JGR Space Physics

METHOD
10.1029/2021JA030031

**Key Points:**

- New computational tool for fast sampling of arbitrary particle distribution functions is presented
- Chebyshev polynomial interpolation allows to approximate grid-based distributions and accelerates the solution of inversion problem
- We illustrate the use of Chebsampling via sampling non-Harris current sheets and non-Maxwellian velocity distributions

**Correspondence to:**
X. An,
xinan@epss.ucla.edu

**Citation:**
An, X., Artemyev, A., Angelopoulos, V., Lu, S., Pritchett, P., & Decyk, V. (2022). Fast inverse transform sampling of non-Gaussian distribution functions in space plasmas. *Journal of Geophysical Research: Space Physics*, *127*, e2021JA030031. https://doi.org/10.1029/2021JA030031

Received 11 OCT 2021
Accepted 20 APR 2022



# Fast Inverse Transform Sampling of Non-Gaussian Distribution Functions in Space Plasmas

**Xin An[1], 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, 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, USA

**Abstract** Non-Gaussian distributions are commonly observed in collisionless space plasmas. Generating samples from non-Gaussian distributions is critical for the initialization of particle-in-cell simulations that investigate their driven and undriven dynamics. To this end, we report a computationally efficient, robust tool, `Chebsampling`, to sample general distribution functions in one and two dimensions. This tool is based on inverse transform sampling with function approximation by Chebyshev polynomials. We demonstrate practical uses of `Chebsampling` through sampling typical distribution functions in space plasmas.

## 1. 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 equilibrium solutions (Maxwellians or isotropic Gaussians) of the Boltzmann equation (Sinitsyn et al., 2011). 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 significantly 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. Sampling from these non-Gaussian distributions in either configuration or velocity space is often used to load particles in kinetic simulations for further investigation of their driven and undriven dynamics. This problem in space plasmas calls for a flexible, fast sampling algorithm that can handle any non-Gaussian distributions.

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. The application of inverse transform sampling is limited in practice, 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 (Gentle, 2003; Givens & Hoeting, 2012; Wilks, 2011). Instead, rejection sampling (e.g., Gentle, 2003) is usually used for low-dimensional distributions, and the Metropolis-Hastings algorithm (Metropolis et al., 1953) is used for high-dimensional distributions.

The development of Chebyshev technology (e.g., Driscoll et al., 2014; Trefethen, 2019; 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 (see Chapter 8 in Trefethen, 2019). 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 (Olver & Townsend, 2013), 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 report a numerical tool to apply inverse transform sampling with Chebyshev polynomial approximation to distribution functions in one and two dimensions. In Section 2, we describe the algorithm and the implementation of our numerical tool Chebsampling. In Section 3, 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 4, we summarize the results and discuss the pros and cons of our method.

## 2. Methodology

### 2.1. 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, that is,

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

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. Olver & Townsend, 2013, 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

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

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.

### 2.2. 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 (Burden et al., 2015). 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 is 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 $n$th 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 (see chapter 13 in Refs. Platte et al., 2011; Trefethen, 2019), 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 (see Section 4.6.1 in Ref. Mason & Handscomb, 2002), that is,

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

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 $n$th degree polynomial

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

which interpolates $f$ at the Chebyshev points, that is, $p_n(x_j) = f(x_j)$ 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 Appendix 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 (Driscoll et al., 2014). 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 Aurentz and Trefethen (2017) (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 (Clenshaw, 1955; details of this algorithm are described in Appendix B). To better visualize our approach described by Equations 1–8, we sketch the main idea of the fast inverse transform sampling with function approximation by Chebyshev polynomials in Figure 1.

### 2.3. 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 3, inverse transform sampling using the Chebyshev polynomial approximation is very efficient.

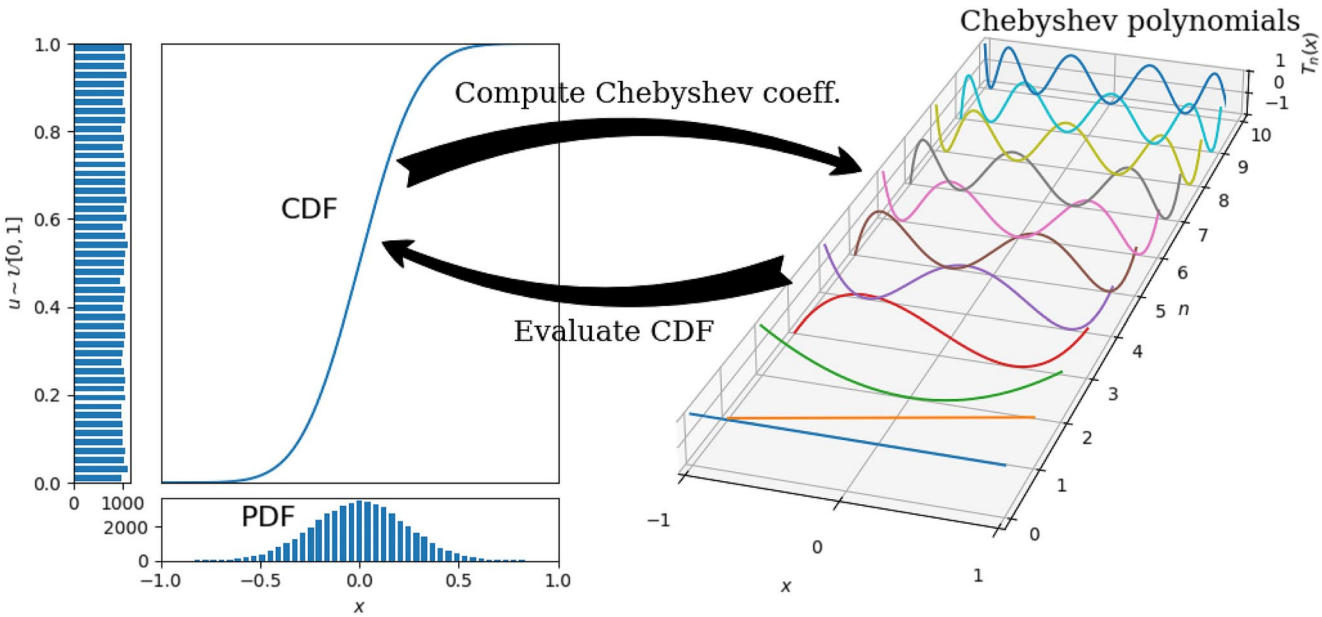


**Figure 1.** Conceptual sketch for the fast inverse transform sampling with function approximation by Chebyshev polynomials. Panels on the left of the arrows represent the inversion problem $F_X(x) = u$, where $u$ is uniformly distributed in the interval [0, 1] and $F_x$ is the cumulative distribution function (CDF). Panels on the right of the arrows represent the Chebyshev polynomials, which are used as the basis functions to interpolate and evaluate the CDF.

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 3).

## 3. Numerical Examples

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

### 3.1. 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.

**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 $N_{\text{samples}}$.

**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 $F(x_j) = F(x_{j-1}) + (f(x_{j-1}) + f(x_j))/2$ where $F(x_1) = 0$ and $j = 2, 3, \cdots, N_g$;

- Normalize the cumulative sum as $F(x_j) = F(x_j)/F\left(x_{N_g}\right)$;
- Progressively select $n = 16, 32, 64, \cdots$ as in a loop, compute the Chebyshev coefficients $c_l$ ($l = 0, 1, \cdots, n$) using Equation 8, and exit the loop if the tail of these coefficients falls below a relative level of prescribed precision;
- 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.

In their seminal paper, Lembege and Pellat (Lembege & Pellat, 1982) 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 $A_y(\varepsilon x, z)\mathrm{e}_y$, 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, $T_{\alpha 0}$ is the temperature of current sheet particles, $q_\alpha$ 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 $\varepsilon$. The current density in Equation 9 is derived by integrating the Boltzmann-type distribution in velocity space. The electrostatic potential $\varphi$ 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 $A_y|_{z=0} = \varepsilon B_0 x$, $\partial A_y/\partial z|_{z=0} = 0$. Here $B_0$ refers to the asymptotic magnetic field at $z \to \pm\infty$, and $\varepsilon B_0$ gives the $z$ component of the magnetic field at $z = 0$. An analytical solution of $A_y$ and $\varphi$ is not available except for the particular choice of parameters, that is, $v_{Di}/T_{i0} = -v_{De}/T_{e0}$. To handle more general scenarios, we solve Equations 9 and 10 numerically and obtain $A_y$ and $\varphi$ on grid.

**Table 1**
*Two Sets of Parameters for Nonpolarized and Polarized Lembege-Pellat Current Sheets*

| | $\frac{v_{Di}}{v_A}$ | $\frac{v_{De}}{v_A}$ | $\frac{T_{i0}}{m_i v_A^2}$ | $\frac{T_{e0}}{m_i v_A^2}$ | $\frac{T_{ib}}{m_i v_A^2}$ | $\frac{T_{eb}}{m_i v_A^2}$ | $\frac{n_b}{n_0}$ | $\frac{L_x}{d_i}$ | $\frac{L_z}{d_i}$ |
|---|---|---|---|---|---|---|---|---|---|
| Nonpolarized | $\frac{5}{12}$ | $-\frac{1}{12}$ | $\frac{5}{12}$ | $\frac{1}{12}$ | $\frac{5}{12}$ | $\frac{1}{12}$ | 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 |

*Note.* The velocities are normalized to the Alfvén velocity $v_A = B_0/\sqrt{4\pi n_0 m_i}$, the temperatures are normalized to $m_i v_A^2$, the densities are normalized to $n_0$, and the length is normalized to the ion inertial length $d_i$.

Table 1 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,

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

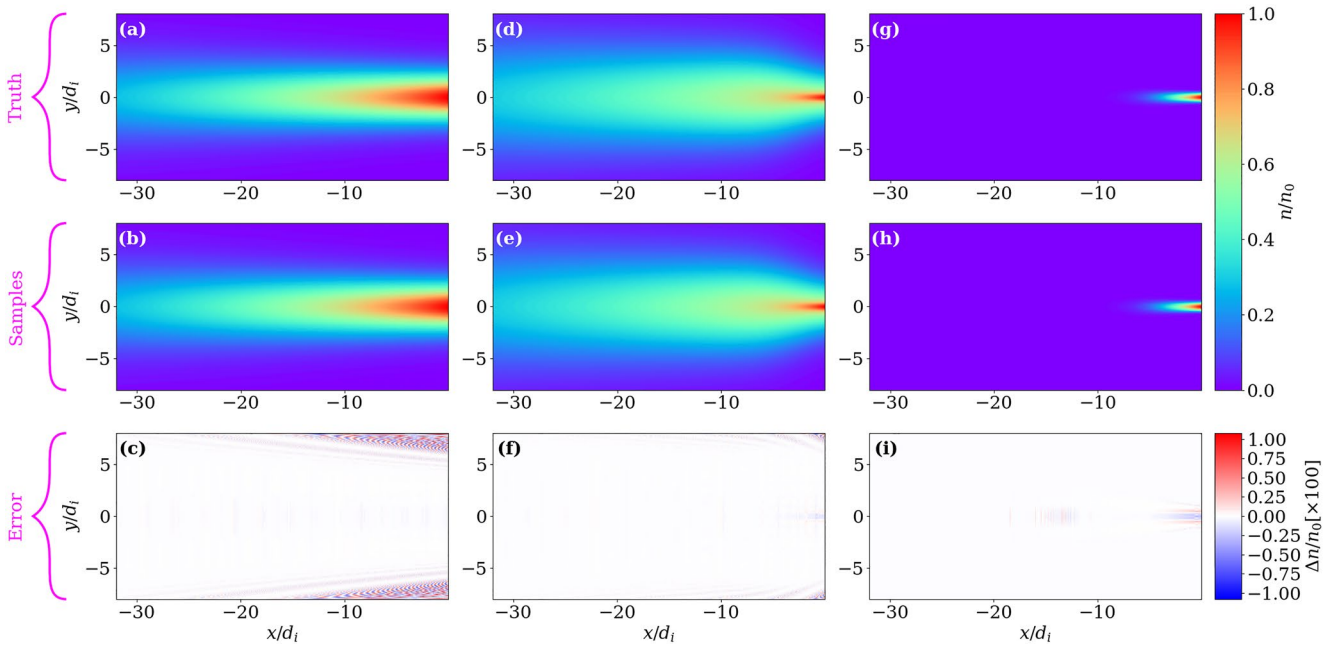


**Figure 2.** Inverse transform sampling of the Lembege-Pellat current sheet. (a–c) The nonpolarized Lembege-Pellat current sheet set up using the first set of parameters in Table 1. (d–i) The ion component (d, e, f) and the electron component (g, h, i) of the polarized Lembege-Pellat current sheet. This current sheet is obtained using the second set of parameters in Table 1. 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.

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 $N_{px} = 20{,}000$ particles in the $x$ direction and $N_{pz} = 10{,}000$ particles in the $z$ direction, which gives a total of $N_{px} \cdot N_{pz} = 2 \times 10^8$ particles. Figures 2a through 2c show the excellent agreement between the ground truth density and the sampled density. The errors [$\lesssim$1%; Figure 2c] come from the low-density region and are negligible for our application (particle-in-cell simulations). This sampling takes about 628 s on a single processor. Using the parallelization scheme outlined in Section 2, we observe that the sampling takes about 1 s on 512 processors. As shown in Figure 3, the wall-clock time used in sampling scales ideally against the number of processors.

Similarly, we generate $2 \times 10^8$ samples for the polarized current sheet. The results are shown in Figures 2d through 2i. In this case, the electron current sheet [Figure 2g] is embedded in the ion current sheet [Figure 2d]. 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 2h and 2i].

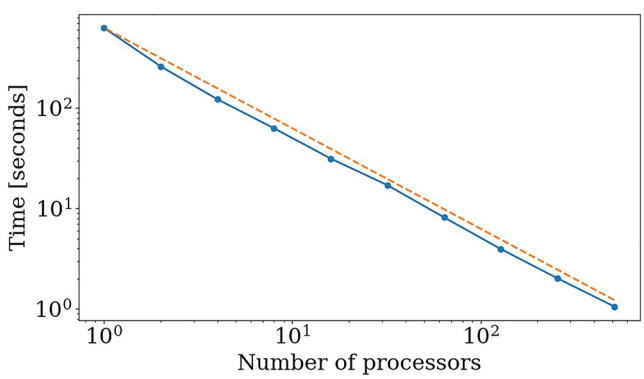


**Figure 3.** Strong scaling of `Chebsampling` for sampling the nonpolarized Lembege-Pellat current sheet. The $x$ and $y$ axes represent the number of processors and the elapsed wall-clock time, respectively. The dashed line represents the ideal scaling.

In Table 2, we compare the performance of inverse transform sampling with rejection sampling for the distributions shown in Figure 2. For relatively fat distribution functions as in Figures 2a and 2d, rejection sampling is more efficient than inverse transform sampling. For highly peaked distribution functions as in Figure 2g, 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 computational 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.

**Table 2**
*Performance Comparison of Inverse Transform Sampling (ITS) With Rejection Sampling (RS)*

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

*Note.* In this comparison, $2 \times 10^6$ samples are generated using a single processor for each case. The current sheet distributions in the three rows correspond to Figures 2a, 2d, and 2g, respectively.

### 3.2. 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 (Štverák et al., 2009):

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

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 (Harrison & Neukirch, 2009):

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

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 (Artemyev et al., 2020; Damiano et al., 2015; Vasko et al., 2017):

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

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 4 shows the results of generating $2 \times 10^8$ samples for each of the three velocity distributions. The sampling times for these three cases are about 7–9 s 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 4a and 4b]. Because such flat-top distributions have also been found in the magnetic reconnection region (Asano et al., 2008) and the shock region (Wilson III et al., 2019), sampling them could be useful for other studies.

We list the computational details in sampling the six representative distribution functions in Table 3. From each distribution, $2 \times 10^8$ samples are generated on 64 processors. For all cases, the time it takes to compute the Chebyshev coefficients ($t_{\text{cheb}}$) is about 1,000 times shorter than that of the bisection root finding ($t_{\text{bisc}}$). For distributions that have steep gradients such as in the polarized current sheet, the number of Chebyshev coefficients can be large and thus it takes longer to compute those coefficients.

## 4. 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 comput-

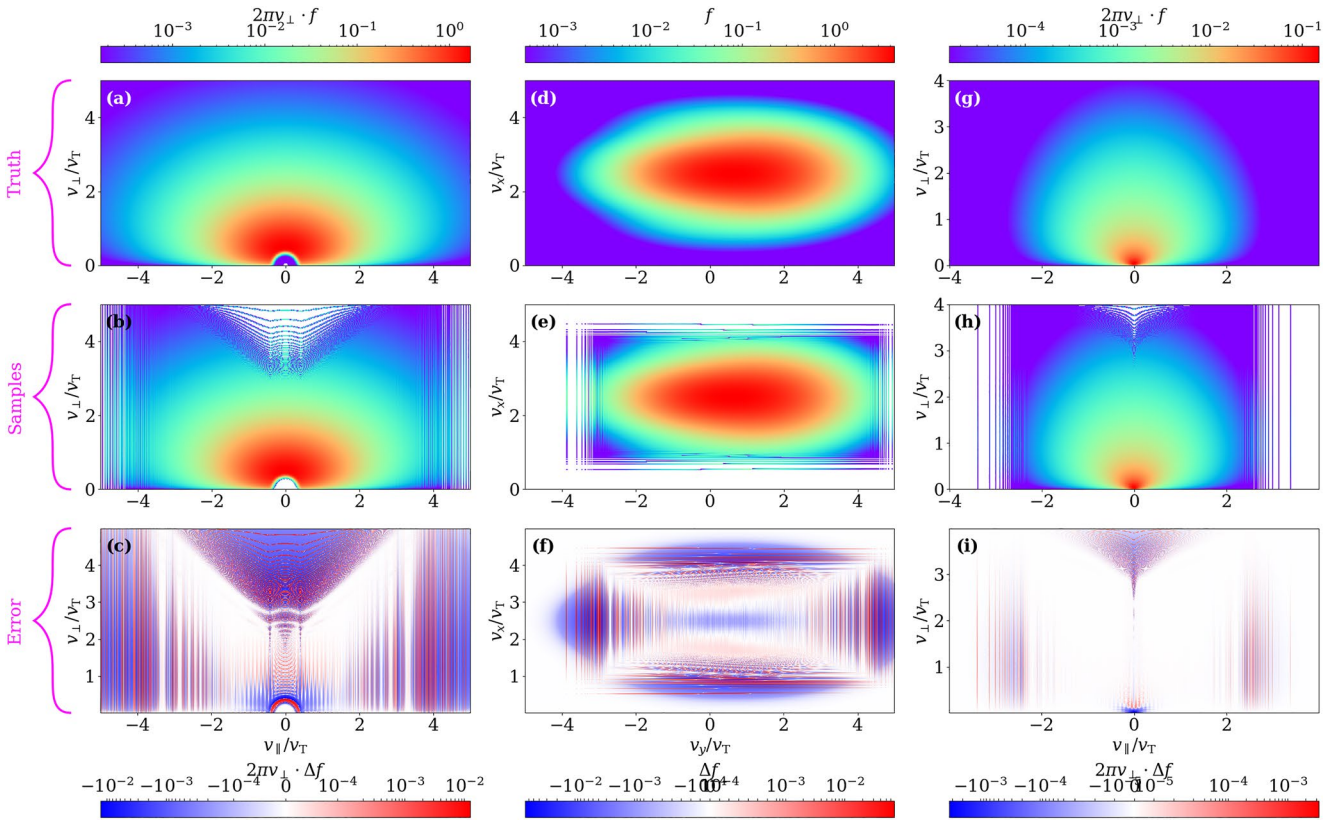


**Figure 4.** Inverse transform sampling of velocity distributions in space plasmas. (a–c) Halo electrons in the solar wind. (d–f) The electron distribution in the force-free current sheet. (g–i) The electron distribution in the injection regions in Earth's magnetotail. The three rows from top to bottom show the ground-truth velocity distributions, the sampled velocity distributions, and the difference between the sampled and ground-truth distributions, respectively. The sampling errors appearing as lines instead of random dots are caused by setting nonrandom, evenly distributed $u_k = (k - 0.5)/N_y$ (where $k = 1, 2, \cdots, N_y$) in Equation 3 [similarly for $u_j$ in Equation 4] instead of random samples uniformly distributed in the interval [0, 1], which corresponds to the "quiet start" (Birdsall & Langdon, 2018).

ers and has the potential to be used for fully three-dimensional sampling in physical systems. The practical use of this tool is illustrated through typical examples in space plasmas.

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 is practical. The sample size in two or three dimensions is

**Table 3**
*Computational Details of the Six Representative Distribution Functions*

| | $\epsilon_{cheb}$ | $N_{cutoff}$ | $t_{cheb}$ [ms] | $\epsilon_{bisc}$ | $N_{bisc}$ | $t_{bisc}$ [s] |
|---|---|---|---|---|---|---|
| Nonpolarized current sheet | $10^{-8}$ | 7–38 | 8.2 | $10^{-14}$ | 46–48 | 11.3 |
| Polarized ion current sheet | $10^{-8}$ | 16–144 | 16 | $10^{-14}$ | 46–48 | 48.1 |
| Polarized electron current sheet | $10^{-8}$ | 24–190 | 59 | $10^{-14}$ | 48 | 69.2 |
| Halo electrons in the solar wind | $10^{-8}$ | 14–360 | 6.8 | $10^{-14}$ | 47–48 | 8.5 |
| Electrons in force-free current sheets | $10^{-8}$ | 24–28 | 7.9 | $10^{-14}$ | 48 | 7.9 |
| Electrons in the injection fronts | $10^{-8}$ | 12–376 | 4.6 | $10^{-14}$ | 47–48 | 6.4 |

*Note.* $\epsilon_{cheb}$ and $\epsilon_{bisc}$ are the relative error controls in Chebyshev polynomial interpolation and bisection root finding, respectively. $N_{cutoff}$ is the number of Chebyshev coefficients to interpolate each distribution using Chebyshev polynomials. $N_{bisc}$ is the number of iteration for convergence of root finding using the bisection method. $t_{cheb}$ and $t_{bisc}$ are the time costs of Chebyshev polynomial interpolation and bisection root finding, respectively. Note that $N_{cutoff}$ is shown as a range because it varies for the marginal and conditional distributions.

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, because one needs to perform approximately the same number of inversions as the sample size. Although function approximation in two dimensions starts to emerge (Townsend & Trefethen, 2013), fundamental algorithmic issues on how to numerically approximate general distribution functions with more variables remain. Once these issues have been resolved, the inverse transform sampling method will be immediately useable in higher dimensions. Rejction sampling has a similar problem in higher dimensions. As the dimensions get larger, the ratio of the embedded volume to the total volume goes to zero. Thus a significant number of unwanted samples are rejected before a useful sample is obtained. In high dimensions, the Metropolis-Hastings algorithm is usually used, which is beyond the scope of our study.

## 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 (see Refs. Ahmed & Fisher, 1968; Mason & Handscomb, 2002; Orszag, 1971a, 1971b). 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 $\theta$, 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 $2\pi$-periodic functions of $\theta$, 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 $\alpha_r$, $\beta_r$ 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

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

The Chebyshev sum is

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

## Data Availability Statement

The code `Chebsampling` that has been developed in this manuscript is publicly available at https://doi.org/10.5281/zenodo.6109523. A compute capsule for reproducing the runs in this manuscript has been set up at https://codeocean.com/capsule/0988490/tree/v2.

**Acknowledgments**
We are grateful to J. Hohl for editing the manuscript. The work was supported by NASA awards 80NSSC18K1122, 80NSSC20K1788, and 80NSSC20K0917, and NSF award 2108582. 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.

## References

Ahmed, N., & Fisher, P. (1968). Study of algorithmic properties of Chebyshev coefficients. *International Journal of Computer Mathematics*, *2*(1–4), 307–317. https://doi.org/10.1080/00207167008803043

Artemyev, A., Zhang, X.-J., Angelopoulos, V., Mourenas, D., Vainchtein, D., Shen, Y., et al. (2020). Ionosphere feedback to electron scattering by equatorial whistler mode waves. *Journal of Geophysical Research: Space Physics*, *125*(9), e2020JA028373. https://doi.org/10.1029/2020ja028373

Asano, Y., Nakamura, R., Shinohara, I., Fujimoto, M., Takada, T., Baumjohann, W., et al. (2008). Electron flat-top distributions around the magnetic reconnection region. *Journal of Geophysical Research*, *113*(A1), A01207. https://doi.org/10.1029/2007ja012461

Aurentz, J. L., & Trefethen, L. N. (2017). Chopping a Chebyshev series. *ACM Transactions on Mathematical Software*, *43*(4), 1–21. https://doi.org/10.1145/2998442

Birdsall, C. K., & Langdon, A. B. (2018). *Plasma physics via computer simulation*. CRC press.

Burden, R. L., Faires, J. D., & Burden, A. M. (2015). *Numerical analysis*. Cengage learning.

Clenshaw, C. W. (1955). A note on the summation of Chebyshev series. *Mathematics of Computation*, *9*(51), 118–120. https://doi.org/10.1090/s0025-5718-1955-0071856-0

Damiano, P., Johnson, J., & Chaston, C. (2015). Ion temperature effects on magnetotail alfvén wave propagation and electron energization. *Journal of Geophysical Research: Space Physics*, *120*(7), 5623–5632. https://doi.org/10.1002/2015ja021074

Driscoll, T. A., Hale, N., & Trefethen, L. N. (2014). *Chebfun guide*. Pafnuty Publications.

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

Givens, G. H., & Hoeting, J. A. (2012). *Computational statistics* (Vol. 703). John Wiley & Sons.

Harrison, M. G., & Neukirch, T. (2009). One-dimensional Vlasov-mMxwell equilibrium for the force-free harris sheet. *Physical Review Letters*, *102*(13), 135003. https://doi.org/10.1103/physrevlett.102.135003

Lembege, B., & Pellat, R. (1982). Stability of a thick two-dimensional quasineutral sheet. *The Physics of Fluids*, *25*(11), 1995–2004. https://doi.org/10.1063/1.863677

Mason, J. C., & Handscomb, D. C. (2002). *Chebyshev polynomials*. CRC press.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092. https://doi.org/10.2172/4390578

Olver, S., & Townsend, A. (2013). *Fast inverse transform sampling in one and two dimensions*. arXiv preprint arXiv:1307.1223.

Orszag, S. A. (1971a). Accurate solution of the Orr–Sommerfeld stability equation. *Journal of Fluid Mechanics*, *50*(4), 689–703. https://doi.org/10.1017/s0022112071002842

Orszag, S. A. (1971b). Galerkin approximations to flows within slabs, spheres, and cylinders. *Physical Review Letters*, *26*(18), 1100–1103. https://doi.org/10.1103/physrevlett.26.1100

Platte, R. B., Trefethen, L. N., & Kuijlaars, A. B. (2011). Impossibility of fast stable approximation of analytic functions from equispaced samples. *SIAM Review*, *53*(2), 308–318. https://doi.org/10.1137/090774707

Sinitsyn, A., Dulov, E., & Vedenyapin, V. (2011). *Kinetic Boltzmann, Vlasov and related equations*. Elsevier.

Štverák, Š., Maksimovic, M., Trávníček, P. M., Marsch, E., Fazakerley, A. N., & Scime, E. E. (2009). Radial evolution of nonthermal electron populations in the low-latitude solar wind: Helios, cluster, and ulysses observations. *Journal of Geophysical Research*, *114*(A5), A05104. https://doi.org/10.1029/2008ja013883

Townsend, A., & Trefethen, L. N. (2013). An extension of chebfun to two dimensions. *SIAM Journal on Scientific Computing*, *35*(6), C495–C518. https://doi.org/10.1137/130908002

Trefethen, L. N. (2019). *Approximation theory and approximation practice* (extended edition). SIAM.

Vasko, I., Agapitov, O., Mozer, F., Bonnell, J., Artemyev, A., Krasnoselskikh, V., et al. (2017). Electron-acoustic solitons and double layers in the inner magnetosphere. *Geophysical Research Letters*, *44*(10), 4575–4583. https://doi.org/10.1002/2017gl074026

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

Wilson, L. B., III, Chen, L.-J., Wang, S., Schwartz, S. J., Turner, D. L., Stevens, M. L., et al. (2019). Electron energy partition across interplanetary shocks. i. methodology and data product. *The Astrophysical Journal - Supplement Series*, *243*(1), 8. https://doi.org/10.3847/1538-4365/ab22bd