NHDSPy Documentation

Release v0.1+8.g48f038c

David Stansby

Jan 24, 2019

Contents

1 Installing **3**

2 Running **5**

3 Example **7**

4 Indices and tables **13**

Python Module Index **15**

nhdspy is a wrapper for the NHDS dispersion relation solver. For the original NHDS code see https://github.com/danielver02/NHDS.

If you use this software to produce data for publication, please cite the NHDS paper: http://iopscience.iop.org/article/10.3847/2515-5172/aabfe3

CHAPTER 1

Installing

To install nhdspy, run

```
pip install nhdspy
```

CHAPTER 2

Running

NHDS is written in fortran, so requires a working fortran compiler. When nhdspy is first run it attempts to compile NHDS; if it is not successfully it gracefully exits and prints the fortran compile error.

CHAPTER 3

Example

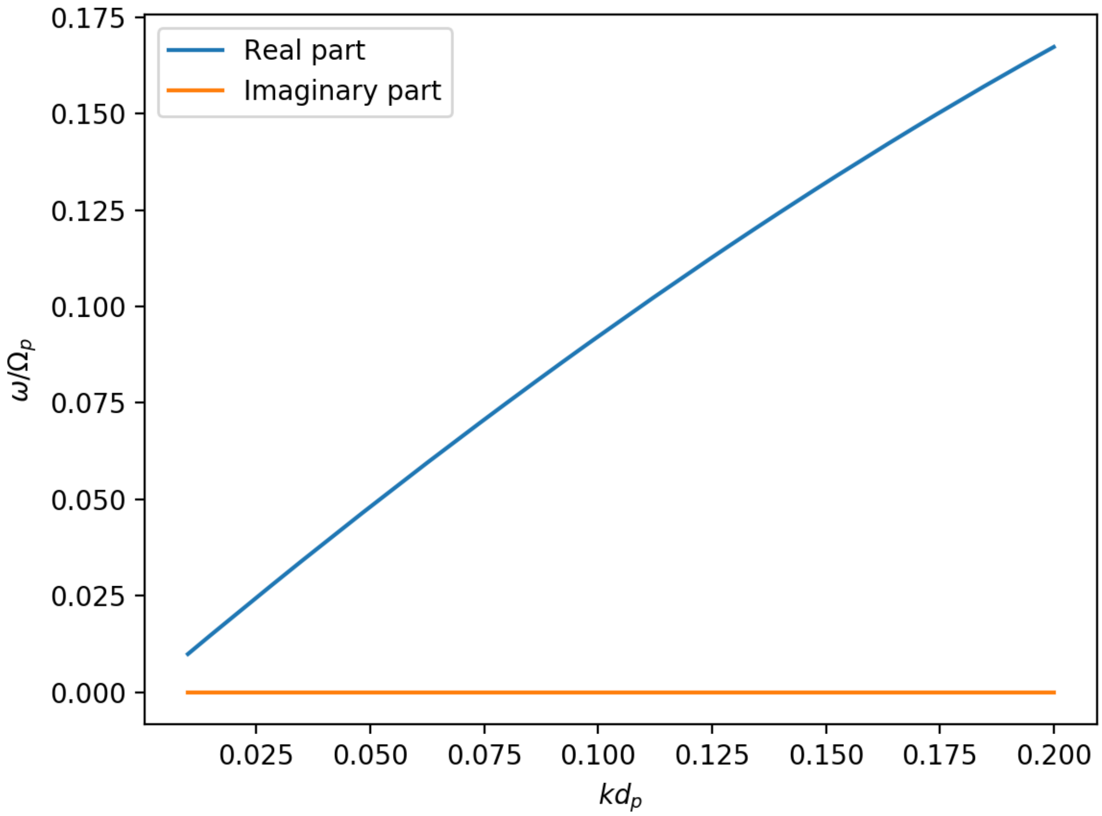
```
import nhdspy
import matplotlib.pyplot as plt

electrons = nhdspy.Species(-1, 1 / 1836, 1, 0, 1, 1)
protons = nhdspy.Species(1, 1, 1, 0, 1, 1)

va = 1e-4
theta_kB = 0.001
omega_0 = 0.009 - 0.0001 * 1j
kzmin = 0.01
kzmax = 0.2
input = nhdspy.InputParams([electrons, protons], omega_0, theta_kB,
                           va, kzmin, kzmax)
output = nhdspy.run(input)

fig, ax = plt.subplots()
ax.plot(output.kz, output.omega_real, label='Real part')
ax.plot(output.kz, output.omega_imag, label='Imaginary part')
ax.set_xlabel('$kd_{p}$')
ax.set_ylabel(r'$\omega / \Omega_{p}$')
ax.legend()

plt.show()
```



3.1 Code reference

3.1.1 nhdspy Package

Functions

format_input_file(input)	Function to create input file.
run(input)	Run the dispersion solver for a given input.

format_input_file

`nhdspy.`**`format_input_file`**(*input*)

Function to create input file.

Parameters

input [InputParams]

run

`nhdspy.run(input)`

Run the dispersion solver for a given input.

Parameters

input [InputParams]

Classes

InputParams(species, omega_guess, . . . [, . . .])	Input parameters for calculating a dispersion relation.
Result(input, run_output)	Result of running the dispersion solver.
Species(q, m, n, v_d, t_ani, beta_par)	A single bi-Maxwellian species.

InputParams

class `nhdspy.InputParams`(*species*, *omega_guess*, *propagation_angle*, *va*, *kzmin*, *kzmax*, *kzsteps=200*, *numiter=1000*, *det_D_threshold=1e-16*, *n_bessel=1000*, *bessel_zero=1e-50*, *vxsteps=100*, *vysteps=100*, *vzsteps=100*)

Bases: `object`

Input parameters for calculating a dispersion relation.

Parameters

species [list of *Species*] List of the particle species for which to calculate the dispersion. **Note** that the order of the species matters, as some of the parameters below are defined relative to the first species in the *species* list.

omega_guess [complex] Initial guess for the frequency, normalised to the gyro frequency of the first species.

propagation_angle [float] Propagation angle to calculate dispersion relation for. Defined with respect to the magnetic field.

va [float] Ratio of the Alfvén speed to the speed of light. The number density in the Alfvén speed is taken from the first species.

kzmin [float] Start of kz range.

kzmax [float] End of kz range.

Other Parameters

kzsteps [int, optional] Number of steps in kz to calculate dispersion relation at. Points are linearly spaced between *kzmin* and *kzmax*.

numiter [int, optional] Maximum number of iterations in the Newton method

det_D_threshold [float, optional] Threshold for the determinant of the dispersion tensor. If det D <= det_D_threshold, the Newton iteration will be stopped.

n_bessel [int, optional] Maximum of sum in Bessel functions (both regular and modified). Can be very low (e.g., 3) for quasi-parallel propagation.

bessel_zero [float, optional] If I_n is less than this value, higher n are neglected.

vxsteps, vysteps, vzsteps [int, optional] Steps in the x,y,z directions.

Attributes

nspecies Number of species.

total_charge Total charge density.

total_current Total current density.Result is normalised to the first species' charge, number density, and the Alfvén speed.

Attributes Summary

nspecies	Number of species.
total_charge	Total charge density.
total_current	Total current density.Result is normalised to the first species' charge, number density, and the Alfvén speed.

Attributes Documentation

nspecies
Number of species.

total_charge
Total charge density. Result is normalised to the first species' charge and number density.

total_current
Total current density.Result is normalised to the first species' charge, number density, and the Alfvén speed.

Result

class nhdspy.**Result**(*input*, *run_output*)
Bases: `object`

Result of running the dispersion solver.

Attributes

EyEx The electric field component ratio E_y/E_x.

EzEx The electric field component ratio E_z/E_x.

kz Wave vector normalised to the proton intertial length (kd_p).

omega_imag Imaginary part of frequency normalised to the proton gyro-frequency (γ/Ω_p).

omega_real Real part of frequency normalised to the proton gyro-frequency (ω/Ω_p).

Attributes Summary

EyEx	The electric field component ratio E_y/E_x.
EzEx	The electric field component ratio E_z/E_x.
kz	Wave vector normalised to the proton intertial length (kd_p).

Continued on next page

Table 4 – continued from previous page

omega_imag	Imaginary part of frequency normalised to the proton gyro-frequency (γ/Ω_p).
omega_real	Real part of frequency normalised to the proton gyro-frequency (ω/Ω_p).

Attributes Documentation

EyEx
The electric field component ratio E_y/E_x. Returned as complex numbers.

EzEx
The electric field component ratio E_z/E_x. Returned as complex numbers.

kz
Wave vector normalised to the proton intertial length (kd_p).

omega_imag
Imaginary part of frequency normalised to the proton gyro-frequency (γ/Ω_p).

omega_real
Real part of frequency normalised to the proton gyro-frequency (ω/Ω_p).

Species

class nhdspy.**Species**(*q*, *m*, *n*, *v_d*, *t_ani*, *beta_par*)
Bases: `object`

A single bi-Maxwellian species.

Parameters

- **q** [float] Particle charge, in units of proton charge. e.g. for a proton enter `1`, for an electron enter `-1`.
- **m** [float] Particle mass, in units of proton mass. e.g. for a proton enter `1`.
- **n** [float] Number density, relative to the number density used to define the Alfvén speed.
- **v_d** [float] Drift speed as a fraction of the Alfvén speed.
- **t_ani** [float] Temperature anisotropy (perpendicular temperature divided by parallel temperature).
- **beta_par** [float] Parallel beta (thermal pressure divided by magnetic pressure).

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

nhdspy, 8

Index

E

EyEx (nhdspy.Result attribute), 11
EzEx (nhdspy.Result attribute), 11

F

format_input_file() (in module nhdspy), 8

I

InputParams (class in nhdspy), 9

K

kz (nhdspy.Result attribute), 11

N

nhdspy (module), 8
nspecies (nhdspy.InputParams attribute), 10

O

omega_imag (nhdspy.Result attribute), 11
omega_real (nhdspy.Result attribute), 11

R

Result (class in nhdspy), 10
run() (in module nhdspy), 9

S

Species (class in nhdspy), 11

T

total_charge (nhdspy.InputParams attribute), 10
total_current (nhdspy.InputParams attribute), 10