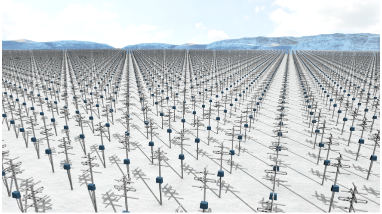
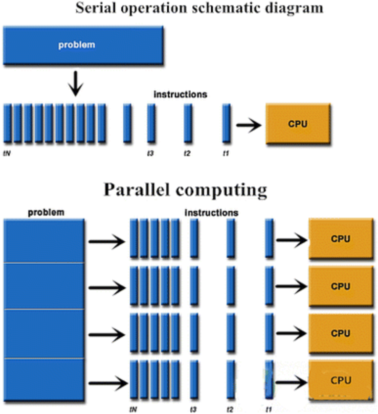
A Summary of Methods and Codes for Computing the Plasma Dispersion Function and IS Spectrum

Stephan C. Buchert (scb@irfu.se),
Swedish Institute of Space Physics

20^{th} EISCAT_3D User Meeting 2025-12-03, Kiruna

E3D data analysis





- E3D: many beams, tristatic volume scattering, . . .
- $\sim 100 - 1000$ more data to analyse!
- parallel computing, super computers,
- and/or efficient computation of the plasma dispersion function.

Special Function

Numerical calculation of the IS spectrum for Maxwellian distributions rests on either

1) the plasma dispersion function, code name `friedconte(z)`: B.D. Fried, S.D. Conte, The plasma dispersion function. New York Academic Press, 1961.
2) complex error function `erfc(z)`:

$$\texttt{friedconte}(z) = i\sqrt{\pi}\exp\left(-z^2\right)\left(1 + \texttt{erfc}(iz)\right) \tag{1}$$

3) the Faddeeva function

$$\texttt{faddeeva}(z) = \exp\left(-z^2\right)\texttt{erfc}(-\texttt{iz}) \tag{2}$$

4) the Dawson integral

$$D(x) = \exp\left(-x^2\right)\int_0^x t^2 dt \tag{3}$$

Any one of the four functions is needed.

Fried-Conte

```
      SUBROUTINE PLASMA
C*THIS ROUTINE COMPUTES THE COMPLEX PLASMA DISPERSION FUNCTION
C GIVEN BY:
C              Z(S)=I*SQRT(PIE)*EXPC(-S**2)*(1.+ERFC(I*S)
C WHERE:
C          I=SQRT(-1.) ;  S=X+I*Y=COMPLEX ARGUMENT
C FOR ABS(Y).GT.1.0, THE CONTINUED FRACTION EXPANSION GIVEN BY FRIED
C AND CONTE (1961) IS USED; WHILE FOR ABS(Y).LE.1.0, THE FOLLOWING
C DIFFERENTIAL EQUATION IS SOLVED:
C            D Z(S)
C            ------ = -2.*(1.+S*Z(S))
C             D S
C SUBJECT TO Z(0)=I*SQRT(PIE)
C
C     "F(K)"=TRUE FREQUENCY.
C     "X(K)"=NORMALIZED FREQUENCY.
C     "SCALEF"=FREQUENCY SCALING FACTOR FOR NORMALIZATION.
C ----------------------------------------------------------------------
C BY WES SWARTZ
```

- Arecibo IS analysis written by Wes Swartz
- FORTRAN, IBM Mainframe
- Solving the diff. equation is not easy to parallelize :-(

Padé approximations and related

(Invented at Umeå U.)

Rapid Computation of the Plasma Dispersion Function: Rational and Multi-pole Approximation, and Improved Accuracy

Huasheng Xie[1,2,*]

[1]Hebei Key Laboratory of Compact Fusion, Langfang 065001, China
[2]ENN Science and Technology Development Co., Ltd., Langfang 065001, China

(Dated: April 30, 2024)

The plasma dispersion function $Z(s)$ is a fundamental complex special integral function widely used in the field of plasma physics. The simplest and most rapid, yet accurate, approach to calculating it is through rational or equivalent multi-pole expansions. In this work, we summarize the numerical coefficients that are practically useful to the community. Besides the Padé approximation to obtain coefficients, which are accurate for both small and large arguments, we also employ optimization methods to enhance the accuracy of the approximation for the intermediate range. The best coefficients provided here for calculating $Z(s)$ can deliver twelve significant decimal digits. This work serves as a foundational database for the community for further applications.

Matlab code for Z fun with optimized J=8 pole

```
function Zeta = zfunJ8(z)
Zeta=0.*z;
bj(1)=0.00383968430671409 - 0.0119854387180615i;
bj(2)=-0.321597857664957 - 0.218883985607935i;
bj(3)=2.55515264319988 + 0.613958600684469i;
bj(4)=-2.73739446984183 + 5.69007914897806i;
cj(1)=2.51506776338386 - 1.60713668042405i;
cj(2)=-1.68985621846204 - 1.66471695485661i;
cj(3)=0.981465428659098 - 1.70017951305004i;
cj(4)=-0.322078795578047 - 1.71891780447016i;
bj(5:8)=conj(bj(1:4));
cj(5:8)=-conj(cj(1:4));

idx=(imag(z)>=0);

Zeta(~idx)=2i*sqrt(pi)*exp(-(z(~idx)).^2);
for j=1:length(bj)
  Zeta(idx)=Zeta(idx)+bj(j)./(z(idx)-cj(j));
  Zeta(~idx)=Zeta(~idx)+conj(bj(j)./(conj(z(~idx))-cj(j)));
end
end
```

Python code for Z fun with optimized J=8 pole

```
import numpy as np

def zfunJ8(z):
    Zeta = np.zeros_like(z, dtype=complex)
    bj = np.array([0.00383968430671409 - 0.0119854387180615j,
             -0.321597857664957 - 0.218883985607935j,
             2.55515264319988 + 0.613958600684469j,
             -2.73739446984183 + 5.69007914897806j])
    cj = np.array([2.51506776338386 - 1.60713668042405j,
             -1.68985621846204 - 1.66471695485661j,
             0.981465428659098 - 1.70017951305004j,
             -0.322078795578047 - 1.71891780447016j])
    bj = np.concatenate((bj, np.conj(bj[::-1])))
    cj = np.concatenate((cj, -np.conj(cj[::-1])))

    idx = np.imag(z) >= 0
    Zeta[~idx] = 2j * np.sqrt(np.pi) * np.exp(-(z[~idx])**2)
    for j in range(len(bj)):
        Zeta[idx] += bj[j] / (z[idx] - cj[j])
        Zeta[~idx] += np.conj(bj[j] / (np.conj(z[~idx]) - cj[j]))

    return Zeta
```

FIG. 10. Sample code for calculate Z function with optimized $J = 8$ pole for all range of argument z, with max errors of 10^{-6}. One who needs higher accurary, can use the larger J coefficients, such as $J = 10, 12, 16, 20, 24$.

- Matlab and Python not compiled, i.e., not fast?

Non-Maxwellian

$$\frac{e\,(\mathbf{E}_0 + \mathbf{v}' \times \mathbf{z}_0 B_0)}{m_i} \cdot \frac{\partial F(\mathbf{v}')}{\partial \mathbf{v}'} = -\nu_{in}[F(\mathbf{v}') - F_n(\mathbf{v}')] \quad (1)$$

where $\mathbf{B}_0 = B_0\mathbf{z}_0$, $\mathbf{E}_0 = E_0\mathbf{y}_0$, e and m_i are the ion charge and mass, F and F_n are the ion and neutral velocity distribution functions normalized so that their velocity integral is unity, $\mathbf{v}'$ and $\mathbf{E}_0$ are measured in the reference frame of the neutrals, and F_n is taken to be Maxwellian in $\mathbf{v}'$. Transferring to the frame moving with the velocity $\mathbf{v}_d = \mathbf{E}_0 \times \mathbf{B}_0/B_0^2 = (E_0/B_0)\mathbf{x}_0$ and introducing

$$\mathbf{v} = \mathbf{v}' - \mathbf{v}_d \qquad v_x = v_\perp \cos\theta \qquad v_y = v_\perp \sin\theta$$

$$v'^2 = v_z^2 + v_\perp^2 \sin^2\theta + (v_\perp \cos\theta + v_d)^2 \quad (2)$$

$$\Omega_i = eB_0/m_i$$

we can rewrite (1) as

$$\frac{\partial}{\partial\theta} F(v_\perp, v_z, \theta) = \frac{\nu_{in}}{\Omega_i}\,[F(v_\perp, v_z, \theta) - F_n(v_\perp, v_z, \theta)] \quad (3)$$

This equation can be easily integrated for arbitrary ν_{in}/Ω_i. However, we are most interested in the limit $\nu_{in}/\Omega_i << 1$, in which case, (3) implies that $\partial F/\partial\theta \simeq 0$; hence F can be split into two terms, the first of which depends only on $v_\perp$ and v_z, while the second is of the order of ν_{in}/Ω_i and depends on $v_\perp$, v_z, and θ. Furthermore, if we average (3) over θ, the left-hand side is exactly zero, since F is periodic in θ, and we obtain

$$F \cong \bar{F}_n = (2\pi)^{-1} \int_0^{2\pi} F_n \, d\theta$$

$$= (2\pi v_n)^{-3/2}\, I_0\!\left(\frac{v_\perp v_d}{v_n^2}\right) \exp\left(\frac{-(v_\perp^2 + v_z^2 + v_d^2)}{v_n^2}\right) \quad (4)$$

where I_0 is the zero-order Bessel function of imaginary argu-

Ott and Farley, JGR, 1975

Computing the generalized plasma dispersion function for non-Maxwellian plasmas, with applications to Thomson scattering

C. R. Skolar, W. J. Longley, and L. V. Goodwin
Center for Solar-Terrestrial Research, New Jersey Institute of Technology, Newark, NJ 07102, USA

(*lindsay.v.goodwin@njit.edu)

(*william.longley@njit.edu)

(*chirag.skolar@njit.edu)

(Dated: 5 February 2025)

Kinetic plasma studies often require computing integrals of the velocity distribution over a complex-valued pole. The standard method is to solve the integral in the complex plane using the Plemelj theorem, resulting in the standard plasma dispersion function for Maxwellian plasmas. For non-Maxwellian plasmas, the Plemelj theorem does not generalize to an analytic form, and computational methods must be used. In this paper, a new computational method is developed to accurately integrate a non-Maxwellian velocity distribution over an arbitrary set of complex valued poles. This method works by keeping the integration contour on the real line, and applying a trapezoid rule-like integration scheme over all discretized intervals. In intervals containing a pole, the velocity distribution is linearly interpolated, and the analytic result for the integral over a linear function is used. The integration scheme is validated by comparing its results to the analytic plasma dispersion function for Maxwellian distributions. We then show the utility of this method by computing the Thomson scattering spectra for several non-Maxwellian distributions: the kappa, super Gaussian, and toroidal distributions. Thomson scattering is a valuable plasma diagnostic tool for both laboratory and space plasmas, but the technique relies on fitting measured wave spectra to a forward model, which typically assumes Maxwellian plasmas. Therefore, this integration method can expand the capabilities of Thomson scatter diagnostics to regimes where the plasma is non-Maxwellian, including high energy density plasmas, frictionally heated plasmas in the ionosphere, and plasmas with a substantial suprathermal electron tail.

I. INTRODUCTION

The plasma dispersion function[1] is typically defined for a Maxwellian distribution as

$$Z_M(z) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{\exp(-v^2)}{v - z} dv \quad (1)$$

have a nonzero imaginary part. This requirement is automatically satisfied when the kinetic equations are Laplace transformed in time, including the case where the imaginary part of the pole limits to 0 for collisionless damping. Furthermore, the list of poles z_i can include complex conjugates, as these will be treated as different poles. In this paper, we develop a novel method for numerically integrating Eq. 3.

Non-Maxwellian

PDRK: A General Kinetic Dispersion Relation Solver for Magnetized Plasma*

XIE Huasheng (谢华生), XIAO Yong (肖湧)

Institute for Fusion Theory and Simulation and the Department of Physics,
Zhejiang University, Hangzhou 310027, China

Abstract A general, fast, and effective approach is developed for numerical calculation of kinetic plasma linear dispersion relations. The plasma dispersion function is approximated by J-pole expansion. Subsequently, the dispersion relation is transformed to a standard matrix eigenvalue problem of an equivalent linear system. Numerical solutions for the least damped or fastest growing modes using an 8-pole expansion are generally accurate; more strongly damped modes are less accurate, but are less likely to be of physical interest. In contrast to conventional approaches, such as Newton's iterative method, this approach can give either all the solutions in the system or a few solutions around the initial guess. It is also free from convergence problems. The approach is demonstrated for electrostatic dispersion equations with one-dimensional and two-dimensional wavevectors, and for electromagnetic kinetic magnetized plasma dispersion relation for bi-Maxwellian distribution with relative parallel velocity flows between species.

References

[9] J. A. C. Weideman. 1994. Computation of the complex error function. SIAM J. Numer. Anal. 31, 5, 1497–1518.

[10] R. N. Franklin, The computation of the plasma dispersion function, Plasma Phys. 10, 805 (1968).

[11] G. Németh, A. Ág, and G. Páris, Two-sided Padé approximations for the plasma dispersion function, J. Math. Phys. 22, 1192 (1981).

[12] B. D. Fried, Burton, C. L. Hedrick and J. Mccune, Two-Pole Approximation for the Plasma Dispersion Function, Physics of Fluids, 11, 1, 249-252 (1968).

[13] B. S. Newberger, Efficient numerical computation of the plasma dispersion function, Comput. Phys. Commun. 42, 305 (1986).

[14] A. Tjulin, A. I. Eriksson, and M. André, Physical interpretation of the Padé approximation of the plasma dispersion function, J. Plasma Phys. 64, 287 (2000).

https://github.com/hsxie/pdrk/.

[16] H.S. Xie, BO: A unified tool for plasma waves and instabilities analysis, Comput. Phys. Comm. 244 (2019) 343-371. Xie, H. S., Denton, R., Zhao, J. S. and Liu, W, BO 2.0: Plasma Wave and Instability Analysis with Enhanced Polarization Calculations arXiv:2103.16014, 2021. https://github.com/hsxie/bo/.

[17] Ronnmark, K. 1982, WHAMP - Waves in Homogeneous Anisotropic Multicomponent Plasmas, Tech. Rep. 179, Kiruna Geophysical Institue, Kiruna, Sweden.

[18] P. Hunana, A. Tenerani, G. P. Zank, M. L. Goldstein, G. M. Webb, M. Velli and L. Adhikari, A brief guide to fluid models with anisotropic temperatures Part 2 - Kinetic theory, Padé approximants and Landau fluid closures, 2019.

[19] G. W. Hammett and W. F. Perkins, Phys. Rev. Lett., 64, 3019 (1990).

Conclusions

- Keep track of developments for computing IS spectra *fast*
- Coordinate efforts by users and EISCAT for E3D data analysis!
- Review numerical methods and existing code;
- Collect code, documentation etc in a maintained repository;
- (ISSI) proposal establish, Lisa Baddeley and Lindsay Goodwin
- develop further the analysis software for common use.