AGU PUBLICATIONS

JGR

Journal of Geophysical Research: Space Physics

TECHNICAL REPORTS: METHODS

10.1002/2016JA023522

LEOPARD: A grid-based dispersion relation solver for arbitrary gyrotropic distributions

Patrick Astfalk[1,2] **and Frank Jenko**[2]

[1]Max-Planck-Institut für Plasmaphysik, Garching, Germany, [2]Department of Physics and Astronomy, University of California, Los Angeles, California, USA

Key Points:
- New dispersion solver for oblique waves in multispecies plasmas with gyrotropic distribution is presented
- LEOPARD can be applied to parametric model distributions, simulation data, or spacecraft measurements
- Investigate saturation stage of resonant fire hose with distribution data from hybrid simulation

Correspondence to:
P. Astfalk,
patrick.astfalk@ipp.mpg.de

Citation:
Astfalk, P., and F. Jenko (2017), LEOPARD: A grid-based dispersion relation solver for arbitrary gyrotropic distributions, *J. Geophys. Res. Space Physics*, *122*, 89–101, doi:10.1002/2016JA023522.

Received 27 SEP 2016
Accepted 5 DEC 2016
Accepted article online 10 DEC 2016
Published online 18 JAN 2017

Abstract Particle velocity distributions measured in collisionless space plasmas often show strong deviations from idealized model distributions. Despite this observational evidence, linear wave analysis in space plasma environments such as the solar wind or Earth's magnetosphere is still mainly carried out using dispersion relation solvers based on Maxwellians or other parametric models. To enable a more realistic analysis, we present the new grid-based kinetic dispersion relation solver LEOPARD (Linear Electromagnetic Oscillations in Plasmas with Arbitrary Rotationally-symmetric Distributions) which no longer requires prescribed model distributions but allows for arbitrary gyrotropic distribution functions. In this work, we discuss the underlying numerical scheme of the code and we show a few exemplary benchmarks. Furthermore, we demonstrate a first application of LEOPARD to ion distribution data obtained from hybrid simulations. In particular, we show that in the saturation stage of the parallel fire hose instability, the deformation of the initial bi-Maxwellian distribution invalidates the use of standard dispersion relation solvers. A linear solver based on bi-Maxwellians predicts further growth even after saturation, while LEOPARD correctly indicates vanishing growth rates. We also discuss how this complies with former studies on the validity of quasilinear theory for the resonant fire hose. In the end, we briefly comment on the role of LEOPARD in directly analyzing spacecraft data, and we refer to an upcoming paper which demonstrates a first application of that kind.

1. Introduction

The rise of spacecraft measurements has given us the possibility to directly probe natural plasma environments such as Earth's magnetosphere and the solar wind, enabling a thorough examination of existing plasma models and their inherent assumptions in both linear and nonlinear regimes.

A typical problem of linear plasma physics, the propagation of small-amplitude waves in magnetized plasma is a rich and diverse field of study which has attracted a lot of attention in the past and is still a hot topic in modern plasma science. Special interest is dedicated to the stability of plasma waves. Whenever the velocity distribution of a plasma deviates from a thermal Maxwell-Boltzmann distribution there is free energy in the system which is accessed by linear eigenmodes and can lead to an exponential growth of wave amplitudes. This wave growth is not only of interest with respect to linear processes, but it is also crucial for the onset of nonlinear effects and turbulence in plasma media.

Stability analysis had been restricted to analytic magnetohydrodynamics and multifluid models until numerical tools enabled the investigation of more generally applicable models. These numerical tools reached their peak of generalization with fully kinetic hot plasma dispersion relation solvers such as WHAMP [*Roennmark*, 1982]. Being more realistic than simple fluid solvers, such kinetic eigenvalue solvers are still limited by the necessity of a prescribed model velocity distribution function. Bi-Maxwellian, loss cone, and kappa distributions—to name only a few—have proven to be useful approximations to real-world distributions [*Baumjohann and Treumann*, 1996; *Dory et al.*, 1965; *Pierrard and Lazar*, 2010]. However, spacecraft measurements have triggered a growing awareness that low collisionality in natural plasma environments can lead to strong deviations from these idealized model distributions [*Marsch*, 2006]. Since the wave dispersion in a plasma sometimes crucially depends on the presence of resonant particle populations which constitute only a tiny fraction of the overall distribution, minor changes in the resonant regime can cause a major impact on the wave dynamics by stabilizing (destabilizing) otherwise unstable (stable) eigenmodes.

A better understanding of the stability of linear waves can also trigger a deeper insight into subsequent nonlinear processes like wave-particle or wave-wave interactions. In turn, wave-particle interactions react back on the distribution function and modify the linear dispersion properties in the plasma. For example, the saturation of linear kinetic instabilities often goes hand in hand with a deformation of the particle velocity distribution [*Seough et al.*, 2014, 2015]. This feedback loop between wave dispersion and resonant plasma heating is currently only accessible with numerically demanding kinetic simulations. The deformation of the distribution function invalidates existing standard dispersion relation solvers and makes a thorough linear stability analysis hardly possible. The problem at this point is not the lack of a rigorous theoretical framework, but it is merely the lack of a proper eigenvalue solver which inhibits a general investigation of realistic velocity distribution effects on linear wave propagation so far. In this paper, we aim to fill this gap by presenting the newly developed dispersion relation solver LEOPARD (Linear Electromagnetic Oscillations in Plasmas with Arbitrary Rotationally-symmetric Distributions) which allows for any gyrotropic distribution function sampled on a two-dimensional velocity grid $(v_\parallel, v_\perp)$ and arbitrary wave propagation angles.

In section 2, we describe the theoretical and numerical foundations of the code. Section 3 presents four benchmark setups with parametric model distributions, an application of the code to simulation data, and an application to data obtained by spacecraft measurements. In section 4, we conclude with a discussion of the results and potential future applications of the code.

2. The Code

Although it is clear from observations that collisionless space plasmas such as the solar wind can easily develop and maintain strong deviations from idealized model distributions, there is—to the best of our knowledge—still no efficient numerical tool which allows for a general examination of linear wave propagation subject to realistic particle velocity distribution functions. *Dum et al.* [1980] represents a first step in this direction. They approximated measured distributions by an expansion in spherical harmonics to determine realistic wave growth. However, the analysis is rather cumbersome and restricted to parallel wave propagation only. More recently, there was a series of publications which present efficient algorithms to compute the Hilbert transform for arbitrary distributions using linear splines or B-splines of arbitrary degree [*Brambilla and Bilato*, 2009; *Micchelli et al.*, 2013; *Bilato et al.*, 2014]. Again, the presented methods only allow for an efficient computation of parallel wave propagation.

In the following, we present an approach which also allows for the analysis of obliquely propagating waves and we comment on the numerical implementation of this method in the newly developed LEOPARD code.

2.1. Linear Kinetic Theory

Starting from Maxwell's equations, we can derive the general dispersion equation for electromagnetic waves in a dielectric medium to be

$$0 = \det\left(\frac{c^2k^2}{\omega^2}\left(\frac{\mathbf{k}\otimes\mathbf{k}}{k^2} - 1\right) + \epsilon\right), \tag{1}$$

where c is the speed of light in vacuo, ω is the wave frequency, $\mathbf{k}$ is the wave vector, and ϵ is the dielectric tensor of the considered medium [see, e.g., *Brambilla*, 1998].

The dielectric tensor can be obtained by linearizing the Vlasov-Maxwell system of equations which, in general, describes the time evolution of a collisionless, multispecies plasma with distribution function f subject to electric and magnetic fields. The determination of the dielectric tensor components is a standard textbook problem and can be found, e.g, in *Brambilla* [1998]. Usually, part of the derivation is the assumption of gyrotropy, i.e.,

$$\frac{\partial f\left(v_\parallel, v_\perp, \phi\right)}{\partial \phi} = 0, \tag{2}$$

where $v_\parallel$ and $v_\perp$ are the particle velocities parallel and perpendicular to the background magnetic field, and ϕ is the gyroangle. Additionally, a further specification of the distribution is usually employed. Commonly, f is chosen to be a Maxwellian or bi-Maxwellian distribution function. Here we also adopt the restriction

to gyrotropic distributions but we refrain from specifying the distribution any further. The corresponding expressions for the tensor components are then given by

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{xx} = \tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2} + 2\pi\sum_\alpha\sum_{n=-\infty}^{\infty} n^2\mu_\alpha^3\tilde{q}_\alpha^4\tilde{n}_\alpha\frac{\tilde{\omega}}{\tilde{k}_\perp^2}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{J_n^2\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{\tilde{k}_\parallel\tilde{v}_\parallel}{\tilde{\omega}}\right)\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}+\frac{\tilde{k}_\parallel\tilde{v}_\perp}{\tilde{\omega}}\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}\right), \tag{3}$$

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{yy} = \tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2} + 2\pi\sum_\alpha\sum_{n=-\infty}^{\infty} \mu_\alpha\tilde{q}_\alpha^2\tilde{n}_\alpha\tilde{\omega}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{J_n'^2\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{\tilde{k}_\parallel\tilde{v}_\parallel}{\tilde{\omega}}\right)\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}+\frac{\tilde{k}_\parallel\tilde{v}_\perp}{\tilde{\omega}}\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}\right), \tag{4}$$

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{zz} = \tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2} + 2\pi\sum_\alpha\sum_{n=-\infty}^{\infty} \mu_\alpha\tilde{q}_\alpha^2\tilde{n}_\alpha\tilde{\omega}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{\tilde{v}_\parallel J_n^2\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\right)\tilde{v}_\perp\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}+\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\tilde{v}_\parallel\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}\right), \tag{5}$$

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{xy} = 2\pi i\sum_\alpha\sum_{n=-\infty}^{\infty} n\mu_\alpha^2\tilde{q}_\alpha^3\tilde{n}_\alpha\frac{\tilde{\omega}}{\tilde{k}_\perp}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{\tilde{v}_\perp J_n\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)J_n'\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{\tilde{k}_\parallel\tilde{v}_\parallel}{\tilde{\omega}}\right)\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}+\frac{\tilde{k}_\parallel\tilde{v}_\perp}{\tilde{\omega}}\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}\right), \tag{6}$$

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{xz} = 2\pi\sum_\alpha\sum_{n=-\infty}^{\infty} n\mu_\alpha^2\tilde{q}_\alpha^3\tilde{n}_\alpha\frac{\tilde{\omega}}{\tilde{k}_\perp}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{J_n^2\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\right)\tilde{v}_\perp\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}+\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\tilde{v}_\parallel\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}\right),$$

$$\tilde{\omega}^2\frac{v_\mathrm{A}^2}{c^2}\epsilon_{yz} = -2\pi i\sum_\alpha\sum_{n=-\infty}^{\infty} \mu_\alpha\tilde{q}_\alpha^2\tilde{n}_\alpha\tilde{\omega}\int_{-\infty}^{\infty}\mathrm{d}\tilde{v}_\parallel\int_0^{\infty}\mathrm{d}\tilde{v}_\perp\frac{\tilde{v}_\perp J_n\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)J_n'\left(\frac{\tilde{k}_\perp\tilde{v}_\perp}{\tilde{q}_\alpha\mu_\alpha}\right)}{\tilde{\omega}-\tilde{k}_\parallel\tilde{v}_\parallel-n\mu_\alpha\tilde{q}_\alpha} \times\left(\left(1-\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\right)\tilde{v}_\perp\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\parallel}+\frac{n\mu_\alpha\tilde{q}_\alpha}{\tilde{\omega}}\tilde{v}_\parallel\frac{\partial\tilde{f}_\alpha}{\partial\tilde{v}_\perp}\right), \tag{7}$$

where the velocity components $\tilde{v}_\parallel = v_\parallel/v_\mathrm{A}$ and $\tilde{v}_\perp = v_\perp/v_\mathrm{A}$ are normalized to the Alfvén velocity $v_\mathrm{A} = B_0/\sqrt{4\pi n_\mathrm{i} m_\mathrm{i}}$, the wave frequency $\tilde{\omega} = \omega/\Omega_\mathrm{i}$ is normalized to the ion gyrofrequency $\Omega_\mathrm{i} = q_\mathrm{i}B_0/(m_\mathrm{i}c)$, the wave number components $\tilde{k}_\parallel = k_\parallel d_\mathrm{i}$ and $\tilde{k}_\perp = k_\perp d_\mathrm{i}$ are normalized to the ion inertial length $d_\mathrm{i} = v_\mathrm{A}/\Omega_\mathrm{i}$, the particle species' mass $\mu_\alpha = m_\mathrm{i}/m_\alpha$ and charge $\tilde{q}_\alpha = q_\alpha/q_\mathrm{i}$ is normalized to the ion mass and ion charge, the density $\tilde{n}_\alpha = n_\alpha/n_\mathrm{i}$ is normalized to the ion density, and $\tilde{f}_\alpha = f_\alpha v_\mathrm{A}^3$ is the normalized velocity distribution function of the particle species.

Instead of describing $\tilde{f}_\alpha$ in a functional form, we rather take the distribution sampled on a two-dimensional grid $(\tilde{v}_\parallel, \tilde{v}_\perp)$ to compute the dielectric tensor elements. This lifts any restrictions imposed by idealized model distributions and allows us to examine the influence of any arbitrary gyrotropic distribution function on linear wave propagation. Naturally, this makes the computation of ϵ more cumbersome but—as we will see in the next section—the numerical demand is still tractable, even for well-resolved distribution data grids.

2.2. Numerics

As with most of the existing kinetic dispersion relation solvers, LEOPARD is essentially a root-finding algorithm which employs Muller's method to determine the solutions $\omega(\tilde{\mathbf{k}})$ of equation (1). The underlying code structure is very similar to the DSHARK code which is described in *Astfalk et al.* [2015]. The most challenging part of the numerical scheme—and therefore the only part of interest here—is the determination of the dielectric tensor components. From equation (3) it becomes apparent that each component consists of a double sum $\sum_{\alpha}\sum_{n=-\infty}^{\infty}$ over the particle species α and the Bessel index n, and a double integral over the velocity components, $\int_{-\infty}^{\infty} \mathrm{d}\tilde{v}_{\parallel} \int_{0}^{\infty} \mathrm{d}\tilde{v}_{\perp}$. Moreover, the integrals are of the form

$$\int_{-\infty}^{\infty} \mathrm{d}\tilde{v}_{\parallel} \frac{\tilde{v}_{\parallel}^{l} F\left(\tilde{v}_{\parallel}, \tilde{v}_{\perp}\right)}{\tilde{v}_{\parallel} - c_1} \tag{8}$$

and

$$\int_{0}^{\infty} \mathrm{d}\tilde{v}_{\perp} \tilde{v}_{\perp}^{m} F\left(\tilde{v}_{\parallel}, \tilde{v}_{\perp}\right) \cdot \left(J_n^2\left(c_2 \tilde{v}_{\perp}\right) \text{ or } J_n'^2\left(c_2 \tilde{v}_{\perp}\right) \text{ or } J_n\left(c_2 \tilde{v}_{\perp}\right) J_n'\left(c_2 \tilde{v}_{\perp}\right)\right), \tag{9}$$

where l and m are integers, c_1 is a complex-valued constant, c_2 is a real-valued constant, and F is a real-valued function of the velocity components.

The distribution function $\tilde{f}_{\alpha}$ is provided to the code as a two-dimensional data grid. This implies that the integral must be evaluated on the grid which can be achieved by using standard numerical integration methods. However, apart from being very demanding, this would also give rise to various sources of errors such as the highly oscillating integrand in equation (9). An alternative approach is to interpolate $F\left(\tilde{v}_{\parallel}, \tilde{v}_{\perp}\right)$ on the velocity grid using a suitable interpolation scheme which allows for solving the integrals analytically. An interpolation method which suits our needs is the cubic spline interpolation which approximates grid-based functions piecewisely using third-order polynomials connecting the data points in a smooth and continuous fashion (see Appendix A). Cubic spline interpolations are easy to implement; they give good fits for $F\left(\tilde{v}_{\parallel}, \tilde{v}_{\perp}\right)$, as long as the velocity resolution is high enough, and they transform the integrals equation (8) and equation (9) in a favorable way. Replacing $F\left(\tilde{v}_{\parallel}, \tilde{v}_{\perp}\right)$ with the corresponding spline interpolation, equation (A1), the integrals equation (8) and equation (9) become piecewise analytically solvable. Determining the integral equation (8) then becomes trivial. Equation (9) is more challenging but still can be solved analytically using hypergeometric functions (see Appendix B). After solving the integrals, we are left with a sum of analytic expressions evaluated at the velocity grid points.

Since we have to compute a double integral, the spline interpolation has to be employed two times. First, for the distribution function $\tilde{f}_{\alpha}$, say, in the parallel direction so that the integral with respect to $\tilde{v}_{\parallel}$ can be solved. Subsequently, the coefficients of the parallel spline interpolation have to be spline-interpolated again with respect to the perpendicular direction to solve the $\tilde{v}_{\perp}$ integral. Which spline interpolation comes first depends on whether $\tilde{f}_{\alpha}$ is subject to $\partial/\partial\tilde{v}_{\parallel}$ or $\partial/\partial\tilde{v}_{\perp}$ in equation (9). The chosen interpolation makes sure that the corresponding derivative can be solved analytically as well.

A strength of the presented method is that the quality of the result solely depends on the quality of the cubic spline interpolation which can be easily checked by plotting the interpolated data. This allows for a quick and intuitive error analysis which is hardly possible in any sophisticated numerical integration schemes. Furthermore, the algorithm is relatively fast. The time needed for a cubic spline interpolation on a reasonably sized grid is negligible. The most time-consuming part of the computation is the determination of the hypergeometric functions.

The LEOPARD code can process an arbitrary number of particle species. In principle, for each species the code has to be provided with a separate distribution data set. However, to increase efficiency, LEOPARD was coupled to a standard bi-Maxwellian solver using the same implementation as the DSHARK code [*Astfalk et al.*, 2015]. So if the user aims to investigate a setup which includes also bi-Maxwellian species, LEOPARD can switch to the more efficient standard algorithm for these species instead of reading the distribution from data.

3. Test Cases

The setup of the LEOPARD code allows for processing arbitrary gyrotropic distribution functions enabling a linear dispersion analysis beyond frequently used model distributions. That said, the distribution has to

be provided to the code as a data set. The required data may be obtained from spacecraft measurements, parametrized model distributions, or kinetic simulations. In this section, we exemplarily show how LEOPARD is applied to data sets from either of the three mentioned sources. We start with a benchmark of bi-Maxwellian and anisotropic kappa scenarios. Then, we examine the saturation stage of the parallel fire hose instability based on hybrid simulation data. And finally, we discuss the application of the LEOPARD code to spacecraft data.

3.1. Parametric Distributions

Parametric models allow us to deduce idealized distribution functions from observations which represent average plasma states and facilitate systematic studies of plasma properties under specific conditions described by certain parameter sets. Arguably the most important and widely used parametric model in space plasma physics is the anisotropic bi-Maxwellian distribution

$$f = \frac{1}{\pi^{3/2}} \frac{1}{v_{\parallel \mathrm{th}}} \frac{1}{v_{\perp \mathrm{th}}^2} \exp\left(-\frac{v_\parallel^2}{v_{\parallel \mathrm{th}}^2} - \frac{v_\perp^2}{v_{\perp \mathrm{th}}^2} \right), \tag{10}$$

with the thermal velocities defined by $v_{\parallel \mathrm{th}} = \sqrt{2T_\parallel/m}$ and $v_{\perp \mathrm{th}} = \sqrt{2T_\perp/m}$ containing the temperatures parallel and perpendicular to the magnetic field, $T_\parallel$ and $T_\perp$, as the two free parameters of the model. Another type of distribution function that enjoys growing interest for the modeling of solar wind and magnetospheric plasmas [*Pierrard and Lazar*, 2010] is the anisotropic kappa distribution,

$$f_\kappa = \frac{1}{\pi^{3/2}} \frac{1}{\kappa^{3/2}} \frac{1}{\theta_\parallel \theta_\perp^2} \frac{\Gamma(\kappa+1)}{\Gamma(\kappa-1/2)} \left(1 + \frac{v_\parallel^2}{\kappa \theta_\parallel^2} + \frac{v_\perp^2}{\kappa \theta_\perp^2} \right)^{-(\kappa+1)}, \tag{11}$$

which contains the additional parameter κ satisfying $3/2 \leq \kappa \leq \infty$. The expressions $\theta_\parallel = \sqrt{\frac{2\kappa-3}{\kappa} \frac{T_\parallel}{m}}$ and $\theta_\perp = \sqrt{\frac{2\kappa-3}{\kappa} \frac{T_\perp}{m}}$ denote the modified thermal velocities, and $\Gamma(x)$ is the gamma function. This power law-type distribution can be understood as an extension of the exponential bi-Maxwellian model which is recovered in the limit $\kappa \to \infty$.

For both types of distributions there already exist specific dispersion relation solvers, such as DSHARK, which are usually faster and more efficient in computing dispersion curves than the code presented in this paper. However, they can serve as a good test bed for benchmarking the LEOPARD code. Therefore, we pick four exemplary scenarios which shall be investigated with LEOPARD, and we benchmark the code's results against the DSHARK code.

Different from the dispersion relation solver presented, e.g., in *Dum et al.* [1980], LEOPARD can not only treat parallel propagating modes, but in addition it also allows for $\theta \neq 0°$. And since we see the main application of the code in studying instabilities in space plasmas, we decided to benchmark a parallel fire hose-unstable setup with $\theta = 0°$ and an oblique fire hose-unstable setup with finite propagation angle $\theta = 45°$. Both setups are tested for a bi-Maxwellian model with anisotropic ions described by $\beta_{\parallel \mathrm{i}} = 4$ and $\beta_{\perp \mathrm{i}} = 2$ and an anisotropic ion kappa distribution with $\beta_{\parallel \mathrm{i}} = 4$, $\beta_{\perp \mathrm{i}} = 2$, and $\kappa_\mathrm{i} = 4$, where the beta parameters are defined by $\beta_{\parallel \mathrm{i}} = 8\pi n_\mathrm{i} T_{\parallel \mathrm{i}}/B_0^2$ and $\beta_{\perp \mathrm{i}} = 8\pi n_\mathrm{i} T_{\perp \mathrm{i}}/B_0^2$. The electrons are assumed to be isotropic and Maxwellian with $\beta_\mathrm{e} = 1$. To simplify the analysis, the electrons are treated with a standard Maxwellian solver (see end of section 2.2), so that the code has to be provided with a distribution data set for the ions only.

The parallel as well as the oblique fire hose instability are both resonant in nature and therefore also serve as a good test to demonstrate the ability of the code to accurately solve for kinetic effects.

Figures 1 and 2 show the results of the benchmark where LEOPARD was tested for different velocity resolutions. Since the oblique fire hose instability is nonpropagating; only the growth rates are shown for $\theta = 45°$, whereas both real frequencies and growth rates are displayed for the parallel fire hose instability. We also provide a cut of the employed distribution function at $\tilde{v}_\perp = 0$, overlaid with the cubic spline interpolations for the corresponding velocity resolutions. The outer parts of the spline-interpolated distributions are apparently subject to unphysical oscillations introduced by the boundary conditions of the cubic splines. The higher the velocity resolution, the less severe are the spline overshoots, and the better are the results. In fact, for resolutions $\Delta\tilde{v}_\parallel = 0.494$ and $\Delta\tilde{v}_\parallel = 0.976$ the dispersion curves agree almost perfectly well with the predictions of DSHARK. However, for the lowest resolution, $\Delta\tilde{v}_\parallel$=1.397, the spline overshoots intrude into the inner parts of the distribution function where the resonant particle populations are located which drive

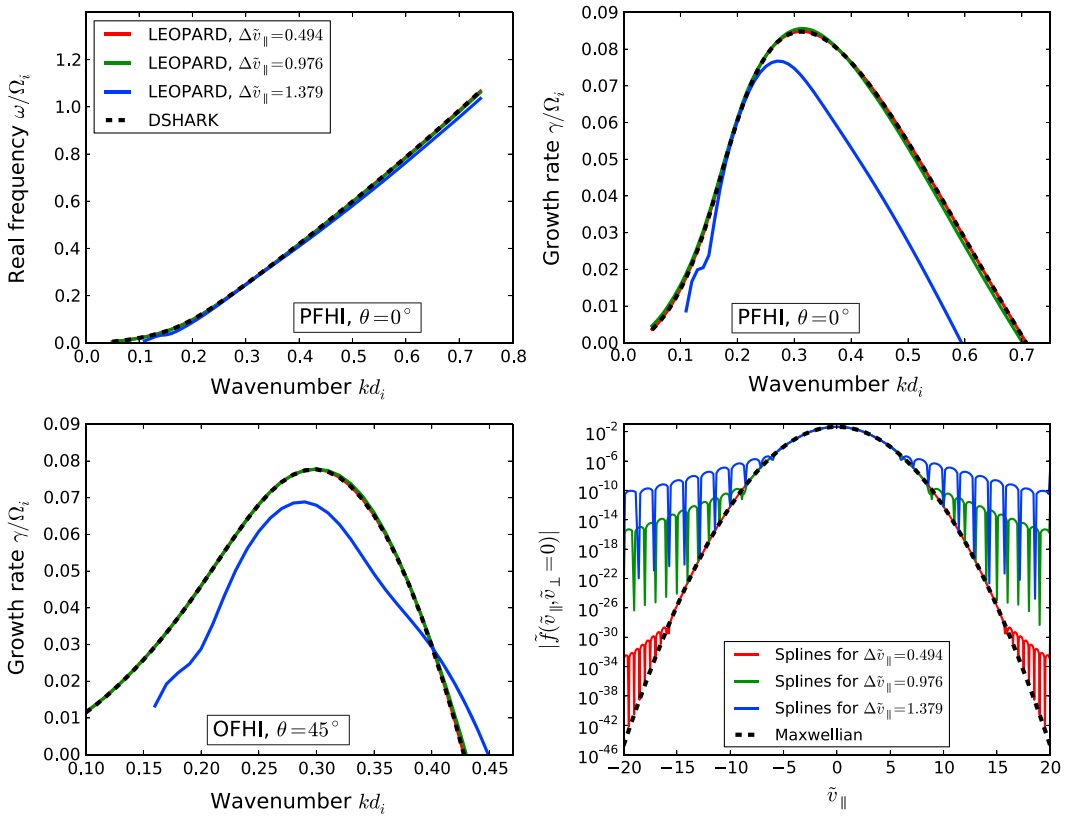

Figure 1. Results of the bi-Maxwellian fire hose benchmark. (top row) The real frequencies (left) and the growth rates (right) of the parallel fire hose instability (PFHI) determined by DSHARK for a bi-Maxwellian setup with $\beta_{\parallel i} = 4$ and $\beta_{\perp i} = 2$ and by LEOPARD for corresponding model distributions with different velocity resolutions. (bottom row) The growth rates of the oblique fire hose instability (OFHI) for the same setups but $\theta = 45°$ and a cut of the bi-Maxwellian distribution at $\tilde{v}_\perp = 0$ overlaid with the spline interpolations for the different velocity resolutions used in LEOPARD.

the fire hose instabilities. This leads to obvious deviations in the growth rates. The real frequencies on the other hand are only slightly changed since they depend rather on the overall characteristics of the distribution function and are therefore less vulnerable to the spline overshoots.

So if the velocity resolution of the distribution function is not too coarse, the spline interpolation will provide a good approximation and the LEOPARD code will accurately reproduce the dispersion curves for both parallel and obliquely propagating modes.

3.2. Application to Simulation Data

Using parametric distribution functions, we are still restricted to idealized models which are of limited applicability in real-world scenarios. The next step to more authenticity is to consider distribution functions obtained from grid-based kinetic simulations. In principle, such simulations already allow for investigating dispersion properties for arbitrary distribution functions. However, a linear eigenvalue solver such as LEOPARD can perform this task in a much more efficient and accurate way. Since LEOPARD allows for directly correlating an increase or decrease of the observed magnetic field amplitude with the local shape of the simulated distribution, it can help to identify stabilizing or destabilizing features of the distribution and, e.g., also allows for tracking how resonant particle heating changes the dispersion properties in a plasma and causes saturation of kinetic instabilities. As an exemplary study, we present a numerical analysis of the saturation stage of the bi-Maxwellian fire hose-unstable setup described in section 3.1. We carried out hybrid simulations with the semi-Lagrangian *Hybrid Vlasov Maxwell* code (HVM), presented in *Mangeney et al.* [2002] and *Valentini et al.* [2007]. The simulation setup is one-dimensional in position space and three-dimensional in velocity space. The spatial grid is chosen to be aligned with the background magnetic field. The initial distribution function of the ions is given by a bi-Maxwellian with $\beta_{\parallel i} = 4.0$ and $\beta_{\perp i} = 2.0$, and the isotropic (fluid) electrons have $\beta_e = 1.0$, in agreement with section 3.1. The instability is excited by introducing random noise perturbations

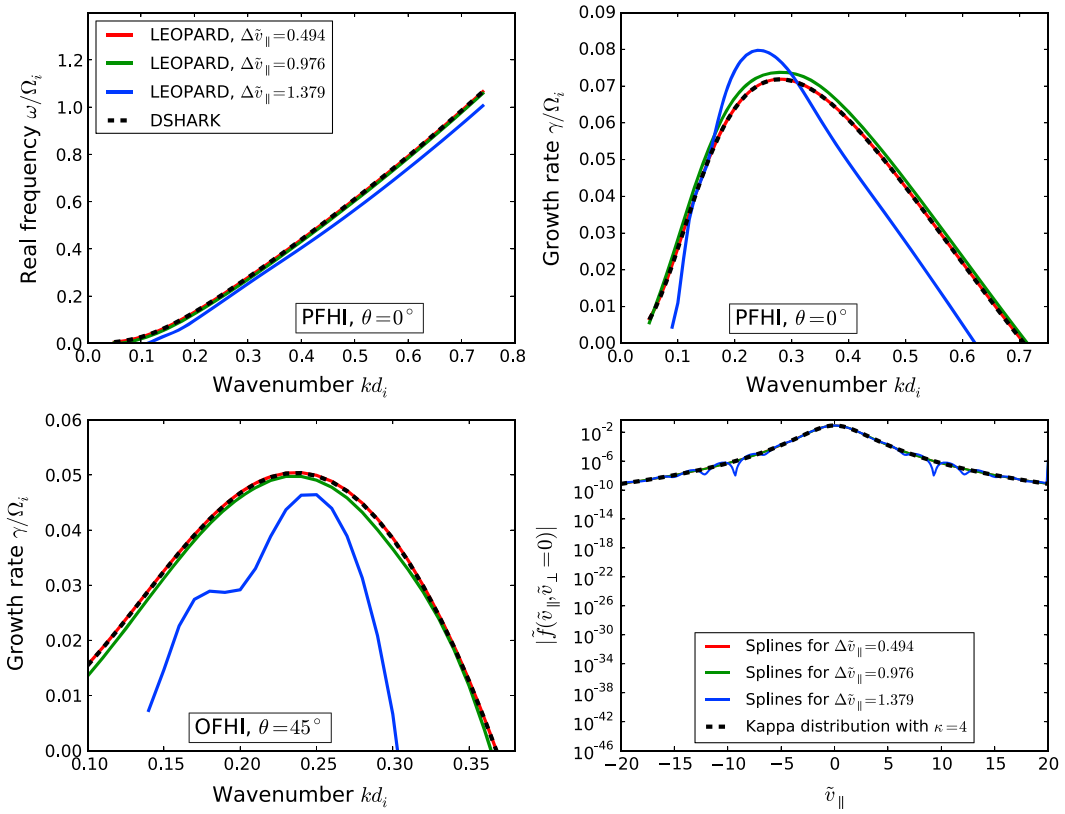

Figure 2. Results of the fire hose benchmark with an anisotropic kappa distribution. (top row) The real frequencies (left) and the growth rates (right) of the parallel fire hose instability (PFHI) determined by DSHARK for a kappa distribution with $\beta_{\parallel i}=4$, $\beta_{\perp i}=2$ and $\kappa_i=4$ and by LEOPARD for corresponding model distributions with different velocity resolutions. (bottom row) The growth rates of the oblique fire hose instability (OFHI) for the same setups but $\theta = 45°$ and a cut of the kappa distribution at $\tilde{v}_\perp = 0$ overlaid with the spline interpolations for the different velocity resolutions used in LEOPARD.

to the initial magnetic field amplitudes. For the analysis, we picked out a single point on the spatial grid and studied the time evolution of the local ion distribution function and the magnetic field amplitudes at that point. As we can see in Figure 3, the initial temperature anisotropy in the ion distribution drives an exponential growth of the magnetic field amplitudes which fades around $t\Omega_i = 200$ where the instability saturates.

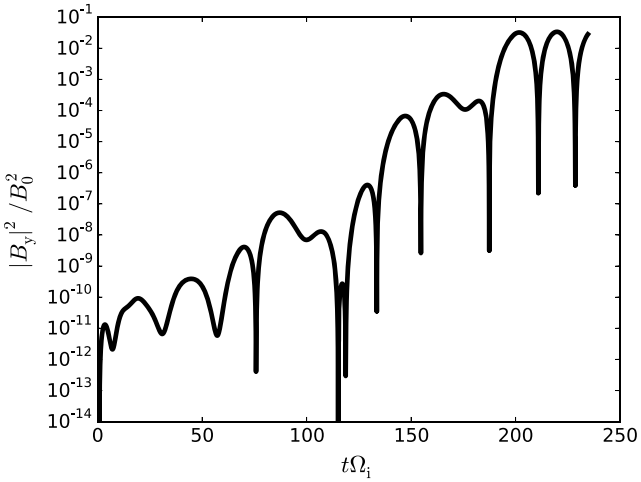

Figure 3. Time evolution of the local magnetic field amplitude B_y for a parallel fire hose-unstable setup simulated with the hybrid code HVM, starting from a bi-Maxwellian ion distribution.

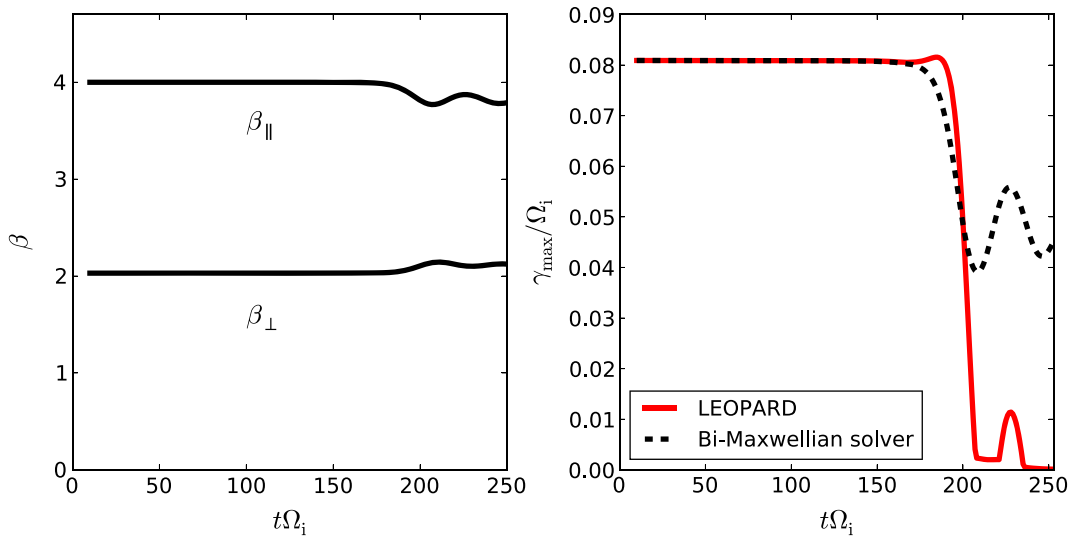

Figure 4. (left) Time evolution of the local ion beta parameters $\beta_{\parallel i}$, $\beta_{\perp i}$ and (right) time evolution of the local maximum growth rate in the simulated system which was determined with a bi-Maxwellian solver, based on the local ion beta parameters, and with the LEOPARD code, based on the local ion distribution function.

The further analysis is split into two parts. First, we compute the second-order velocity moments of the ion distribution to derive the time evolution of the temperature anisotropy at the given point on the grid. The obtained $\beta_{\parallel i}(t)$ and $\beta_{\perp i}(t)$ are shown in Figure 4 (left). Based on the computed anisotropies, we then employ a bi-Maxwellian dispersion relation solver to get the corresponding fire hose growth rates (again, we used the DSHARK code here; see dashed lines in Figure 4 (right)). In the second step, we drop the restriction imposed by a bi-Maxwellian model. Instead, we gyroaverage the local distribution and directly feed it into the LEOPARD code to get more realistic predictions of the linear dispersion properties during the saturation stage. Comparing the results of both procedures in Figure 4 reveals obvious discrepancies which can be understood by examining Figures 5–7. The pitch angle diffusion of the resonant particles caused by the

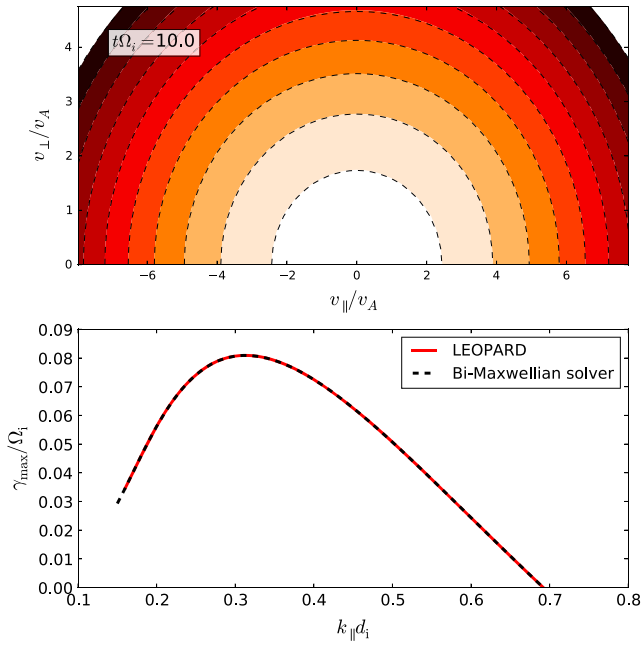

Figure 5. (top) Local ion velocity distribution function (filled contours) and corresponding bi-Maxwellian model based on the local beta parameters (dashed contours) in the early stage of growth. (bottom) The fire hose growth based on a bi-Maxwellian solver and LEOPARD.

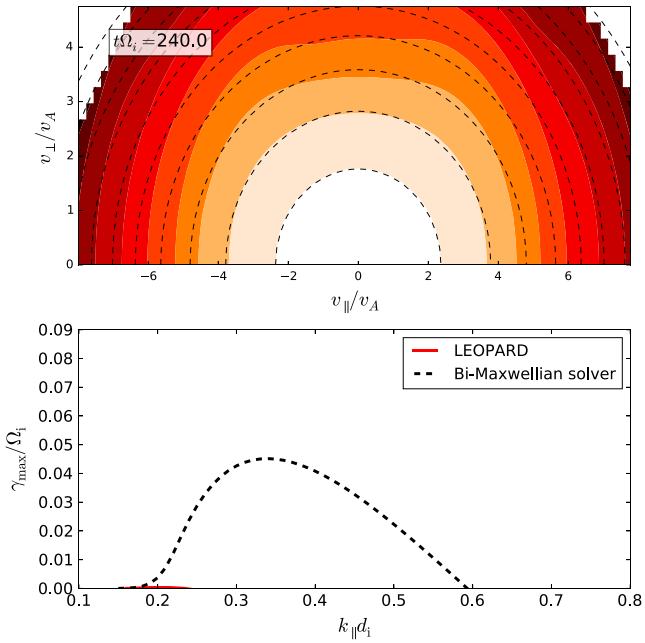

Figure 6. (top) Local ion velocity distribution function (filled contours) and corresponding bi-Maxwellian model based on the local beta parameters (dashed contours) in the late saturation stage. (bottom) The fire hose growth based on a bi-Maxwellian solver and LEOPARD.

high-amplitude parallel fire hose fluctuations introduces non-Maxwellian features in the distribution function (see Figure 6). The particles get scattered to higher $v_\perp$, and the tails in the parallel velocity direction become strongly depopulated for $v_\perp = 0$. The bi-Maxwellian solver cannot account for these deformations and systematically overpredicts the growth rates in the system. In fact, according to the predictions of the DSHARK code, the system would still be highly unstable even after the magnetic field amplitudes in the simulation clearly saturated. The maximum growth rate oscillates in phase with the temperature anisotropy around values as high as $\gamma_{max}/\Omega_i > 0.04$. Of course, one could attribute this obvious mismatch to the fact that in the saturation stage the high field amplitudes invalidate the linear approach and nonlinear effects might obscure the stability analysis. However, in contrast to the bi-Maxwellian solver the growth rates based on LEOPARD meet our expectations very well. While the magnetic field amplitudes saturate around $t\Omega_i = 200$, it predicts a strong reduction of the growth rates to $\gamma_{max}/\Omega_i \sim 10^{-3}$. Around $t\Omega_i = 230$, γ_{max} goes slightly up again which seems to be related to the fact that the local macroscopic temperature anisotropy is temporarily increasing again. After that, however, LEOPARD predicts decreasing γ_{max} again, and the fire hose branch eventually becomes stable around $t\Omega_i = 240$, while the bi-Maxwellian solver still shows strong instability.

Figure 7 provides further qualitative insight into this premature saturation of the parallel fire hose instability. It shows the ion velocity distribution during the stage of growth (Figure 7, top) and after saturation (Figure 7, bottom) with overplotted single-wave characteristics of the most unstable mode. Particles which are in cyclotron resonance with a single wave roughly conserve their kinetic energy in the frame comoving with the wave [see, e.g., *Kennel and Engelmann*, 1966], i.e., they obey

$$v_\perp^2 + \left(v_\parallel - \frac{\omega_k}{k_\parallel}\right)^2 \approx \text{const.} \tag{12}$$

The single-wave characteristics plotted in Figure 7 show the contours of the conserved energy in the comoving frame of the most unstable mode. Gradients in the phase space density along these contours provide a source of free energy. Thus, pitch angle scattering induced by the growing wave amplitude is expected to lead to a diffusion of the particles along the contours in the direction of decreasing phase space density which stabilizes the plasma [see, e.g., *Lyons and Williams*, 1984]. Comparing Figure 7 (top and bottom), we indeed see that after saturation the ion velocity distribution is smoothed out along the single-wave characteristics which eventually leads to the suppression of the fire hose growth.

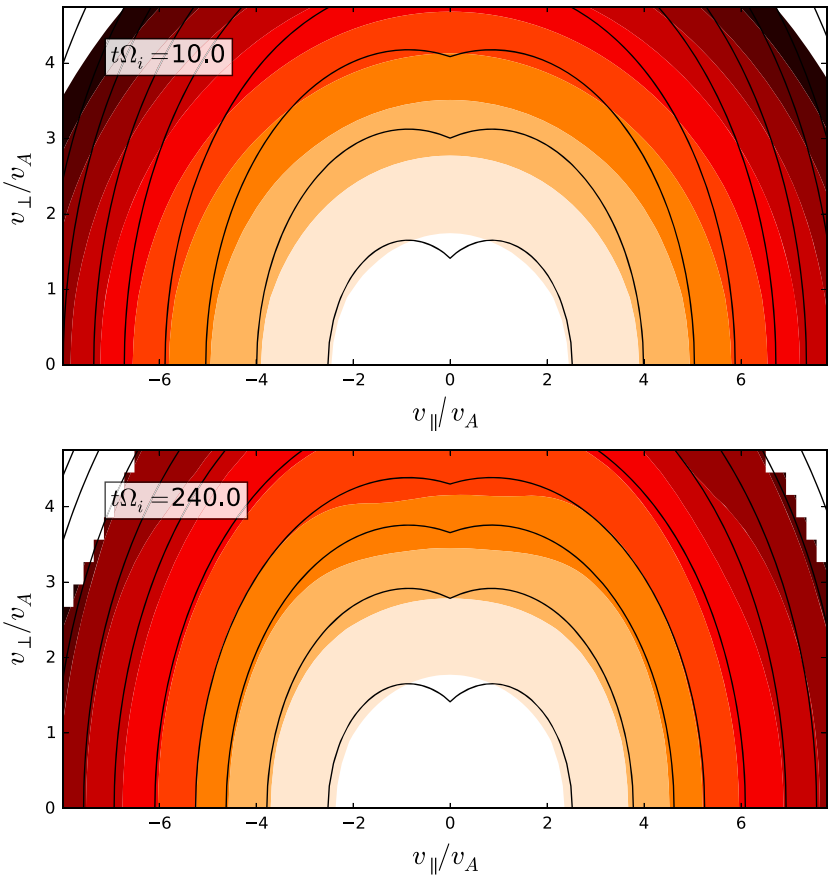

Figure 7. Local ion velocity distribution function (filled contours) in (top) the early stage of growth and in (bottom) the late saturation stage with overplotted single-wave characteristics determined by equation (12).

This observation confirms that for the given setup the saturation of the parallel fire hose instability is not primarily governed by the reduction of the macroscopic pressure anisotropy but it is rather due to resonant pitch angle scattering. Furthermore, it shows that a linear stability analysis can still produce useful results even in the presence of high fluctuation amplitudes as long as the distribution function is correctly described.

For further discussion on how these results comply with former investigations of the parallel fire hose saturation, see section 4.

3.3. Application to Space Measurements

After discussing the application of the LEOPARD code to parametric distributions and simulation data, we now conclude by shortly outlining what we consider the main purpose of the code in future studies, namely, the application to distribution data obtained from spacecraft measurements. Since measurements of space plasma distribution functions are usually performed in a three-dimensional cartesian velocity space, a proper transformation to field-aligned coordinates and a subsequent averaging over the gyroangle is required before the data can be processed by the code. Due to gyrophase bunching, space plasmas can generate a significant agyrotropy [see, e.g., *Eastman et al.*, 1981; *Gary et al.*, 1986], which alters the dispersion properties. While performing the gyroaveraging, it is therefore important to ensure that the gyrotropy assumption inherent to the code is still tolerably satisfied. This—of course—also applies to distribution data from kinetic simulations of collisionless plasmas discussed above.

In *Dum et al.* [1980], wave growth was determined based on distribution data from the Helios 1 and 2 spacecraft. At this point, we refrain from demonstrating the application of LEOPARD to a specific data set but we merely refer to an upcoming paper where the LEOPARD code is applied to measurements of the ARTEMIS spacecraft, THEMIS 1 and 2. In this upcoming work, it will be shown that the wave growth of ion beam instabilities can crucially depend on the actual shape of the ion beam in velocity space. This reveals the limitations of

conventionally used drift bi-Maxwellian dispersion solvers and justifies the need for an arbitrary distribution dispersion solver in the space plasma community.

The performance of the LEOPARD code crucially depends on the quality of the applied cubic spline interpolation which furthermore relies on the velocity resolution of the provided data. For parametric model distributions and simulation data, the resolution is solely limited by the numerical demand and can generally be chosen high enough to ensure a robust analysis with LEOPARD. Spacecraft measurements, on the other hand, are limited by the resolution of their instrumentation which might not yet meet the requirements for a reliable cubic spline interpolation in all cases. Checking the spline interpolation for unphysical overshoots provides an intuitive way to locate errors stemming from such poor data resolution.

4. Discussion and Outlook

In this paper, we presented a newly developed dispersion relation solver for arbitrary gyrotropic distribution functions which — to the authors' knowledge — is the first kinetic eigenvalue solver of that kind, allowing for arbitrary propagation angles and an arbitrary number of particle species. The required distribution function is provided to the code as a data set sampled on a $(\tilde{v}_{\parallel}, \tilde{v}_{\perp})$ grid. Applying a cubic spline interpolation to the data then allows for analytically solving the velocity integrals which appear in the standard dispersion formalism.

In section 3, we successfully benchmarked the code with a bi-Maxwellian and an anisotropic kappa model distribution. Subsequently, we investigated the saturation stage of the parallel fire hose instability by examining data produced with the hybrid code HVM, and we clearly demonstrated that for the chosen setup the saturation of the parallel fire hose instability is mainly achieved by the pitch angle scattering of resonant particles and not primarily by the reduction of the pressure anisotropy. This investigation can be understood as a follow-up on the discussions in, e.g., *Gary et al.* [1998], *Matteini et al.* [2006], *Seough et al.* [2015], and *Astfalk and Jenko* [2016] concerning the resonant nature of the parallel fire hose instability and the effect of the deformation of the distribution function on its saturation. *Seough et al.* [2015] presented a thorough study, utilizing Particle-in-cell simulations to investigate the saturation of the parallel fire hose instability for different $\beta_{\parallel \mathrm{i}}$ and $\beta_{\perp \mathrm{i}}$ and comparing the simulation results to the predictions of quasilinear theory. It was found that especially for lower plasma β, there was an obvious discrepancy between simulation and theory. This was attributed to either the presence of nonlinear wave-wave interactions or the non-Maxwellian deformation of the distribution due to pitch angle scattering which the chosen approach did not account for — *Seough et al.* [2015] assumed that the distribution's bi-Maxwellian shape is preserved throughout the saturation process. Based on the results of our short investigation, we suggest that the deviation of the distribution from a bi-Maxwellian shape may very well be the major cause of the premature instability saturation observed in the lower β setups of *Seough et al.* [2015]. The pitch angle scattering of the resonant particles clearly inhibits further growth, while the quasilinear theory based on a bi-Maxwellian model would still predict further instability. A more thorough analysis of this and similar problems is left for the future.

Finally, we briefly discussed the application of LEOPARD to direct measurements of spacecraft in natural plasma environments. A first systematic study of this kind is postponed to an upcoming paper where the code will be applied to THEMIS measurements. In section 3.3, we mentioned that the limited resolution of spacecraft data can be a severe issue for the performance of the LEOPARD code. However, the improved instrumentation of upcoming space missions such as Solar Orbiter and THOR will enable a more and more accurate linear dispersion analysis. In both simulation and real-world applications, LEOPARD will allow for directly linking observed growth (or damping) in magnetic field amplitudes with features in the local distribution functions with unprecedented accuracy. It is our hope that this will further deepen the knowledge of linear wave physics in both basic theory and natural plasma environments such as the solar wind or planetary magnetospheres.

Appendix A: Cubic Spline Interpolation

The cubic spline interpolation for a given set of n data points $(x_1, y_1), \ldots, (x_n, y_n)$ can generally be written as

$$S(x) = \sum_{i=1}^{n-1} S_i(x) = \sum_{i=1}^{n-1} a_i \times (x - x_i)^3 + b_i \times (x - x_i)^2 + c_i \times (x - x_i) + d_i, \tag{A1}$$

where $x_i \leq x \leq x_{i+1}$ and $1 \leq i \leq n - 1$.

The parameters a_i, b_i, c_i, and d_i are the spline coefficients which have to be determined. Several types of spline interpolation schemes satisfying different boundary conditions can be found in the literature [see, e.g., *De Boor et al.*, 1978; *McKinley and Levine*, 1998]. In most of our cases, natural splines (also known as free splines) have proven to be the best choice. The corresponding spline coefficients can be determined using the following conditions:

(I) $S_i(x_i) = y_i$ for $1 \leq i \leq n-1$
(II) $S_i(x_{i+1}) = y_{i+1}$ for $1 \leq i \leq n-1$
(III) $S'_{i-1}(x_i) = S'_i(x_i)$ for $2 \leq i \leq n-1$
(IV) $S''_{i-1}(x_i) = S''_i(x_i)$ for $2 \leq i \leq n-1$
(V) $S''_1(x_1) = 0$ and $S''_{n-1}(x_n) = 0$.

The conditions (I) and (II) ensure the continuity of the splines at each grid point. Conditions (III) and (IV) furthermore ensure the continuity of the first and second derivatives at the grid points to guarantee the smoothness of the splines. And condition (V) is the boundary condition for natural splines. These conditions create a complete set of equations which uniquely determines the coefficients of the spline interpolation. This system of equations takes the form of a tridiagonal matrix and can be solved using the Thomas algorithm.

The spline interpolation method implemented in the LEOPARD code was thankfully provided by Prof. Alexander Godunov, Associate Professor of physics at Old Dominion University, Norfolk, VA, USA.

Appendix B: Hypergeometric Function

The generalized hypergeometric function can be written as

$$ {}_pF_q\left(a_1,\ldots,a_p;b_1,\ldots,b_q;z\right) = \sum_{n=0}^{\infty} \frac{\prod_{k=1}^{p}\left(a_k\right)_n}{\prod_{k=1}^{q}\left(b_k\right)_n} \times \frac{z^n}{n!}, \tag{B1} $$

where $\left(x_k\right)_n = \Gamma(x_k+n)/\Gamma(x_k)$ denotes the Pochhammer symbol [see, e.g., *Slater*, 1966].

The hypergeometric functions ${}_1F_2$ and ${}_2F_3$ appear in the code when solving the $v_\perp$ integral, equation (9), after the cubic spline interpolation, equation (A1), was applied to the grid-based quantity $F\left(\tilde{v}_\parallel,\tilde{v}_\perp\right)$. Exemplarily, the solution of, e.g., $I^m = \int \mathrm{d}\tilde{v}_\perp \tilde{v}_\perp^{m-1} J_n^2\left(\lambda_\alpha \tilde{v}_\perp\right)$ is given by

$$ I^m = \frac{1}{2n+m}\frac{\tilde{v}_\perp^m}{2^{2n}}\frac{\left(\lambda_\alpha\tilde{v}_\perp\right)^{2n}}{\Gamma(n+1)^2}\,{}_2F_3\left(n+\frac{1}{2},n+\frac{m}{2};n+1,n+1+\frac{m}{2},2n+1;-\left(\lambda_\alpha\tilde{v}_\perp\right)^2\right). \tag{B2} $$

For numerical reasons, the hypergeometric functions are not directly computed with equation (B1) but with the continued fraction formula

$$ {}_pF_q\left(a_1,\ldots,a_p;b_1,\ldots,b_q;z\right) = 1 + \frac{\left(z\prod_{k=1}^{p}a_k\right)}{\left(\prod_{k=1}^{q}b_k\right)\left(1+K_k\left[-\frac{z\prod_{j=1}^{p}(k+a_j)}{(k+1)\prod_{j=1}^{q}(k+b_j)},\ \frac{z\prod_{j=1}^{p}(k+a_j)}{(k+1)\prod_{j=1}^{q}(k+b_j)}+1\right]_1^{\infty}\right)}. \tag{B3} $$

Acknowledgments
This work was facilitated by the Max-Planck/Princeton Center for Plasma Physics and by the Plasma Science and Technology Institute at UCLA. We warmly acknowledge F. Califano for providing access to the HVM code. We also thank A. Godunov for providing his cublic spline interpolation algorithm. Furthermore, we thank P. Crandall, S. Dorfman, R. Bilato, and the referees for their helpful comments and suggestions. To get access to the presented results and the source code of the presented dispersion relation solver, please contact the corresponding author (patrick.astfalk@ipp.mpg.de).

References

Astfalk, P., and F. Jenko (2016), Parallel and oblique firehose instability thresholds for bi-kappa distributed protons, *J. Geophys. Res. Space Physics*, *121*, 2842–2852, doi10.1002/2015JA022267.

Astfalk, P., T. Görler, and F. Jenko (2015), DSHARK: A dispersion relation solver for obliquely propagating waves in bi-kappa-distributed plasmas, *J. Geophys. Res. Space Physics*, *120*, 7107–7120, doi10.1002/2015JA021507.

Baumjohann, W., and R. A. Treumann (1996), *Basic Space Plasma Physics*, Imperial College Press, London.

Bilato, R., O. Maj, and M. Brambilla (2014), An algorithm for fast Hilbert transform of real functions, *Adv. Comput. Math.*, *40*(5), 1159–1168, doi10.1007/s10444-014-9345-4.

Brambilla, M. (1998), *Kinetic Theory of Plasma Waves: Homogeneous Plasmas*, International Series of Monographs on Physics, vol. 96, Clarendon, Oxford, U. K.

Brambilla, M., and R. Bilato (2009), Advances in numerical simulations of ion cyclotron heating of non-Maxwellian plasmas, *Nuclear Fusion*, *49*(8), 85004.

De Boor, C., C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor (1978), *A Practical Guide to Splines*, vol. 27, Springer-Verlag, New York.

Dory, R. A., G. E. Guest, and E. G. Harris (1965), Unstable electrostatic plasma waves propagating perpendicular to a magnetic field, *Phys. Rev. Lett.*, *14*, 131–133, doi10.1103/PhysRevLett.14.131.

Dum, C. T., E. Marsch, and W. Pilipp (1980), Determination of wave growth from measured distribution functions and transport theory, *J. Plasma Phys.*, *23*, 91–113, doi10.1017/S0022377800022170.

Eastman, T. E., R. R. Anderson, L. A. Frank, and G. K. Parks (1981), Upstream particles observed in the Earth's foreshock region, *J. Geophys. Res.*, *86*, 4379–4395, doi10.1029/JA086iA06p04379.

Gary, S. P., M. F. Thomsen, and S. A. Fuselier (1986), Electromagnetic instabilities and gyrophase-bunched particles, *Phys. Fluids*, *29*, 531–535, doi10.1063/1.865441.

Gary, S. P., H. Li, S. O'Rourke, and D. Winske (1998), Proton resonant firehose instability: Temperature anisotropy and fluctuating field constraints, *J. Geophys. Res.*, *103*, 14,567–14,574, doi10.1029/98JA01174.

Kennel, C. F., and F. Engelmann (1966), Velocity space diffusion from weak plasma turbulence in a magnetic field, *Phys. Fluids*, *9*, 2377–2388, doi10.1063/1.1761629.

Lyons, L. R., and D. J. Williams (1984), *Quantitative Aspects of Magnetospheric Physics*, Reidel Publishing Company, 15+231 pp., Dordrecht, Boston and Lancaster.

Mangeney, A., F. Califano, C. Cavazzoni, and P. Travnicek (2002), A numerical scheme for the integration of the Vlasov-Maxwell system of equations, *J. Comput. Phys.*, *179*, 495–538, doi10.1006/jcph.2002.7071.

Marsch, E. (2006), Kinetic physics of the solar corona and solar wind, *Living Rev. Sol. Phys.*, *3*, doi10.12942/lrsp-2006-1.

Matteini, L., S. Landi, P. Hellinger, and M. Velli (2006), Parallel proton fire hose instability in the expanding solar wind: Hybrid simulations, *J. Geophys. Res.*, *111*, A10101, doi10.1029/2006JA011667.

McKinley, S., and M. Levine (1998), Cubic spline interpolation, *College Redwoods*, *45*(1), 1049–1060.

Micchelli, C. A., Y. Xu, and B. Yu (2013), On computing with the Hilbert spline transform, *Adv. Comput. Math.*, *38*(3), 623–646, doi10.1007/s10444-011-9252-x.

Pierrard, V., and M. Lazar (2010), Kappa distributions: Theory and applications in space plasmas, *Sol. Phys.*, *267*, 153–174, doi10.1007/s11207-010-9640-2.

Roennmark, K. (1982), Waves in Homogeneous, Anisotropic Multicomponent Plasmas (WHAMP), *Tech. Rep.*, Kiruna Geophys. Inst., Kiruna, Sweden.

Seough, J., P. H. Yoon, and J. Hwang (2014), Quasilinear theory and particle-in-cell simulation of proton cyclotron instability, *Phys. Plasmas*, *21*(6), 62118, doi10.1063/1.4885359.

Seough, J., P. H. Yoon, and J. Hwang (2015), Simulation and quasilinear theory of proton firehose instability, *Phys. Plasmas*, *22*(1), 12303, doi10.1063/1.4905230.

Slater, L. J. (1966), *Generalized Hypergeometric Functions*, Cambridge Univ. Press, Cambridge, U. K.

Valentini, F., P. Trávníček, F. Califano, P. Hellinger, and A. Mangeney (2007), A hybrid-Vlasov model based on the current advance method for the simulation of collisionless magnetized plasma, *J. Comput. Phys.*, *225*, 753–770, doi10.1016/j.jcp.2007.01.001.