respectively. Notably, our input PDF data are defined on grid, which is more flexible when an analytical expression of the input PDF is not available. For the 2D joint PDF, we apply the 1D sampling algorithm repetitively to generate samples from marginal and conditional distribution functions. As demonstrated in Section III, inverse transform sampling using the Chebyshev polynomial approximation is very efficient.

To generate a large number of samples, we parallelize the 2D inverse transform sampling algorithm. First, N_{py} samples of y are drawn from the marginal distribution function $f_Y(y)$, which is executed on all processors. Second, the tasks of sampling the conditional distribution function $f_{X|Y}(x|y)$ are evenly divided among processors based on the y samples, such that the load is balanced on each processor. The x samples drawn from the conditional distribution function are stored in local memory. This parallelization scheme yields nearly ideal scaling of the computational cost against the number of processors (see performance tests in Section III).

Algorithm 1: 1D inverse transform sampling using the Chebyshev polynomial approximation.

Input: PDF data $f(x_j)$ defined on a 1D uniform grid ($j = 1, 2, \cdots, N_g$) and the desired number of samples N_{samples}.

Output: Samples x_m ($m = 1, 2, \cdots, N_{\text{samples}}$) distributed according to $f(x)$.

- Calculate the cumulative sum of $f(x_j)$ using the recursive relation $F(x_j) = F(x_{j-1}) + \left(f(x_{j-1}) + f(x_j)\right)/2$ where $F(x_1) = 0$ and $j = 2, 3, \cdots, N_g$;
- Normalize the cumulative sum as $F(x_j) = F(x_j)/F(x_{N_g})$;
- Construct the Chebyshev coefficients c_l ($l = 0, 1, \cdots, N_{\text{cutoff}}$) from data $F(x_j)$;
- Generate samples x_m by solving the root-finding problem $F(x_m) = u_m$ with the bisection method, where $F(x) = \sum_{l=0}^{N_{\text{cutoff}}} c_l T_l(x)$, $u_m = (m - 0.5)/N_{\text{samples}}$, and $m = 1, 2, \cdots, N_{\text{samples}}$.

Algorithm 2: 2D inverse transform sampling using the Chebyshev polynomial approximation.

Input: PDF data $f(x_j, y_k)$ defined on a 2D uniform grid ($j = 1, 2, \cdots, N_{gx}$; $k = 1, 2, \cdots, N_{gy}$); the desired number of samples in x direction N_{px}; the desired number of samples in y direction N_{py}.

Output: Samples (x_{mn}, y_n) ($m = 1, 2, \cdots, N_{px}$; $n = 1, 2, \cdots, N_{py}$) distributed according to $f(x, y)$.

- Calculate the marginal distribution function $f_Y(y_k) = \sum_{j=1}^{''N_{gx}} f(x_j, y_k)$;
- Draw samples y_n ($n = 1, 2, \cdots, N_{py}$) from the marginal distribution $f_Y(y)$ by performing 1D inverse transform sampling using the Chebyshev polynomial approximation;
- For each sample y_n ($n = 1, 2, \cdots, N_{py}$):
  - Construct the conditional distribution function $f_{X|Y}(x_j|y_n)$ by interpolating $f(x_j, y_k)$ ($j = 1, 2, \cdots, N_{gx}$; $k = 1, 2, \cdots, N_{gy}$) into sampled locations y_n;
  - Draw samples x_{mn} ($m = 1, 2, \cdots, N_{px}$) from the conditional distribution $f_{X|Y}(x|y)$ by performing 1D inverse transform sampling with the Chebyshev polynomial approximation.

III. NUMERICAL EXAMPLES

Below we illustrate the performance and accuracy of our algorithm by applying it to representative distribution functions in space plasmas.

A. 2D Maxwellian current sheets

We first consider the density distribution relevant to a very important plasma equilibrium, the 2D current sheet, which is believed to be formed in the solar corona and has been commonly observed in planetary magnetospheres. In their seminal paper, Lembege and Pellat[13] constructed a two-dimensional current sheet at equilibrium that resembles the planetary magnetotail configuration. In this model, the magnetic field lines in the x-z plane

are described by the vector potential $A_y(\varepsilon x, z)\mathbf{e}_y$, where $|\varepsilon| \ll 1$ indicates weak dependence of A_y on x. The vector potential is determined by Ampere's law

$$\frac{\partial^2}{\partial z^2} A_y = -4\pi \sum_\alpha q_\alpha n_0 \frac{v_{D\alpha}}{c} \exp\left(-\frac{q_\alpha \varphi}{T_{\alpha 0}} + \frac{v_{D\alpha} q_\alpha A_y}{c T_{\alpha 0}}\right), \tag{9}$$

where $\varphi(x, z)$ is the electrostatic potential, n_0 is the reference density, $v_{D\alpha}$ is the drift velocity, T_{α_0} is the temperature of current sheet particles, q_α is the charge, and c is the speed of light. The subscript $\alpha = e, i$ represents electrons and ions, respectively. Note that $\partial^2 A_y/\partial x^2$ is omitted in Equation (9), and thus the equation is precise to order ε. The current density in Equation (9) is derived by integrating the Boltzmann-type distribution in velocity space. The electrostatic potential φ is determined by the quasi-neutrality condition

$$\sum_\alpha q_\alpha n_0 \exp\left(-\frac{q_\alpha \varphi}{T_{\alpha 0}} + \frac{v_{D\alpha} q_\alpha A_y}{c T_{\alpha 0}}\right) + q_\alpha n_b \exp\left(-\frac{q_\alpha \varphi}{T_{\alpha b}}\right) = 0. \tag{10}$$

Here two populations, the current sheet population (i.e., the current-carrying one) and the background population (i.e., the non-current-carrying one), are represented by the first and the second terms, respectively. In this example, we solve Equations (9) and (10) in the rectangular domain $[-L_z/2 \leqslant z \leqslant L_z/2] \times [-L_x \leqslant x \leqslant 0]$ with the boundary condition $A_y\big|_{z=0} = \varepsilon B_0 x,\ \partial A_y/\partial z\big|_{z=0} = 0$. Here B_0 refers to the asymptotic magnetic field at $z \to \pm\infty$, and εB_0 gives the z component of the magnetic field at $z = 0$. An analytical solution of A_y and φ is not available except for the particular choice of parameters, i.e., $v_{Di}/T_{i0} = -v_{De}/T_{e0}$. To handle more general scenarios, we solve Equations (9) and (10) numerically and obtain A_y and φ on grid.

Table I shows the two sets of parameters that are used as examples below. The first set of parameters satisfies $v_{Di}/T_{i0} = -v_{De}/T_{e0}$ so the electrostatic potential is zero everywhere in the domain (a nonpolarized current sheet). The second set of parameters has the relation $|v_{Di}/T_{i0}| < |v_{De}/T_{e0}|$, and thus gives a nonzero electric field (a polarized current sheet). This plasma equilibrium is used as an initial condition for numerical simulations helpful in solving many problems related to plasma stability and dynamics in planetary magnetotails. Therefore, a critical task is to generate a 2D spatial distribution of plasma particles for a given numerical solution of scalar and vector potentials. For purposes of demonstration, we apply our method to sample the density distribution of the current sheet population,

$$n_{\alpha 0} = n_0 \exp\left(-\frac{q_\alpha \varphi}{T_{\alpha 0}} + \frac{v_{D\alpha} q_\alpha A_y}{c T_{\alpha 0}}\right). \tag{11}$$

	$\frac{v_{Di}}{v_{\rm A}}$	$\frac{v_{De}}{v_{\rm A}}$	$\frac{T_{i0}}{m_i v_{\rm A}^2}$	$\frac{T_{e0}}{m_i v_{\rm A}^2}$	$\frac{T_{ib}}{m_i v_{\rm A}^2}$	$\frac{T_{eb}}{m_i v_{\rm A}^2}$	$\frac{n_b}{n_0}$	$\frac{L_x}{d_i}$	$\frac{L_z}{d_i}$
Non-polarized	$\frac{5}{12}$	$-\frac{1}{12}$	$\frac{5}{12}$	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{1}{12}$	0.2	32	16
Polarized	$\frac{1}{3}$	$-\frac{4}{3}$	$\frac{5}{12}$	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{1}{12}$	0.2	32	16

TABLE I. Two sets of parameters for nonpolarized and polarized Lembege-Pellat current sheets. The velocities are normalized to the Alfvén velocity $v_{\rm A} = B_0/\sqrt{4\pi n_0 m_i}$, the temperatures are normalized to $m_i v_{\rm A}^2$, the densities are normalized to n_0, and the length is normalized to the ion inertial length d_i.

For the first set of parameters, the density distribution of the ion current sheet is identical to that of the electron current sheet. To sample this density distribution, we use $N_{px} = 20000$ particles in the x direction and $N_{pz} = 10000$ particles in the z direction, which gives a total of $N_{px} \cdot N_{pz} = 2 \times 10^8$ particles. Figures 1(a) through 1(c) show the excellent agreement between the ground truth density and the sampled density. The errors [$\lesssim 1\%$; Figure 1(c)] come from the low-density region and are negligible for our application (particle-in-cell simulations). This sampling takes about 628 seconds on a single processor. Using the parallelization scheme outlined in Section II, we observe that the sampling takes about 1 second on 512 processors. As shown in Figure 2, the wall-clock time used in sampling scales ideally against the number of processors.

Similarly, we generate 2×10^8 samples for the polarized current sheet. The results are shown in Figures 1(d) through 1(i). In this case, the electron current sheet [Figure 1(g)] is embedded in the ion current sheet [Figure 1(d)]. Sampling the electron current sheet is challenging because of the steep gradient at its edge. The Chebyshev projection, which is able to capture the main characteristics of the electron current sheet, gives an accurate sampled distribution [Figures 1(h)-(i)].

In Table II, we compare the performance of inverse transform sampling with rejection sampling for the distributions shown in Figure 1. For relatively fat distribution functions as in Figures 1(a) and 1(d), rejection sampling is more efficient than inverse transform sampling. For highly peaked distribution functions as in Figure 1(g), however, inverse transform sampling outperforms rejection sampling. To represent such a distribution function, inverse transform sampling must only add more Chebyshev coefficients that do not add much com-

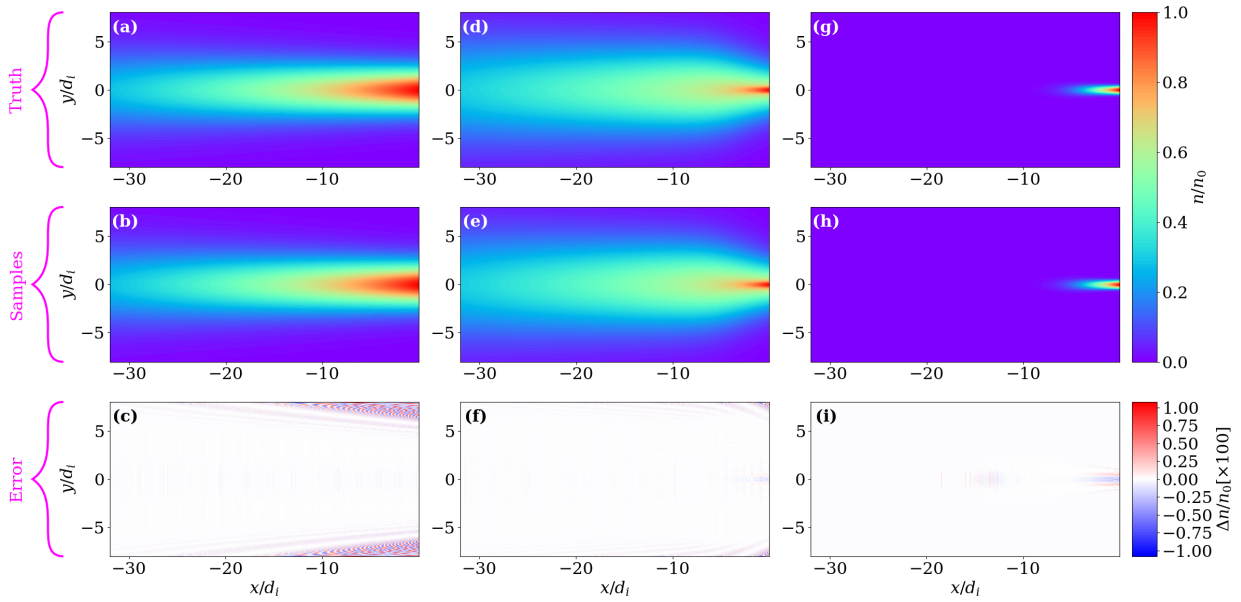

FIG. 1. Inverse transform sampling of the Lembege-Pellat current sheet. (a)-(c) The nonpolarized Lembege-Pellat current sheet set up using the first set of parameters in Table I. (d)-(i) The ion component [(d), (e), (f)] and the electron component [(g), (h), (i)] of the polarized Lembege-Pellat current sheet. This current sheet is obtained using the second set of parameters in Table I. The displayed distributions are for the current sheet population only, as shown in Equation (11). The three rows from top to bottom show the ground-truth density distributions, the sampled density distributions, and the difference between the sampled and ground-truth distributions, respectively.

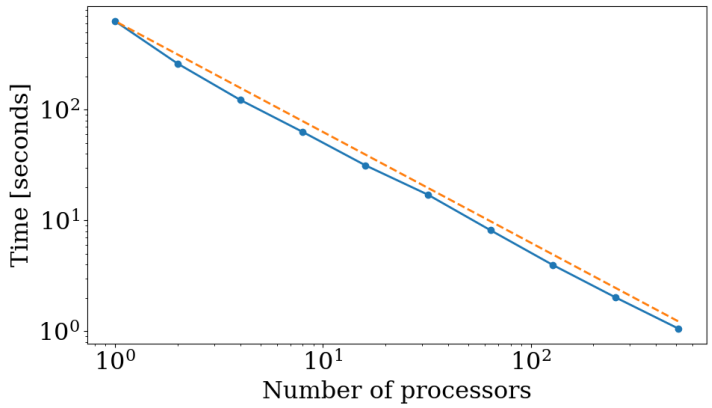

FIG. 2. The elapsed wall-clock time versus the number of processors generating samples for the nonpolarized Lembege-Pellat current sheet. The dashed line represents the ideal scaling.

	ITS [seconds]	RS [seconds]
Non-polarized current sheet	3.2	0.75
Polarized ion current sheet	3.6	0.6
Polarized electron current sheet	9.7	30.6

TABLE II. Performance comparison of inverse transform sampling (ITS) with rejection sampling (RS). In this comparison, 2×10^6 samples are generated using a single processor for each case. The current sheet distributions in the three rows correspond to Figures 1(a), 1(d), and 1(g), respectively.

putational cost, whereas rejection sampling rejects a significant fraction of samples that does add much computational cost (because the ratio of the area under the distribution function to that under the rectangular hat function is small). Therefore, inverse transform sampling avoids the practical limit in rejection sampling and gives a more consistent performance across distribution functions with vastly different shapes.

B. Non-Maxwellian velocity distributions

Furthermore, we consider three non-Maxwellian velocity distributions in the solar wind and the terrestrial magnetosphere:

1. Halo electrons in the solar wind[14]:

$$f(v_\perp, v_\parallel) = \left\{1 - \left[1 + \left(\frac{1}{2\delta}\left(\frac{v_\perp^2}{v_{c\perp}^2} + \frac{v_\parallel^2}{v_{c\parallel}^2}\right)\right)^p\right]^{-q}\right\} \times \left[1 + \frac{1}{2\kappa_h - 3}\left(\frac{v_\perp^2}{v_{h\perp}^2} + \frac{v_\parallel^2}{v_{h\parallel}^2}\right)\right]^{-\kappa_h - 1} \tag{12}$$

   with $\kappa_h = 3$, $v_{h\parallel} = v_T = 1$, $v_{h\perp} = 1/\sqrt{2}$, $v_{c\parallel} = 0.3$, $v_{c\perp} = 0.3$, $\delta = 0.9$, $p = 10$ and $q = 1$;

2. Electrons in the force-free current sheet[15]:

$$f(v_x, v_y, v_z) = \exp\left(-\beta\frac{v_x^2 + v_y^2 + v_z^2}{2}\right) \times \left[\exp\left(\beta u_y(v_y + A_y)\right) + a\cos\left(\beta u_x(v_x + A_x)\right) + b\right] \tag{13}$$

with $\beta = v_T^{-2} = 1$, $u_x = u_y = \sqrt{2}$, $A_x = A_y = 0$, $a = 1$ and $b = 2$;

3. Electrons in the injection regions in the Earth's magnetotail[16–18]:

$$f(v_\perp, v_\parallel) = \left[1 + \frac{1}{\kappa}\left(\frac{v_\perp^2}{v_{c\perp}^2} + \frac{v_\parallel^2}{v_{c\parallel}^2}\right)\right]^{-\kappa-1} \exp\left(-\frac{v_\perp^2}{2v_{h\perp}^2} - \frac{v_\parallel^2}{2v_{h\parallel}^2}\right) \tag{14}$$

with $\kappa = 0.2$, $v_{h\parallel} = v_T = 1$, $v_{h\perp} = \sqrt{2}$, $v_{c\perp} = \sqrt{3/800}v_{h\perp}$ and $v_{c\parallel} = \sqrt{2}v_{c\perp}$.

The velocity distributions in Equations (12) and (14) are uniform in gyrophase, and the velocity distribution in Equation (13) obeys a Maxwellian in the z direction that is separable from the x and y directions. Thus, these sampling problems are essentially two dimensional. Figure 3 shows the results of generating 2×10^8 samples for each of the three velocity distributions. The sampling times for these three cases are about 7–9 seconds on 64 processors. The sampled distributions capture the main trends of the original distributions. The errors are located at the high-energy tails, where the number of particles is limited. For our application in particle-in-cell simulations, such errors will not cause any problem, because the fraction of high-energy particles is very small, and thus their contribution to the charge and current deposition is small compared to the bulk of the distribution. It is noteworthy that the Chebyshev projection can fit the flat-top part of the halo electron distribution [i.e., the truncated core of the distribution with almost no electrons; see Figures 3(a) and 3(b)]. Because such flat-top distributions have also been found in the magnetic reconnection region[19] and the shock region[20], sampling them could be useful for other studies.

IV. SUMMARY AND DISCUSSION

We developed a novel tool, `Chebsampling`, for accurate, efficient sampling of distribution functions in one and two dimensions. It features the use of function approximation by Chebyshev polynomials, which accelerates root finding in the inverse transform sampling. `Chebsampling` is implemented on massively parallel computers and has the potential to be used for fully three-dimensional sampling in physical systems. The practical use of this tool is illustrated through examples from space plasma physics.

Inverse transform sampling is efficient for any distribution functions that can be numerically approximated and evaluated with low cost. The distribution function can be well approximated in one dimension by Chebyshev polynomials, and the inverse sampling method

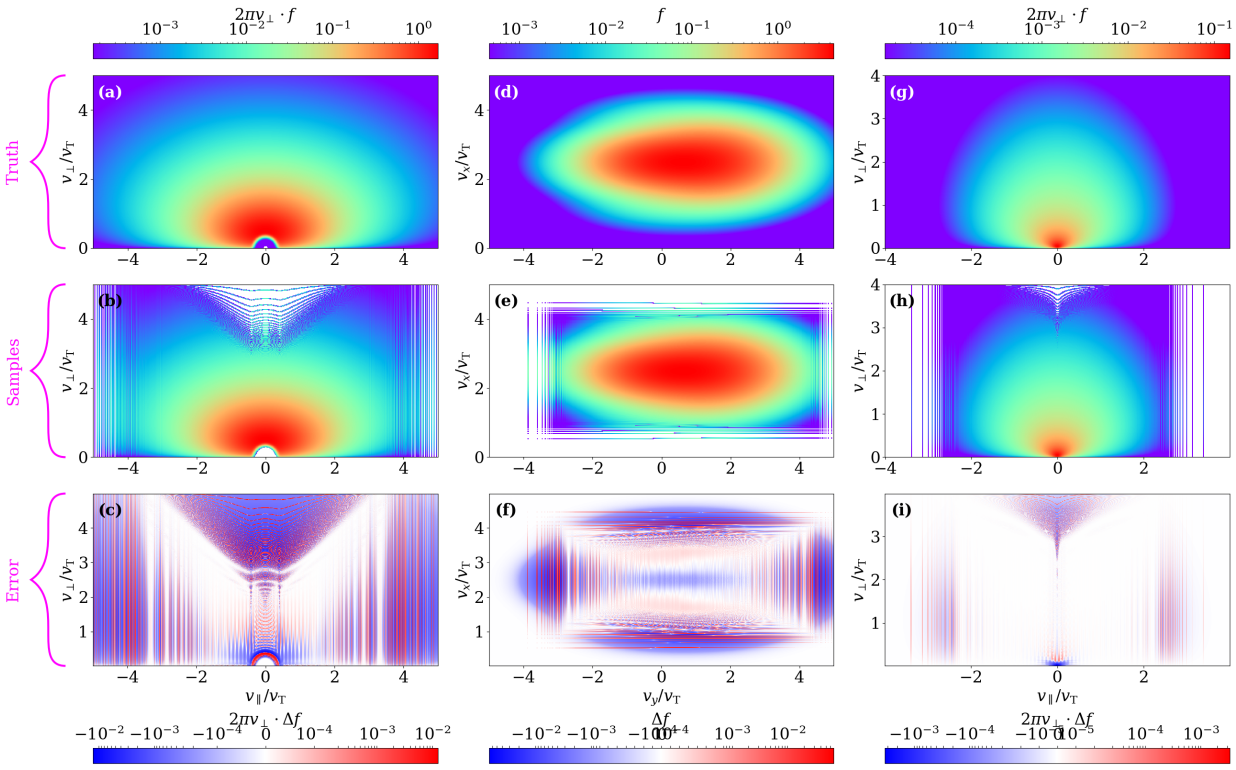

FIG. 3. Inverse transform sampling of velocity distributions in space plasmas. (a)-(c) Halo electrons in the solar wind. (d)-(f) The electron distribution in the force-free current sheet. (g)-(i) The electron distribution in the injection regions in Earth's magnetotail. The three rows from top to bottom show the ground-truth velocity distributions, the sampled velocity distributions, and the difference between the sampled and ground-truth distributions, respectively.

is practical. The sample size in two or three dimensions is relatively small and the time cost is affordable with parallelizations. With increasing sample size, however, using the inverse transform sampling in higher dimensions is challenging. Fundamental algorithmic issues on how to numerically approximate general distribution functions with two or more variables remain. Once these issues have been resolved, the inverse transform sampling method will be immediately usable in higher dimensions.

ACKNOWLEDGMENTS

`Chebsampling` is publicly available at `https://github.com/phyax/Chebsampling`. A compute capsule for verify the runs in this manuscript has been set up at `https://`

codeocean.com/capsule/0988490/tree/v1. The work was supported by NASA awards 80NSSC18K1122, 80NSSC20K1788, and 80NSSC20K0917. We would like to acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. We are grateful to J. Hohl for editing the manuscript.

Appendix A: Computation of the Chebyshev coeffcients c_k

The evaluation of the Chebyshev coefficients c_k through the use of FFT has been well established[10,21–23]. Equation (8) can be viewed as the discrete Chebyshev transform $f(x_k) \rightarrow c_k$. The connection to discrete Fourier transform can be seen through a change in variables

$$g(\theta) = f(\cos\theta), \quad \phi_k = \frac{k\pi}{n}, \quad x_k = \cos(\phi_k). \tag{A1}$$

Equation (8) can be rewritten as

$$c_k = \frac{2}{n}\sum_{j=0}^{n}{}'' \, g\left(\frac{j\pi}{n}\right)\cos\left(\frac{jk\pi}{n}\right) \qquad (k = 0, 1, \cdots, n). \tag{A2}$$

Since $\cos\theta$ and thus $g(\theta)$ are even functions of θ, we can rewrite Equation (A2)

$$c_k = \frac{1}{n}\sum_{j=-n}^{n}{}'' \, g\left(\frac{j\pi}{n}\right)\exp\left(i\frac{jk\pi}{n}\right) \qquad (k = -n, -n+1, \cdots, n). \tag{A3}$$

Furthermore, since $\cos\theta$ and thus $g(\theta)$ are 2π-periodic functions of θ, we can rewrite Equation (A3) in the form of discrete Fourier transform

$$c_k = \frac{1}{n}\sum_{j=0}^{2n-1} g\left(\frac{j\pi}{n}\right)\exp\left(i\frac{jk\pi}{n}\right) \qquad (k = 0, 1, \cdots, 2n-1). \tag{A4}$$

Appendix B: Evaluation of the Chebyshev sum

The Clenshaw algorithm is a recursive method to calculate the sum of Chebyshev polynomials. Let us consider a general sum

$$S_n(x) = \sum_{j=0}^{n} a_j P_j(x), \tag{B1}$$

where $P_j(x)$ satisfies the recurrence relation

$$P_{r+1}(x) + \alpha_r P_r(x) + \beta_r P_{r-1}(x) = 0, \tag{B2}$$

and α_r, β_r may be functions of x as well as of r.

We construct the sequence $b_n, b_{n-1}, \cdots, b_0$, where $b_{n+1} = b_{n+2} = 0$ and

$$b_r + \alpha_r b_{r+1} + \beta_{r+1} b_{r+2} = a_r, \qquad (r = n, n-1, \cdots, 0). \tag{B3}$$

By replacing a_j in Equation (B1) with the sequence $\{b_j\}$ and using the recurrence relation (B2), we obtain

$$S_n(x) = b_0 P_0(x) + b_1\{\alpha_0 P_0(x) + P_1(x)\}. \tag{B4}$$

In the case of Chebyshev polynomials, we have

$$P_r(x) = T_r(x), \quad \alpha = -2x, \quad \beta = 1. \tag{B5}$$

The recurrence relation is

$$b_r - 2x b_{r+1} + b_{r+2} = a_r, \qquad (r = n, n-1, \cdots, 0). \tag{B6}$$

The Chebyshev sum is

$$S_n(x) = \sum_{j=0}^{n} a_j T_j(x) = b_0 - b_1 x. \tag{B7}$$

REFERENCES

[1] A. Sinitsyn, E. Dulov, and V. Vedenyapin, *Kinetic Boltzmann, Vlasov and Related Equations. Imprint: Elsevier, 2011. ISBN: 978-0-12-387779-6* (Elsevier, 2011).

[2] J. E. Gentle, *Random number generation and Monte Carlo methods*, Vol. 381 (Springer, 2003).

[3] D. S. Wilks, *Statistical methods in the atmospheric sciences*, Vol. 100 (Academic press, 2011).

[4] G. H. Givens and J. A. Hoeting, *Computational Statistics*, Vol. 703 (John Wiley & Sons, 2012).

[5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," The journal of chemical physics **21**, 1087–1092 (1953).

[6] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition* (SIAM, 2019).

[7]T. A. Driscoll, N. Hale, and L. N. Trefethen, "Chebfun guide," (2014).

[8]S. Olver and A. Townsend, "Fast inverse transform sampling in one and two dimensions," arXiv preprint arXiv:1307.1223 (2013).

[9]R. B. Platte, L. N. Trefethen, and A. B. Kuijlaars, "Impossibility of fast stable approximation of analytic functions from equispaced samples," SIAM review **53**, 308–318 (2011).

[10]J. C. Mason and D. C. Handscomb, *Chebyshev polynomials* (CRC press, 2002).

[11]J. L. Aurentz and L. N. Trefethen, "Chopping a chebyshev series," ACM Transactions on Mathematical Software (TOMS) **43**, 1–21 (2017).

[12]C. W. Clenshaw, "A note on the summation of chebyshev series," Mathematics of Computation **9**, 118–120 (1955).

[13]B. Lembege and R. Pellat, "Stability of a thick two-dimensional quasineutral sheet," The Physics of Fluids **25**, 1995–2004 (1982).

[14]Š. Štverák, M. Maksimovic, P. M. Trávníček, E. Marsch, A. N. Fazakerley, and E. E. Scime, "Radial evolution of nonthermal electron populations in the low-latitude solar wind: Helios, cluster, and ulysses observations," Journal of Geophysical Research: Space Physics **114** (2009).

[15]M. G. Harrison and T. Neukirch, "One-dimensional vlasov-maxwell equilibrium for the force-free harris sheet," Physical Review Letters **102**, 135003 (2009).

[16]P. Damiano, J. Johnson, and C. Chaston, "Ion temperature effects on magnetotail alfvén wave propagation and electron energization," Journal of Geophysical Research: Space Physics **120**, 5623–5632 (2015).

[17]I. Vasko, O. Agapitov, F. Mozer, J. Bonnell, A. Artemyev, V. Krasnoselskikh, G. Reeves, and G. Hospodarsky, "Electron-acoustic solitons and double layers in the inner magnetosphere," Geophysical Research Letters **44**, 4575–4583 (2017).

[18]A. Artemyev, X.-J. Zhang, V. Angelopoulos, D. Mourenas, D. Vainchtein, Y. Shen, I. Vasko, and A. Runov, "Ionosphere feedback to electron scattering by equatorial whistler mode waves," Journal of Geophysical Research: Space Physics **125**, e2020JA028373 (2020).

[19]Y. Asano, R. Nakamura, I. Shinohara, M. Fujimoto, T. Takada, W. Baumjohann, C. Owen, A. Fazakerley, A. Runov, T. Nagai, *et al.*, "Electron flat-top distributions around the magnetic reconnection region," Journal of Geophysical Research: Space Physics **113** (2008).

[20]L. B. Wilson III, L.-J. Chen, S. Wang, S. J. Schwartz, D. L. Turner, M. L. Stevens, J. C. Kasper, A. Osmane, D. Caprioli, S. D. Bale, *et al.*, "Electron energy partition

across interplanetary shocks. i. methodology and data product," The Astrophysical Journal Supplement Series **243**, 8 (2019).

[21]N. Ahmed and P. Fisher, "Study of algorithmic properties of chebyshev coefficients," International Journal of Computer Mathematics **2**, 307–317 (1968).

[22]S. A. Orszag, "Accurate solution of the orr–sommerfeld stability equation," Journal of Fluid Mechanics **50**, 689–703 (1971).

[23]S. A. Orszag, "Galerkin approximations to flows within slabs, spheres, and cylinders," Physical Review Letters **26**, 1100 (1971).