VLASOV1m: One-dimensional electrostatic Vlasov code in MATLAB® hands-on exercises

Takayuki Umeda
Information Initiative Center, Hokkaido University

Ver.3: March 3, 2024 for public repository in GitHub
Ver.2: September 6, 2018 for ISSS-13 @ UCLA, USA
Ver.1: July 27, 2013 for ISSS-11 @ NCU, Taiwan

1 Overview of VLASOV1m

The "VLASOV1m" code was developed on the script language environment MATLAB®[1] for educational purposes (but not for scientific computing) in the hands-on session of the 11st International School for Space Simulations (ISSS-11) held at National Central University in Taiwan in July 2013. Then, the VLASOV1m code was also used in the hands-on session of the 13rd International School for Space Simulations (ISSS-13) held at University of California Los Angeles in United Stats of America in September 2018, to experience procedures of computer simulations and to study basic numerical schemes. Scientific applications of the VLASOV1m code are limited to *electrostatic problems* in one-dimensional configuration space and one-dimensional velocity space (i.e., in two-dimensional $x - v_x$ phase space), since "hyper-dimensional" (i.e., more than three dimensions in configuration plus velocity spaces) Vlasov simulations require huge computational resources [Umeda, 2012].

Since the basic concept of the Vlasov simulations is similar to that of the Particle-In-Cell (PIC) simulations, users are recommended to read the bibles of KEMPO1 (Kyoto university ElectroMagnetic Particle cOde-1D) [Omura and Matsumoto, 1993; Omura, 2007] in advance. The VLASOV1m code adopts a special Graphical User Interface (GUI) based on MATLAB® for input parameters and real-time graphical diagnostics, which is modified from the one used in the KEMPO1m code[2] [Omura, 2007] developed for the hands-on session of the 7th International School for Space Simulations (ISSS-7) held at Kyoto University in Japan in March 2007. Also, the structure of the VLASOV1m code is similar to that of the KEMPO1m code. Therefore, it would be helpful for users to perform particle simulations (especially one-dimensional electrostatic simulations) using the KEMPO1m code to be familiar with the GUI as well as the programming based on MATLAB®.

Honesty saying, there is no "standard" numerical scheme for Vlasov simulations, unlike PIC simulations. In other words, numerical schemes for Vlasov simulations are still developing. The time-advance algorithm used in the VLASOV1m code is the (operator) splitting scheme [Cheng and Knorr, 1976], which is the most popular and stable one used widely in Vlasov simulations. By

[1] https://www.mathworks.com/products/matlab.html
[2] The KEMPO1 code is free from http://space.rish.kyoto-u.ac.jp/software/.

using the operator splitting, the Vlasov equation for a particle species "s,"

$$\frac{\partial f_s}{\partial t} + v_x \frac{\partial f_s}{\partial x} + \frac{q_s}{m_s} E_x \frac{\partial f_s}{\partial v_x} = 0 \tag{1}$$

is separated into the two advection equations,

$$\frac{\partial f_s}{\partial t} + v_x \frac{\partial f_s}{\partial x} = 0 \quad \left(\frac{\mathrm{d}}{\mathrm{d}t} f_s \left[x - v_x t \right] = 0 \right) \tag{2}$$

and

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s} E_x \frac{\partial f_s}{\partial v_x} = 0 \quad \left(\frac{\mathrm{d}}{\mathrm{d}t} f_s \left[v_x - \frac{q_s}{m_s} E_x t \right] = 0 \right). \tag{3}$$

These advection equations mean that the shape (profile) of the distribution function f_s does not change along a trajectory moving with a particle. In the splitting scheme by Cheng and Knorr [1976], these two advection equations are solved alternately in time, which is consistent with the second-order leap-frog time stepping used in PIC simulations. Although a large number numerical schemes have been proposed for solving the Vlasov (advection) equations, there is no "impeccable" numerical scheme because of several numerical reasons, such as numerical dissipation (i.e., diffusion), numerical oscillation, numerical instabilities and computational costs. For educational purposes, the VLASOV1m code offers a limited number of classic and modern numerical schemes used in Vlasov simulations to study how numerical schemes affect simulation results.

2 Start up

VLASOV1m can be launched from MATLAB® command window. The detailed procedures are as follows.

- Extract (unzip) "vlasov1m_src.zip" which contains all the source files.
- Launch MATLAB® from the VLASOV1m work directory (e.g., "vlasov1m_src").
- Type "`vlasov1m`" at the command line.
  ```
  >> vlasov1m
  ```

Then, an window for initial setting of a Vlasov simulation (Figure 1) is opened.

3 Input parameter sets

A set of default parameters is automatically loaded from a file named "input_tmp.dat" (which has been installed in the VLASOV1m work directory) and is shown in the window (Figure 1). Users can modify the parameter set suitable to their own purposes via GUI shown in Figure 1. The current parameters can be saved to a specified file by the "SAVE" button (then the data save window is opened, and users can write a filename to be saved). The saved sets of parameters can be loaded from other parameter files in the work directory by the "LOAD" button at the bottom (then the data load window is opened, and users can select a corresponding file).

Note that the parameter files (e.g., "input_tmp.dat") are written in ASCII format. Hence, users can modify them via a text editor as well.

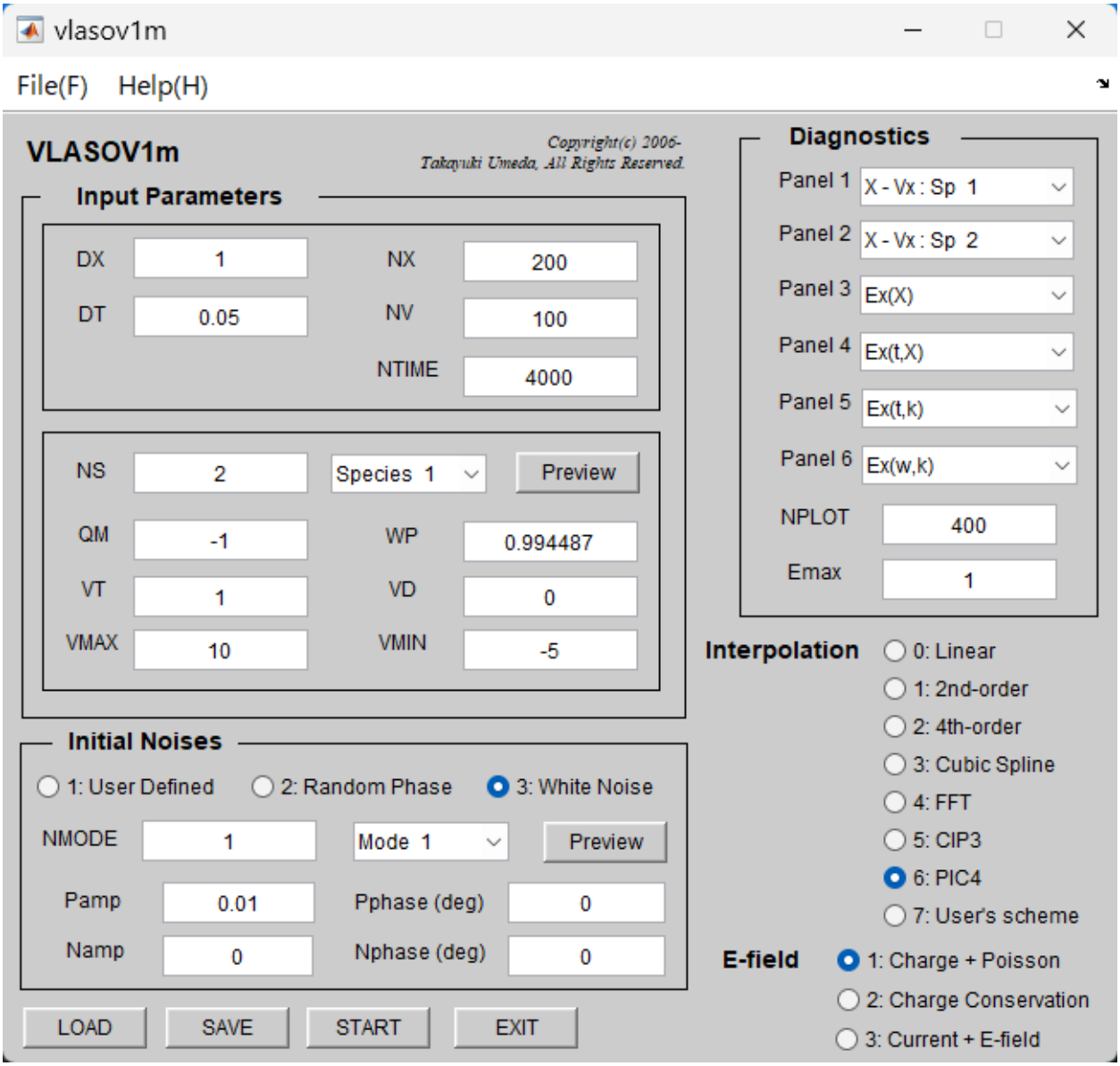

Figure 1: Window for inputing simulation data for VLASOV1m.

The parameter set consists of four parts, i.e., "Input Parameters" for setting up a simulation initial condition, "Initial Noise" for starting the simulation, "Diagnostics" for real-time graphical output, and numerical schemes, i.e., "Interpolation" for solving advection equations and "E-field" for solving electric fields. All the tables for the input parameter sets are found in the menu bar of the data input window **[Help→Parameter]**.

3.1 Input parameters

Table 1 shows the meaning of each parameter of input parameters used in the VLASOV1m code. VLASOV1m can treat multi-component plasmas by specifying "NS" as the number of species, in each of which the basic parameters (the charge to mass ratio, plasma frequency, drift/thermal velocity, maximum/minimum velocity) can be fixed independently. Users can change the species

Table 1: List of parameters for simulation set up.

DX	: Grid spacing.
DT	: Time step.
NX	: Number of grid points for the configuration space (N_x).
NV	: Number of grid points for the velocity space (N_v).
NTIME	: Number of time steps in a simulation run.
NS	: Number of particle species.
QM(s)	: Charge-to-mass ratio q_s/m_s of the species s.
WP(s)	: Plasma frequency of the species s.
VT(s)	: Thermal velocity of the species s.
VD(s)	: Drift velocity of the species s.
VMAX(s)	: Maximum velocity of the species s.
VMIN(s)	: Minimum velocity of the species s.

from "Species #" pop-up menu. For readability, given parameters of all particle species can be seen in one window by clicking "Preview" button.

In VLASOV1m, the initial velocity distribution for each species is given by the Maxwellian distribution function,

$$f_s(v_x) \equiv \frac{1}{\sqrt{2\pi V_t(s)}} \exp\left[-\frac{(V_d(s)-v_x)^2}{2V_t(s)^2}\right]. \tag{4}$$

In Vlasov simulations, the density n itself cannot be defined but the charge density qn or the mass density mn is defined. In VLASOV1m, velocity distribution functions are normalized to unity but the charge density is computed by $q_s n_s = m_s\omega_p^2/q_s$ (where $\epsilon_0 = 1$ is assumed for simplicity).

3.2 Initial noises

Vlasov simulations need a seed perturbation as initial noises. Otherwise nothing happens like fluid simulations. Table 2 shows the meaning of each parameter of initial noises used in the VLASOV1m code. VLASOV1 can treat multi-component initial noises by specifying "NMODE" as the number of wave mode for initial perturbation, in each of which the amplitude and the phase can be fixed independently. In VLASOV1m, the electric perturbation takes the following form,

$$E_x(x,t) = \sum_m \{P_{amp}(m)\cos(\omega_m t - k_m x + P_{phase}(m)) + N_{amp}(m)\cos(\omega_m t + k_m x + N_{phase}(m))\} \tag{5}$$

with $t = 0$ for initial noises. The wavenumber k_m is determined from the mode number m, i.e., $k_m = 2\pi m/N_x\Delta x$. The frequency of each mode ω_m is determined from the linear dispersion relation of Langmuir waves. Then, the initial density perturbation is determined from the Poisson's equation. Initial perturbations for other physical quantities such as the bulk velocity and the temperature for electron species ($QM(s) < 0$) are determined from the electron fluid equations. For more detail, see Appendix A.

On the other hand, initial perturbations for ion species ($QM(s) > 0$) are set as the white noise, where the amplitude of each mode is constant but the phase is random. Also, initial perturbations of the ion bulk velocity and temperature are set to be zero for simplicity, which means that initial perturbations for ion species are phase standing. Users can change the mode number from "Mode #" pop-up menu. For readability, given parameters of all modes can be seen in one window by clicking "Preview" button.

Table 2: List of input parameters for initial noises.

1: User defined	: Both amplitude and phase of each mode are defined based on the input parameters listed below.
2: Random phase	: The amplitude of each mode is defined based on the input parameters listed below, but phase of each mode is set to be uniformly random.
3: White noise	: The initial noises are set as a white noise. The amplitude of each mode is defined by "$\max(Pamp, Namp)/NX$," while the phase of each mode is set to be uniformly random.
NMODE	: Number of modes for initial noises.
Pamp(m)	: Amplitude of the wave mode number m propagating in the positive direction.
Pphase(m)	: Phase of the wave mode number m propagating in the positive direction.
Namp(m)	: Amplitude of the wave mode number m propagating in the negative direction.
Nphase(m)	: Phase of the wave mode number m propagating in the negative direction.

Table 3: List of input parameters for diagnostics.

Emax	: Maximum range for plotting electric fields, current density and charge density.
NPLOT	: Number of diagnostics to be made throughout the simulation run. In the VLASOV1m interface windows, the six different diagnostics listed below can be specified.
f(Vx)	: Averaged (reduced) velocity distribution function, $f(v_x)$.
Ex(X)	: Plot of $E_x(x)$.
Rho(X)	: Plot of $\rho(x)$.
Jx(X)	: Plot of $J_x(x)$.
Ex(k)	: Wavenumber spectrum of $E_x(k)$.
Ex(w,k)	: $\omega - k$ diagram of E_x.
Ex(t,X)	: Contour plot of $E_x(x,t)$.
Ex(t,k)	: Contour plot of $E_x(k,t)$.
Energy	: Energy history plot of kinetic and electric energy densities.
X - Vx	: $x - v_x$ phase space plot of distribution function $f_s(x, v_x)$ for the species s.

3.3 Diagnostics

After fixing the input parameter sets, click "START" button to start a simulation run. A new window is opened (Figure 2) where real-time diagnostic figure panels are made as the simulation time step advances. VLASOV1m provides various types of plots for the real-time diagnostics. Table 3 shows the meaning of each parameter of diagnostics used in VLASOV1m. Before starting the simulation run, users can select six different diagrams from the pop-up menu in the data input window (see Figure 1). Every diagram is re-drawn as the simulation time step advances, except the wave dispersion relation plot is drawn after the simulation run is completed.

Specific functions in the figure window are as follows:

– Press Space (or any other) key: pause/restart the simulation run.

– Press the zoom in/out button in the menu bar, and click the diagram: the plot area is zoom in/out.

– Press Esc key: exit the simulation run.

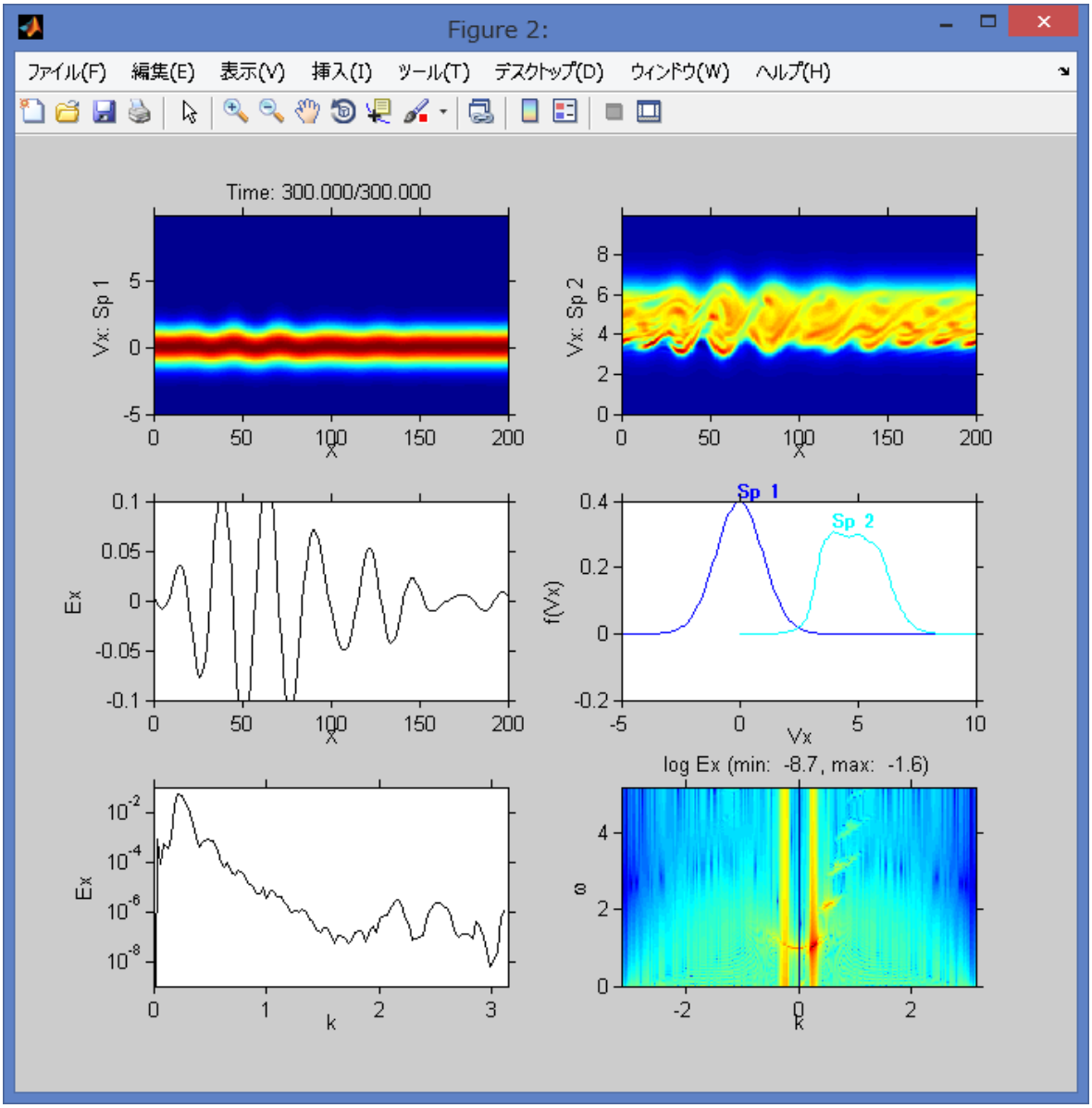

Figure 2: Window for real-time diagnostics of VLASOV1m.

It should be noted that the GUI of VLASOV1m, i.e., "vlasov1m.m," is not compatible with GNU Octave,[3] as of March 2023, which is a clone of MATLAB® for free. However, the main program of VLASOV1m, i.e., "vlasov1_main.m," is compatible with GNU Octave. The "vlasov1_main.m" program loads the input data sets from "input_tmp.dat." So, users can run VLASOV1m on GNU Octave by editing "input_tmp.dat" and by typing "`vlasov1_main`" at the command line.

```
>> vlasov1_main
```

3.4 Numerical schemes

Table 4 shows the meaning of each option for numerical schemes used in VLASOV1m.

[3] https://octave.org/

Table 4: List of input parameters for numerical schemes.

	Interpolation schemes.
0: Linear	: Linear (first-order Lagrange) interpolation.
1: 2nd-order	: Second-order Lagrange interpolation.
2: 4th-order	: Fourth-order Lagrange interpolation.
3: Cubic Spline	: Cubic spline interpolation [Cheng and Knorr, 1976].
4: FFT	: Fast Fourier transformation (spetral method) [e.g., Klimas, 1983].
5: CIP3	: Third-order constrained interpolation profile (CIP) scheme [Nakamura and Yabe, 1999].
6: PIC4	: Fourth-order central positive and non-oscillatory conservative scheme [Umeda et al., 2012].
7: User's scheme	: Numerical scheme developed by users themselves. Please modify "velocity.m" and "position.m."
	Procedures for obtained electric field.
1: Charge + Poisson	: Charge density $\rho \equiv \sum_s \int f_s \mathrm{d}v_x$ and Poisson's equation.
2: Charge Conservation	: Current density based on the charge conservation and Maxwell's equation. This option is valid ONLY with conservative schemes (0,1,2,6).
3: Current + Efield	: Current density $J_x \equiv \sum_s \int v_x f_s \mathrm{d}v_x$ and Maxwell's equation.

3.4.1 Interpolation.

Since Cheng and Knorr [1976], advection equations are solved with the "semi-Lagrangian" schemes, in which an advection equation is solved in the form of the "Lagrangian" derivative, i.e., $\frac{\mathrm{d}}{\mathrm{d}t} f(x - vt) = 0$. Physical quantities (a distribution function f in the present case) between grid cells are interpolated from the physical quantities on the neighboring grid cells. VLASOV1m provides various kinds of numerical interpolation schemes. Lagrange polynomial interpolations in the first, second and fourth degrees are provided for educational purposes. The cubic spline interpolation is a traditional scheme used since 1970's [e.g., Cheng and Knorr, 1976]. The spectral method (such as Fast Fourier Transformation: FFT) is also traditional scheme widely adopted since 1970's [e.g., Shoucri and Gagne, 1976; Klimas, 1983]. In 1990's and 2000's, the constrained interpolation profile (CIP) scheme was used for Vlasov simulations [e.g., Nakamura and Yabe, 1999] because of its high accuracy.

The time integration of advection equations (e.g., Eq.(2)) is generally performed by the following procedure,

$$f_{i,j}^{t+\Delta t} \leftarrow f_{i,j}^{t} + \delta_x v f_{i,j}. \tag{6}$$

Here, the distribution function f is descritized at $[i, j]$ space on the $x - v_x$ plane, and $\delta_x v f$ means the time integral of the spatial derivative (difference) of $v_x f_{i,j}$ from time t to $t + \Delta t$. The allow means "insert" (which commonly corresponds to "=" in programming languages). However, it is a recent trend to use "conservative" schemes for numerical interpolation in Vlasov simulations, in which Eq.(6) is rewritten in the following form

$$f_{i,j}^{t+\Delta t} \leftarrow f_{i,j}^{t} - U_{i+\frac{1}{2},j} + U_{i-\frac{1}{2},j}. \tag{7}$$

It is easily found that the total value of the distribution function, i.e., $\sum_i f_{i,j}$ is conserved during the computation. With conservative schemes, it is easy to introduce slope limiters which ensures specific properties such as positivity and monotonicity.

As an example of recent conservative schemes, the fourth-order central, positive and non-oscillatory interpolation scheme for conservation laws (PIC4) [Umeda et al., 2012] is provided. This scheme is based on the conservative form of the fourth-order central Lagrange interpolation with a special flux limiter that preserves both non-oscillatory and positivity [Umeda, 2008]. As a concept of the "non-oscillatory" scheme, a new extremum (local maximum and minimum) is not generated but the existence of an already-existing extremum is allowed. The Lagrange interpolations (of any degree) are one of conservative schemes.

Users can also introduce their own numerical schemes by editing "velocity.m" and "position.m" files.

3.4.2 Electric fields.

Another parameter for the numerical schemes is the procedure for solving electrostatic fields. The electric field is obtained by solving either Poisson's equation (8) or Maxwell's equation (9) in one dimension,

$$\frac{\partial E_x}{\partial x} = \frac{\rho}{\epsilon_0}, \tag{8}$$

$$\frac{\partial E_x}{\partial t} = -\frac{J_x}{\epsilon_0}. \tag{9}$$

VLASOV1m adopts the same staggered grid system as KEMPO1m, where the charge density (distribution function) is defined on the full-integer grids, i, while the current density and the electric field are defined on the half-integer grids, $i + \frac{1}{2}$.

The option 1: "Charge + Poisson" uses the same procedure as the electrostatic version of KEMPO1m for solving electrostatic fields. That is, compute the charge density ρ by integrating distribution functions over the velocity,

$$\rho_i^t = \sum_s q_s \Delta v_s \sum_j f_{s,i,j}^t. \tag{10}$$

Then, the electrostatic field is computed based on the spatial difference form of Poisson's equation (8),

$$\frac{E_{x,i+\frac{1}{2}}^t - E_{x,i-\frac{1}{2}}^t}{\Delta x} = \frac{\rho_i^t}{\epsilon_0}. \tag{11}$$

It should be noted that the electric field on the full-integer grids is necessary for solving the advection equation in velocity space (3). Electric fields on the full-integer grids are obtained by relocation (interpolation) from that on the half-integer grids [Omura and Matsumoto, 1993].

The option 2: "Charge Conservation" is valid only with the conservative interpolation schemes (Lagrange interpolation and PIC4 schemes in VLASOV1m). With this option, the current density is computed based on the charge conservation law,

$$\frac{\partial \rho}{\partial t} + \frac{\partial J_x}{\partial x} = 0. \tag{12}$$

The difference form of the charge conservation law (12) is easily derived by integration Eq.(7) over the velocity,

$$J_{x,i+\frac{1}{2}}^{t+\frac{\Delta t}{2}} = \sum_s q_s \Delta v_s \sum_j U_{x,s,i+\frac{1}{2},j}^{t+\frac{\Delta t}{2}}. \tag{13}$$

The electric field is computed by based on the temporal difference form of Maxwell's equation (9),

$$\frac{E^{t+\Delta t}_{x,i+\frac{1}{2}} - E^{t}_{x,i+\frac{1}{2}}}{\Delta t} = -\frac{J^{t+\frac{\Delta t}{2}}_{x,i+\frac{1}{2}}}{\epsilon_0}. \tag{14}$$

For the advection in the velocity space, then, electric fields on the full-integer grids are obtained by relocation (interpolation) from that on the half-integer grids.

The option 3: "Current + E-field" gives a good example showing what happens if the charge conservation law (12) is ignored. The current density can be also computed by

$$J^{t+\frac{\Delta t}{2}}_{x,i} = \sum_s q_s \Delta v_s \sum_j v_{x,j} f^{t+\frac{\Delta t}{2}}_{s,i,j} \tag{15}$$

Then, the electric field can be computed by

$$\frac{E^{t+\Delta t}_{x,i} - E^{t}_{x,i}}{\Delta t} = -\frac{J^{t+\frac{\Delta t}{2}}_{x,i}}{\epsilon_0}. \tag{16}$$

Here, electric fields are defined on the full-integer grid cells unlike the options 1 and 2. This procedure does not need the relocation of electric fields onto the full-integer grids. However, the current density given by Eq.(15) does not ensure the charge conservation law (12), which distorts plasma physics.

4 Exercises

4.1 Normalization and Courant condition

In any numerical simulations, all quantities can take an arbitrary unit. Users should always be aware of the ratio between simulation parameters and the physical quantities. In Vlasov1m, users can determine the simulation system size ($NX \times DX$) and spatial grid size (DX) with respect to the spatial characteristics of plasma, such as the Debye length λ_D, and the simulation total time ($NTIME \times DT$) and time step (DT) with respect to the temporal characteristics of plasma, such as plasma frequency ω_p.

In Vlasov simulations, there are two "Counrant conditions" in both configuration (x) and velocity (v_x) spaces. Plasma particles cannot move over one spatial grid size in one time step, i.e., $VMAX < DX/DT$. Plasma particles also cannot accelerate over one velocity grid size in one time step, i.e., $QM \cdot \max(E_x) < DV/DT$, where the velocity grid size is $DV \equiv (VMAX - VMIN)/NV$. It is not easy to check the second Courant condition on the velocity space, because it is impossible to predict the maximum amplitude of electric field (e.g., the saturation amplitude of plasma instabilities) from theoretical analyses.

Users are also recommended to make any modification on parameters and to take a view of any diagnostic diagrams in the following themes for exercises. As an exercise of this subsection, set a larger amplitude of initial noises in the default input parameter, and check what happens if the condition is violated.

4.2 Electrostatic linear dispersion relation

Linear dispersion relations of Langmuir (electron plasma) waves and ion acoustics waves to be confirmed. Load the sample input file "omega-k_diagram.dat" and preview input parameters for each species and initial noises.

- Change the thermal velocity for each species VT and the spatial grid size DX (and the time step DT) to check how the dispersion relation changes.

4.3 Langmuir wave propagation and Landau damping

Stable propagation of Langmuir wave at a lower mode number to be confirmed. Load the sample input file "wave_propagation.dat" and preview input parameters for each species and initial noises. Langmuir waves are one of the most fundamental electrostatic waves in plasma.

The initial perturbations of electrons (i.e., density and bulk velocity) are set based on the linear dispersion relation of Langmuir waves obtained from the electron fluid equations,

$$\omega^2 = \omega_{pe}^2 + 3k_x^2 V_{te}^2 \tag{17}$$

Confirm the stable propagation of Langmuir waves, accompanied by a small oscillation of electric fields. Note that this oscillation is due to the difference between full kinetic (nonlinear) dispersion relation based on the Vlasov equation and the linear dispersion relation based on the electron fluid equations.

- Change the mode number of initial noises and confirm the feature of wave propagation in both positive ($+x$) and negative ($-x$) directions.
- Change the interpolation scheme and check how the result is affected.
- Change the electric field solver to confirm that the "Charge Conservation" scheme gives the same result as the "Charge + Poisson" scheme. Also, check how the result is (nonphysically) modified by using the "Current + E-field" scheme.

Next, put a single-mode perturbation at a higher wave number for initial noises. Users could find that the amplitude of the wave mode decreases at a constant rate as the wave mode propagate, which is well-known as the Landau damping.

- Check that the Landau damping rate obtained from the simulation run corresponds to the analytic linear damping rate in Maxwellian electron plasma, except with the linear interpolation. On a system size of $NX = 200$, $DX = 0.5$ and the thermal velocity $VT = 1$, the wavenumber and the frequency of Mode 5, 6, 7, 8 is given as $k_x = 0.314, 0.377, 0.439, 0.503$, and $\omega = 1.139, 1.194, 1.257, 1.326$, respectively. The full Vlasov dispersion relation gives the damping rate of Mode 5, 6, 7, 8 as $\gamma = -0.017, -0.051, -0.096, -0.156$, respectively. Note that the energy diagram shows $|E_x|^2$.
- Confirm that there occurs a non-physical growth of wave energy at $t \sim 2\pi/k_x \Delta v$, which is known as the "recurrence" [Cheng and Knorr, 1976].

4.4 Parametric decay of Langmuir waves

Instability of large-amplitude Langmuir waves in the presence of ions ($NS = 2$, $|QM(1)| \neq |QM(2)|$, and $QM(1) \times QM(2) < 0$) to be confirmed. Load the sample input file "Langmuir_decay.dat" and preview input parameters for each species and initial noises.

- Change the amplitude of initial noises and confirm the stable propagation of small-amplitude Langmuir waves.
- Change the number of species to delete ion species and confirm the stable propagation of Langmuir waves when ion species is absent.

4.5 Two stream instability and electron holes

Formation of electron phase-space hole in the phase plot by the interaction between the two groups of electrons ($NS = 2$) with a relative velocity to be confirmed. Load the sample input file "two-stream_instability.dat" and preview input parameters for each species and initial noises.

- Confirm the feature of electrons trapped by the coherent electrostatic potential.
- Execute a longer run (larger $NTIME$) and confirm coalescence of potentials.
- Change the interpolation scheme and check how the result is affected.

4.6 Buneman instability

Electrostatic and current-driven instability excited by the presence of a relative drift velocity between thermal electrons and thermal ions ($NS = 2$, $|QM(1)| \neq |QM(2)|$, and $QM(1) \times QM(2) < 0$) to be confirmed. Load the sample input file "Buneman_instability.dat" and preview input parameters for each species and initial noises.

- Change the thermal velocity of ions ($VT(s)$ of $QM(s) > 0$) and check how the nonlinear evolution changes.

4.7 Harmonic Langmuir waves

Generation of multiple harmonic Langmuir waves in the $\omega - k_x$ diagram by the interaction between a high-density and low-density groups of electrons ($NS = 2$ and $WP(1) \gg WP(2)$) with a relative velocity to be confirmed. Load the sample input file "harmonic_Langmuir.dat" and preview input parameters for each species and initial noises.

- Confirm that coherent structure of electron phase-space holes is not formed but electron phase-space vortices are mixed. Note that nonlinear evolution of beam-plasma interactions with various parameters is presented by Omura et al. [1996].
- Confirm that harmonic modes are excited at $n\omega_{pe}$ [Yoon et al., 2003; Umeda et al., 2003].

5 For further studies

There are several higher-accuracy schemes [Minoshima et al., 2015; Tanaka et al., 2017], which could be implemented into VLASOV1m. The direct comparison between the higher-accuracy schemes and CIP3/PIC4 schemes could show the importance of high-accuracy schemes. Also, users are recommended to port VLASOV1m to a Fortran/C++ code for high-performance computing.

A Initial perturbations

One of the most basic plasma waves is the Langmuir waves which is a kind of the electron plasma oscillation longitudinal to the ambient magnetic field. Let us consider that physical quantities depend only on the x coordinate (along the magnetic field). Let us also consider only high frequency waves such that the ion dynamics can be neglected. The starting point is the one-dimensional linearized fluid equations for electrons,

$$\frac{\partial n_1}{\partial t} + n_0 \frac{\partial u_1}{\partial x} + u_0 \frac{\partial n_1}{\partial x} = 0, \tag{18a}$$

$$m_e n_0 \frac{\partial u_1}{\partial t} + m_e n_0 u_0 \frac{\partial u_1}{\partial x} + \frac{\partial p_1}{\partial x} + e n_0 E_1 = 0, \tag{18b}$$

$$\frac{\partial p_1}{\partial t} + \gamma p_0 \frac{\partial u_1}{\partial x} + u_0 \frac{\partial p_1}{\partial x} = 0, \tag{18c}$$

$$\frac{\partial E_1}{\partial x} + \frac{e}{\epsilon_0} n_1 = 0. \tag{18d}$$

Then, a set of thr steady-state solutions is obtained as follows,

$$E_1 = \tilde{E} \cos(\omega t \pm kx), \tag{19a}$$

$$n_1 = \mp k \frac{\epsilon_0}{e} \tilde{E} \sin(\omega t \pm kx), \tag{19b}$$

$$u_1 = -(\omega \mp k u_0) \frac{\epsilon_0}{e n_0} \tilde{E} \sin(\omega t \pm kx), \tag{19c}$$

$$p_1 = \mp k \frac{\gamma \epsilon_0 p_0}{e n_0} \tilde{E} \sin(\omega t \pm kx) = \gamma m_e v_{te}^2 n_1, \tag{19d}$$

and

$$(\omega \mp k u_0)^2 - k^2 \gamma v_{te}^2 - \omega_{pe}^2 = 0. \tag{20}$$

This equation shows the relation between ω and k, which corresponds to the well-know linear dispersion relation of Langmuir waves in a drifting electrons, i.e.,

$$\omega = \pm k u_0 + \sqrt{k^2 \gamma v_{te}^2 + \omega_{pe}^2}. \tag{21}$$

With a given amplitude of the electric field $\tilde{E}_m$ at the mode number m ($k_m = 2\pi m/(N_x \Delta x)$, with sign included), initial perturbations are imposed to the density and drift velocity,

$$\tilde{n}_{e,m} = -k_m \frac{\epsilon_0}{e} \tilde{E}_m, \tag{22a}$$

$$\tilde{V}_{d,m} = -(\omega - k_m V_d) \frac{\epsilon_0}{e n_0} \tilde{E}_m. \tag{22b}$$

References

Cheng, C. Z., and G. Knorr, The integration of the Vlasov equation in configuration space, *J. Comput. Phys.*, **22**, 330–351, 1976; https://doi.org/10.1016/0021-9991(76)90053-X.

Klimas, A. J., Numerical method based on the Fourier-Fourier transform approach for modeling 1-D electron plasma evolution, *J. Comput. Phys.*, **50**, 270–306, 1983; https://doi.org/10.1016/0021-9991(83)90067-0.

Minoshima, T., Y. Matsumoto, and T. Amano, A finite volume formulation of the multi-moment advection scheme for Vlasov simulations of magnetized plasma, *Comput. Phys. Commun.*, **187**, 137–151, 2015; https://doi.org/10.1016/j.cpc.2014.10.023.

Nakamura, T. and T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space, *Comput. Phys. Commun.*, **120**, 122–154, 1999; https://doi.org/10.1016/S0010-4655(99)00247-7.

Omura, Y., One-dimensional electromagnetic particle code: KEMPO1, A tutorial on microphysics in space plasmas, in *Advanced Methods for Space Simulations*, Edited by H. Usui and Y. Omura, pp.1–21, Terra Scientific Pub., Tokyo, 2007; The full text is free in the KEMPO1 code from http://space.rish.kyoto-u.ac.jp/software/.

Omura, Y., and H. Matsumoto, KEMPO1: Technical guide to one-dimensional electromagnetic particle code, in *Computer Space Plasma Physics: Simulation Techniques and Software*, Edited by H. Matsumoto and Y. Omura, pp.21–65, Terra Scientific Pub., Tokyo, 1993; The full text is free in the KEMPO1 code from http://space.rish.kyoto-u.ac.jp/software/.

Omura, Y., H. Matsumoto, T. Miyake, and H. Kojima, Electron beam instabilities as generation mechanism of electrostatic solitary waves in the magnetotail, *J. Geophys. Res.*, **101**, 2,685–2,697, 1996; https://doi.org/10.1029/95JA03145.

Shoucri, M. and R. R. J. Gagne, Numerical solution of the Vlasov equation by transform methods, *J. Comput. Phys.*, **21**, 238–242, 1976; https://doi.org/10.1016/0021-9991(76)90014-0.

Tanaka, S., K. Yoshikawa, T. Minoshima, and N. Yoshida, Multidimensional Vlasov-Poisson simulations with high-order monotonicity- and positivity-preserving schemes, *Astrophys. J.*, **849**, 76 (21pp), 2017; https://doi.org/10.3847/1538-4357/aa901f.

Umeda, T., A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, **60**, 773–779, 2008; https://doi.org/10.1186/BF03352826.

Umeda, T., Simulation of collisionless plasma with the Vlasov method, in *Computer Physics*, Edited by B. S. Doherty and A. N. Molloy, pp.315–332, Nova Science Pub., New York, 2012; https://researchmap.jp/multidatabases/multidatabase_contents/download/875815/98e73872d6ce4618abf7712671017af6/34741?col_no=2&frame_id=1572272.

Umeda, T., Y. Nariyuki, and D. Kariya, A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation, *Comput. Phys. Commun.*, **183**, 1094–1100, 2012; https://doi.org/10.1016/j.cpc.2012.01.011.

Umeda, T., Y. Omura, P. H. Yoon, R. Gaelzer, and H. Matsumoto, Harmonic Langmuir waves. III. Vlasov simulation, *Phys. Plasmas,* **10**, 382–391, 2003; https://doi.org/10.1063/1.1537240.

Yoon, P. H., R. Gaelzer, T. Umeda, Y. Omura, and H. Matsumoto, Harmonic Langmuir waves. I. Nonlinear dispersion relation, *Phys. Plasmas,* **10**, 364–372, 2003; https://doi.org/10.1063/1.1537238.