

Simple PDRK User Guide

-A powerful new kinetic plasma dispersion relation solver

Hua-sheng XIE (谢华生), huashengxie@gmail.com

2018-10-27

Introduction

- Why kinetic dispersion relation important? – Waves and instabilities are one of the most important feature of plasma.
- Why difficult to solve? –Many branches, difficult to convergent in some parameters.
- What is PDRK?– The first kinetic plasma dispersion relation solver that can give all the important solutions at one time without requiring initial guess for root finding.

Equations

PDRK v181027 solves uniform plasma dispersion relation with **loss-cone drift bi-Maxwellian** equilibrium distribution function.

Using Matlab, via matrix transformation method.
(Also python version provided by Dr. Xin TAO at USTC)

For $D(\omega, \mathbf{k})=0$, give \mathbf{k} , solve series $\omega(s)$.

$$f_{s0}(v_{\parallel}, v_{\perp}) = f_{\perp}(v_{\perp}) f_z(v_{\parallel}) \quad (1)$$

$$= \frac{1}{\pi^{3/2} v_{zts} v_{\perp ts}^2} \exp \left[-\frac{(v_{\parallel} - v_{ds})^2}{v_{zts}^2} \right] \left\{ \Delta_s \exp \left(-\frac{v_{\perp}^2}{v_{\perp ts}^2} \right) + \frac{1 - \Delta_s}{1 - \alpha_s} \left[\exp \left(-\frac{v_{\perp}^2}{v_{\perp ts}^2} \right) - \exp \left(-\frac{v_{\perp}^2}{\alpha_s v_{\perp ts}^2} \right) \right] \right\},$$

The dispersion relation is

$$|\mathbf{D}(\omega, \mathbf{k})| = \begin{vmatrix} K_{xx} - \frac{c^2 k^2}{\omega^2} \cos^2 \theta & K_{xy} & K_{xz} + \frac{c^2 k^2}{\omega^2} \sin \theta \cos \theta \\ K_{yx} & K_{yy} - \frac{c^2 k^2}{\omega^2} & K_{yz} \\ K_{zx} + \frac{c^2 k^2}{\omega^2} \sin \theta \cos \theta & K_{zy} & K_{zz} - \frac{c^2 k^2}{\omega^2} \sin^2 \theta \end{vmatrix} = 0, \quad (10)$$

The standard linearized kinetic theory gives [Ichimaru1973, p51]

$$\mathbf{K}(\omega, \mathbf{k}) = \left(1 - \frac{\omega_p^2}{\omega^2} \right) \mathbf{I} + \sum_s \frac{\omega_{ps}^2}{\omega^2} \sum_{n=-\infty}^{\infty} \int d\mathbf{v} \frac{\mathbf{\Pi}_s}{\omega - k_{\parallel} v_{\parallel} - n\Omega_s} \left(\frac{n\Omega_s}{v_{\perp}} \frac{\partial f_{s0}}{\partial v_{\perp}} + k_{\parallel} \frac{\partial f_{s0}}{\partial v_{\parallel}} \right), \quad (11)$$

and

$$\mathbf{\Pi}_s = \begin{pmatrix} \left(\frac{n\Omega_s}{k_{\perp}} J_n \right)^2 & i \frac{n\Omega_s}{k_{\perp}} v_{\perp} J_n J'_n & \frac{n\Omega_s}{k_{\perp}} v_{\parallel} J_n^2 \\ -i \frac{n\Omega_s}{k_{\perp}} v_{\perp} J_n J'_n & (v_{\perp} J'_n)^2 & -i v_{\perp} v_{\parallel} J_n J'_n \\ \frac{n\Omega_s}{k_{\perp}} v_{\parallel} J_n^2 & i v_{\perp} v_{\parallel} J_n J'_n & (v_{\parallel} J_n)^2 \end{pmatrix}, \quad (12)$$

where $\int d\mathbf{v} \equiv 2\pi \int_0^{\infty} v_{\perp} dv_{\perp} \int_{-\infty}^{\infty} dv_{\parallel}$, Bessel function $J_n = J_n(\frac{k_{\perp} v_{\perp}}{\Omega_s})$ and $\omega_p^2 = \sum_s \omega_{ps}^2$.

Solvers compare

	Initial guess?	Fast?	Support high harmonic mode?	Separate modes?	All solutions?
WHAMP [Ronnmark1982]	Must	Fast	Difficult	Difficult	No
NHD [Verscharen 2018]	Must	Middle	?	?	No
HOTRAY [Horne1989]	Must	Middle	Difficult	Difficult	No
PDRK [Xie2016]	Not required	Middle	Easy	Easy	Yes

What makes PDRK attractive? It solves the difficulty of root finding, i.e., not requires initial guess and can give all the important solutions at one time. You **do not need luck** any more.

At v181027, PDRK have support loss cone drift bi-Maxwellian distribution, for both electromagnetic and electrostatic cases.

Steps to run PDRK

- 1. Set species parameters in 'pdrk.in' ;
- 2. Set 'setup.m' , B0, k, theta, etc;
- 3. Run 'main.m' ;
- 4. After run 'plot_all.m' , zoom in and select which branch(es) to further plot;
- 5. Set the 'wpdat' in 'pdrk_wpdat.m' , 'plot_select.m' will search the solutions in the same branches in 'wpdat' , and then store and plot them;
- 6. If you require polarization info, run 'output.m'

Note, v181027 has combined all three models (em3d, es3d, es1d) in PDRK Xie2016PST original paper to a single version.

- ✓ EM3D: iem=1;
- ✓ ES3D: iem=0;
- ✓ ES1D: iem=0, theta=0.

Thus, it is extremely simple to switch between em run and es run.

Typical cases 1: Cold plasma

pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
1	1	8.7e6	2.857e-3	2.857e-3	1.0	1.0	0.0
-1	5.447e-4	8.7e6	2.857e-3	2.857e-3	1.0	1.0	0.0

B0=100.0E-9; N=1; J=8; iem=1;

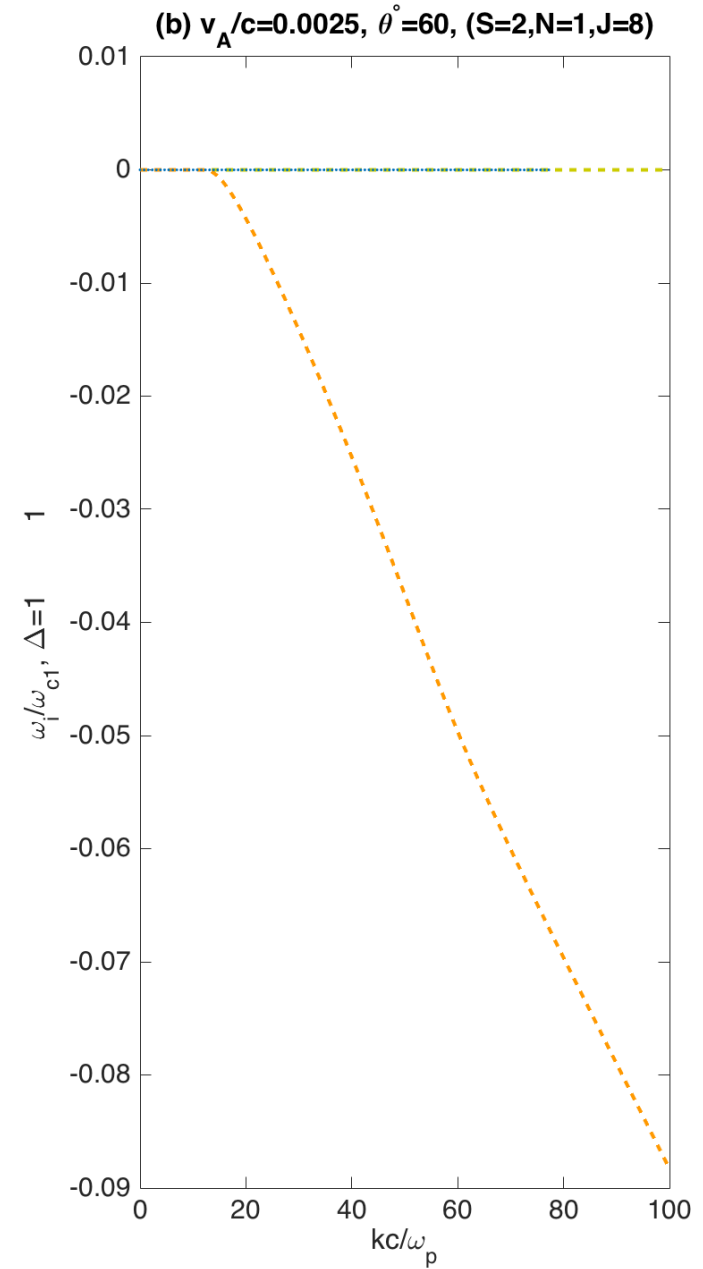
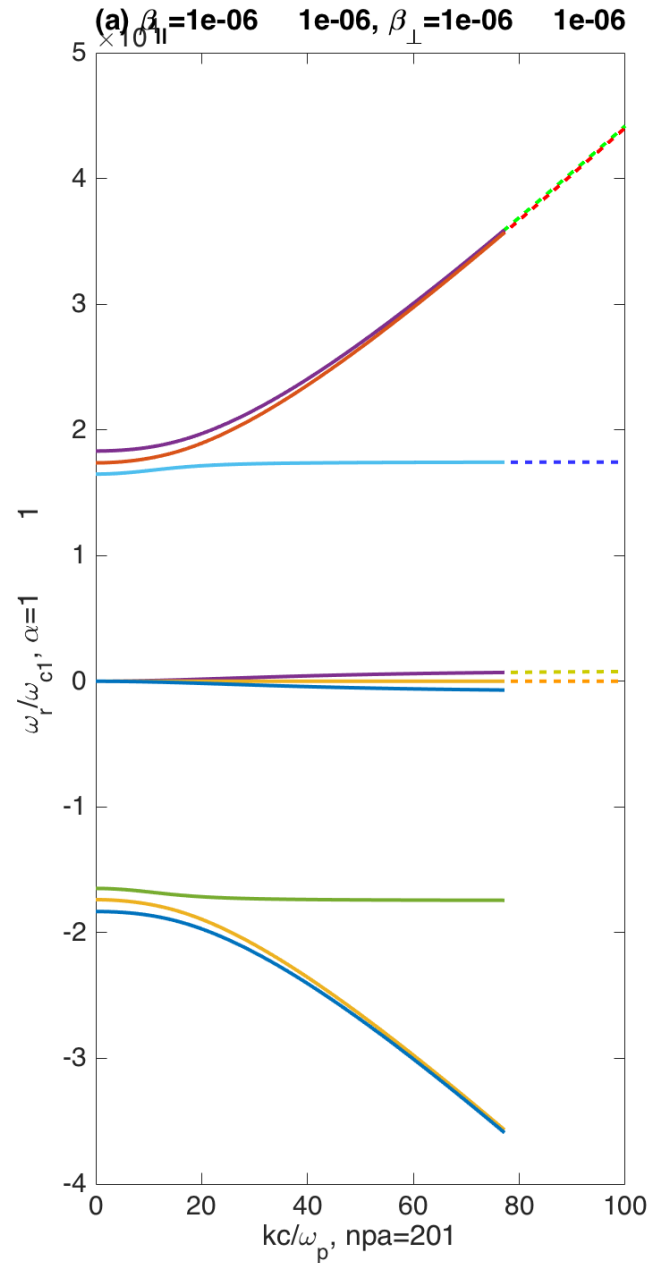
(ipa,ipb) = (1,1) scan k, fixed theta=60

pa=0:0.5:100

Solid lines: Fluid solver
PDRF results.

Dash lines: PDRK results.

We find good agreement,
except a slight difference
at large k for the ion
cyclotron wave, which is
damped due to kinetic
effect.



Typical cases 2: Loss cone mirror

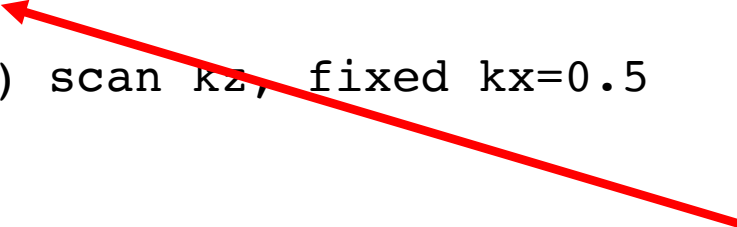
pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
1	1	1.e6	24840.	49680.	0.5	0.1	0.0
-1	5.447e-4	1.e6	24840.	24840.	0.5	0.1	0.0

B0=100.0E-9; **N=3**; J=8; iem=1;

(ipa,ipb) = (3,3) scan kz, fixed kx=0.5

pa=0:0.01:1



Ensure your results are convergent
by trying a larger N.

After run main.m and plot_all.m, zoom in the Fig(b) to select 'wpdat' .

Find a point in 'plot_all' figure and set it into 'pdrk_wpdat.m'

Current Folder

- examples
- input
- modules
- output
- pdrk_main.m

Editor - /ENN/project/pdrk/v2_loss-cone/code/181017/code/input/pdrk_wpdat.m

```
1 % 18-10-19 17:56 Hua-sheng XIE, huashengxie@gmail.com, FRI-ENN, China
2 % Ackn.: Richard Denton (Dartmouth), Xin Tao (USTC), Jin-song Zhao (PMO),
3 % etc ...
4 % Initial data for run pdrk
5
6 % Search the most close d
7 % Initial data for find t
8 % Please use pdrk_plot_all
9 % modify here the initial
10
11 % wpdat(:,1) is pa; wpdat
12 % wpdat(:,3) is Re or Im(
13
14 wpdat=[0.44,0,0.057i;
15        0.22,0,0.051i;
16        0.47,0,-0.1583i;
17        0.41,0,-0.2963i;
18        ];
```

Workspace

Name	Value
bzj	[-0.0173 - 0.046..
c2	8.9
cnj	-5.
col	8
csnj	1x
cSs1	2.1
cwp	2.2770e+05
czj	[2.2377 - 1.6259..
d	345x1 complex d.
d0	345x1 sym
Deltas	[0.1000,0.1000]
dpa	0.0100
dpb	5
epsilon0	8.8542e-12
figstr	'S=2_J=8_N=3_n..
Gamn	5.2642e-14
Gamnp	1.1594e-09
h	1x1 Figure
iab	2

Figure 1

(a) $\beta_{||}=1$, $1, \beta_{\perp}=2$

(b) $v_A/c=0.0073$, $k_x c/\omega_p=0.5$

Plot (a) shows ω_i/ω_{ci} vs $k_z c/\omega_p$ for $\beta_{||}=1$ and $1, \beta_{\perp}=2$. Plot (b) shows ω_i/ω_{ci} vs $k_z c/\omega_p$ for $v_A/c=0.0073$ and $k_x c/\omega_p=0.5$. A point is selected in plot (b) with coordinates X: 0.44, Y: 0.05799.

----- ! Set the 1st species to be ion in 'pdrk.in', if
----- ! you hope wcs1=omega_ci and cwp=c/omega_pi.
wcs1 [1st species cyclotron frequency, Hz] = 9.5791
wps1 [1st species plasma frequency, Hz] = 1316.5763
cwp [c/wps1, m] = 227704.2438

run ./modules/pdrk_kernel.m ...
use 18.1675 s, toal 18.431s
run ./modules/pdrk_plot_all.m ...
use 3.2443 s, toal 21.6753s

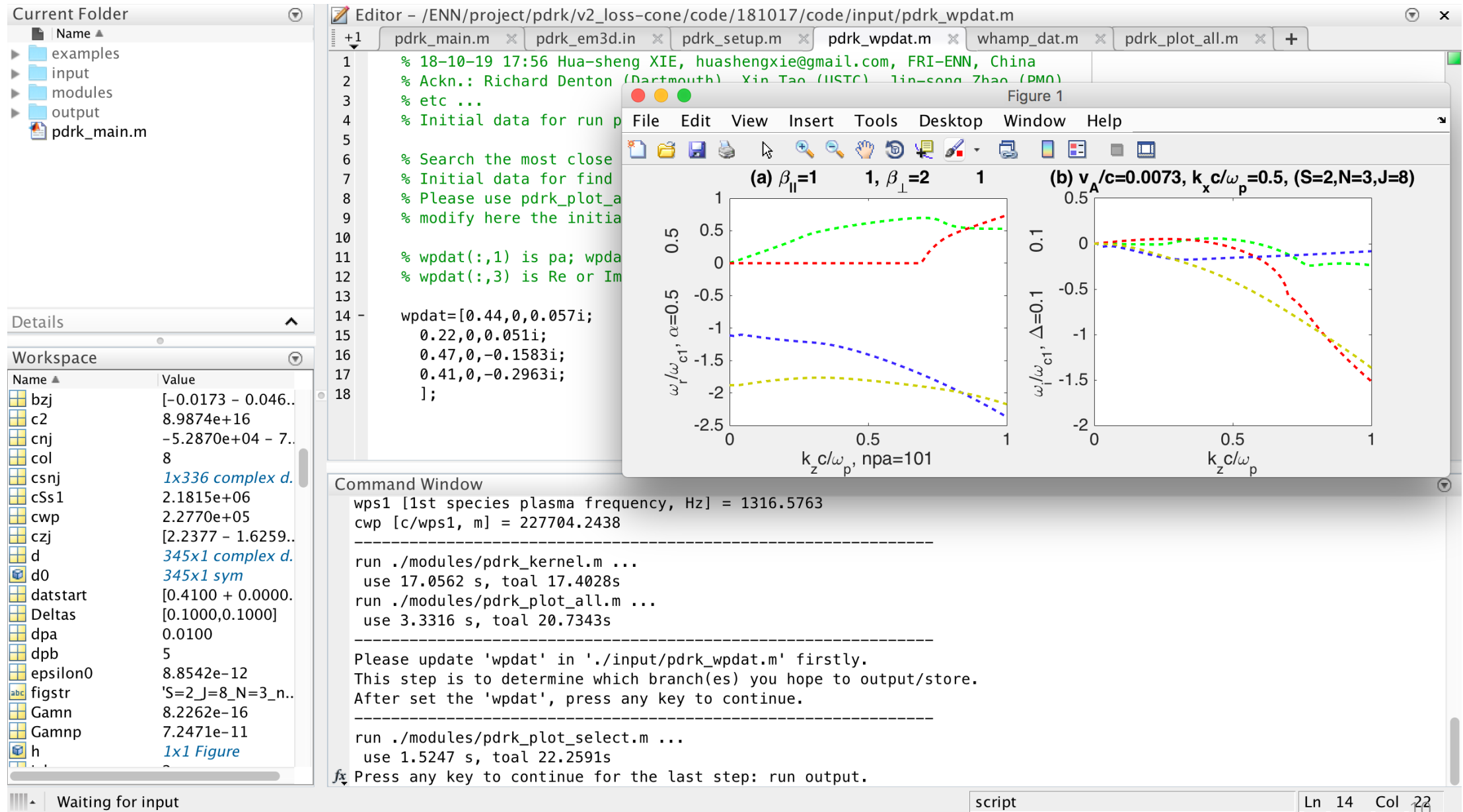
Please update 'wpdat' in './input/pdrk_wpdat.m' firstly.
This step is to determine which branch(es) you hope to output/store.
After set the 'wpdat', press any key to continue.

Waiting for input

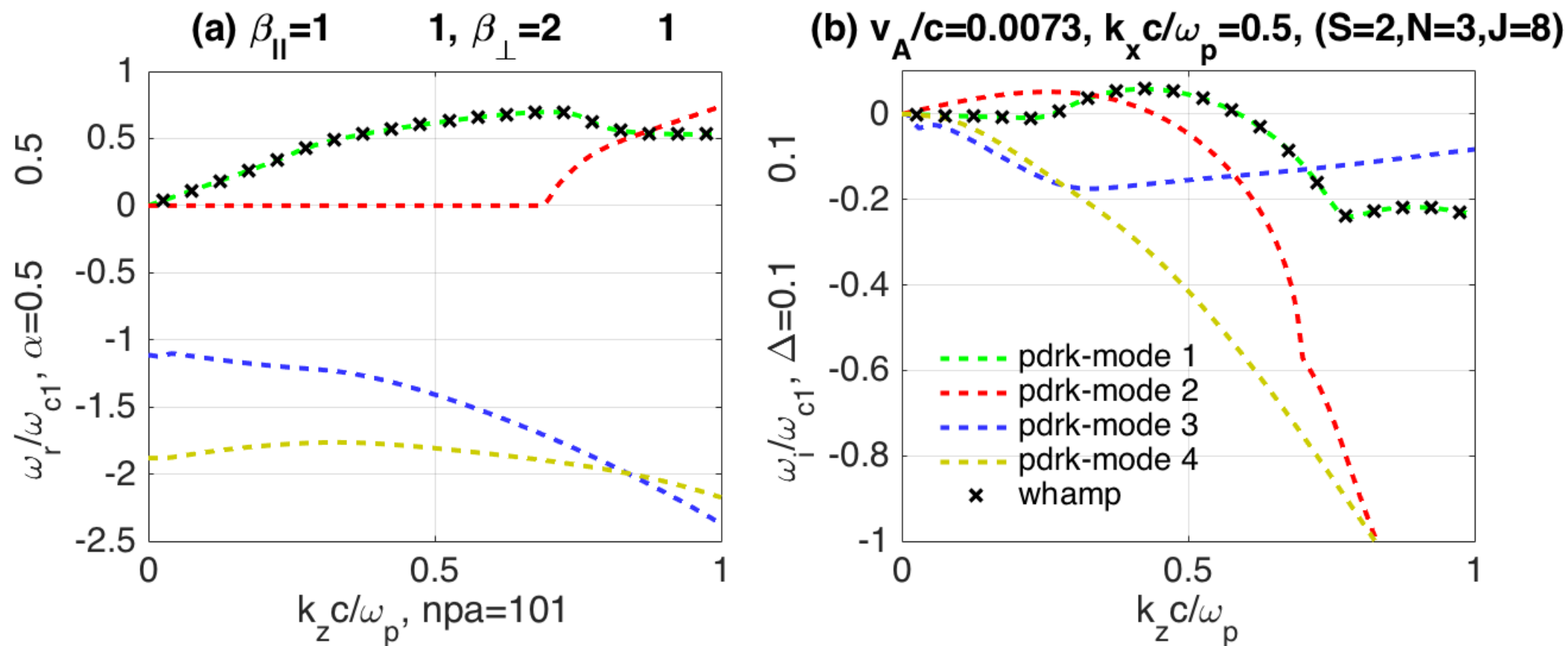
script

Ln 14 Col 89

After set 'wpdat' , press enter, we obtain the follow figure.



Compare to WHAMP data, we find good agreement. However, WHAMP can only find one solution at one time and require good initial guess for root finding.



Typical cases 3: Parallel Multi-species Beam mode

pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
1	1	2.528e5	3.5387e4	3.5387e4	1.0	1.0	0.0
-1	5.447e-4	3.16e5	2.831e4	2.831e4	1.0	1.0	3.7013e-3
1	1	3.16e4	28.31e4	28.31e4	1.0	1.0	3.7013e-2
1	2	3.16e4	28.31e4	28.31e4	1.0	1.0	0.0

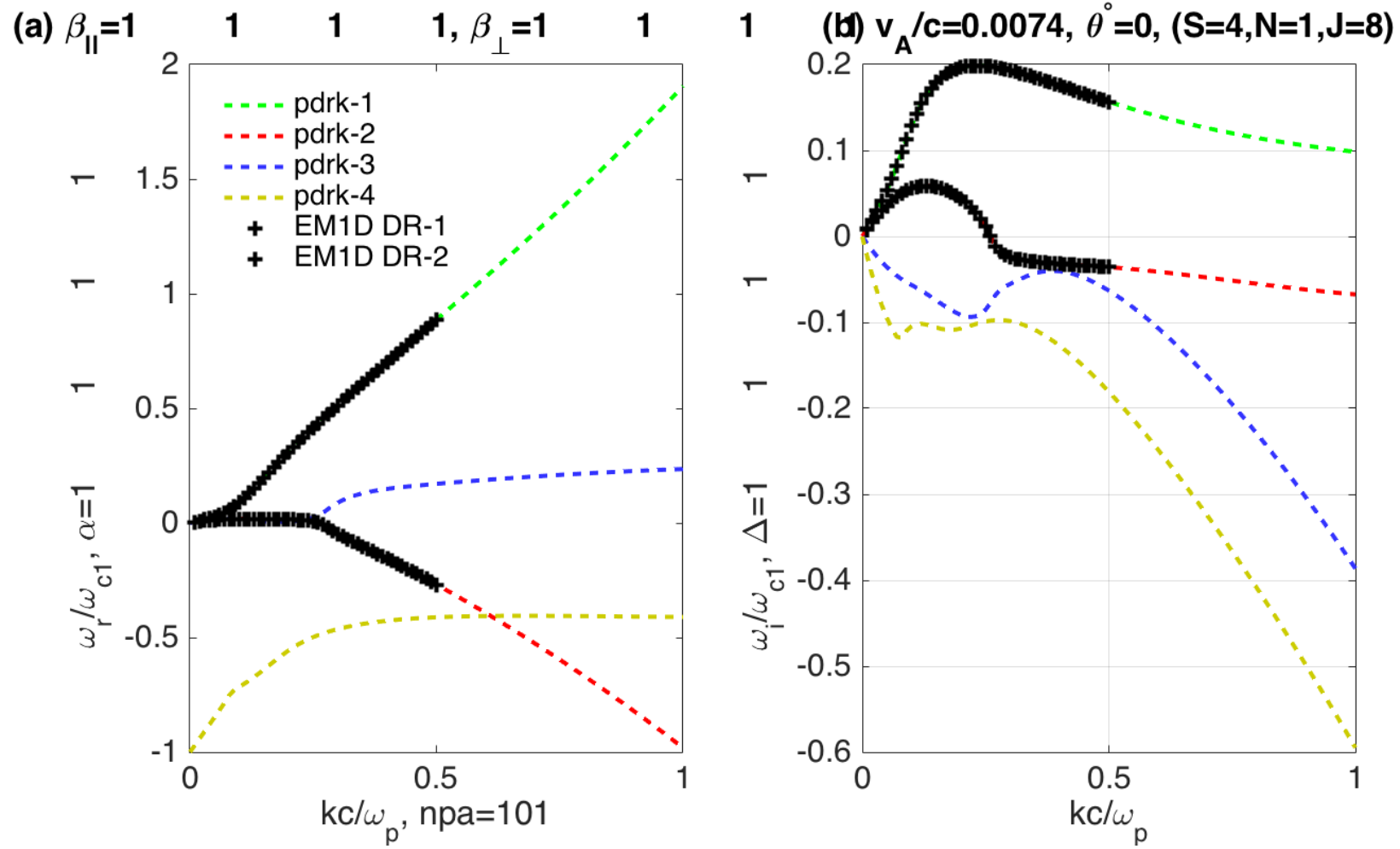
```
B0=60.0E-9; N=1; J=8; iem=1;
```

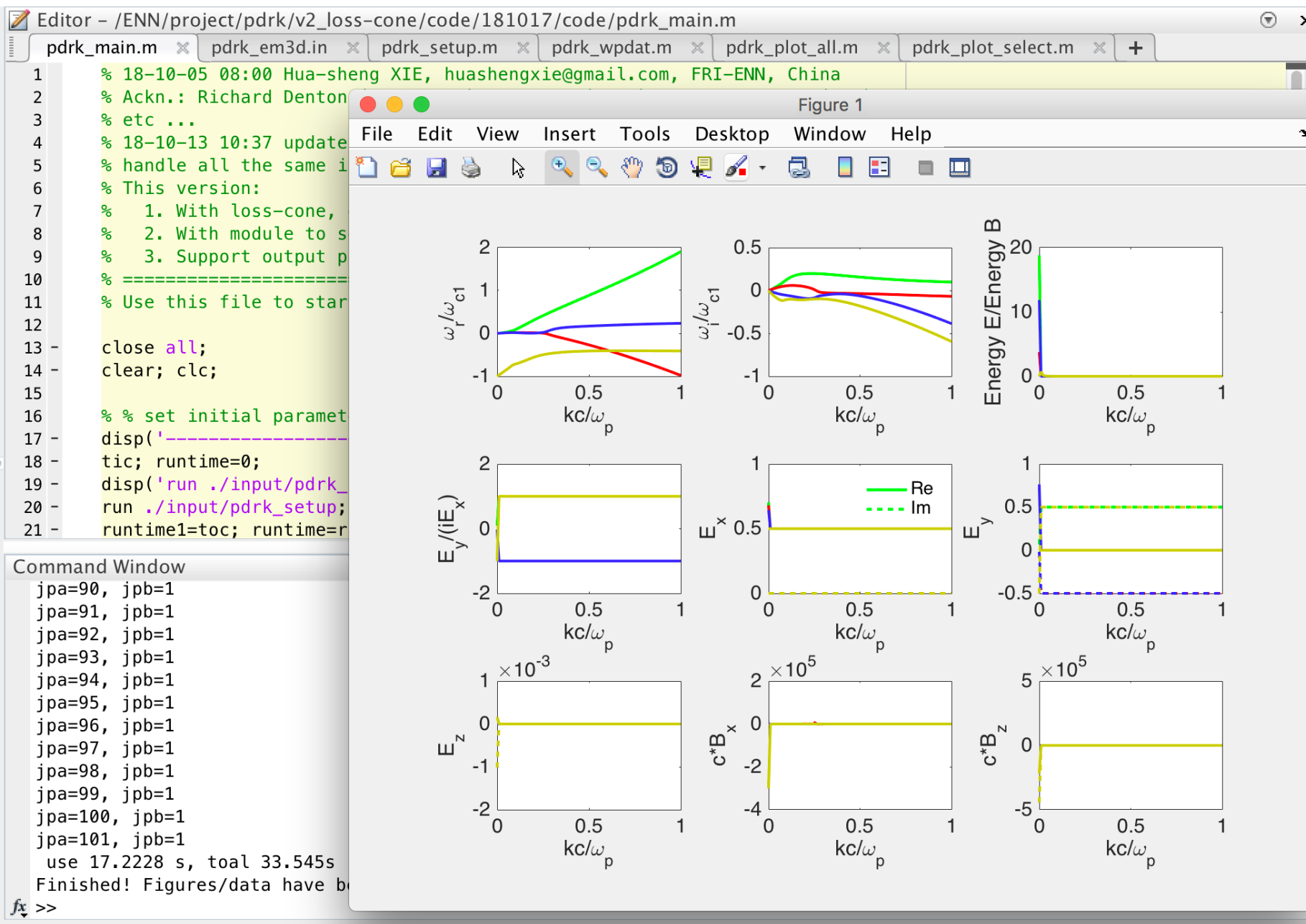
```
(ipa,ipb) = (1,1) scan k, fixed theta=0
```

```
pa=0:0.01:1;
```

```
iout=2;
```

Good agreement to EM1D theta=0 dispersion relation solutions.





lout=2, polarization are also calculate.

We find $E_y/(iE_x)=+1$ or -1 , i.e., only left and right-hand polarized modes. Agree with theory.

Typical cases 4: Dispersion surface

pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
1	1	5.e6	12.94	12.94	1.0	1.0	0.0
-1	5.447e-4	5.e6	12.94	12.94	1.0	1.0	0.0

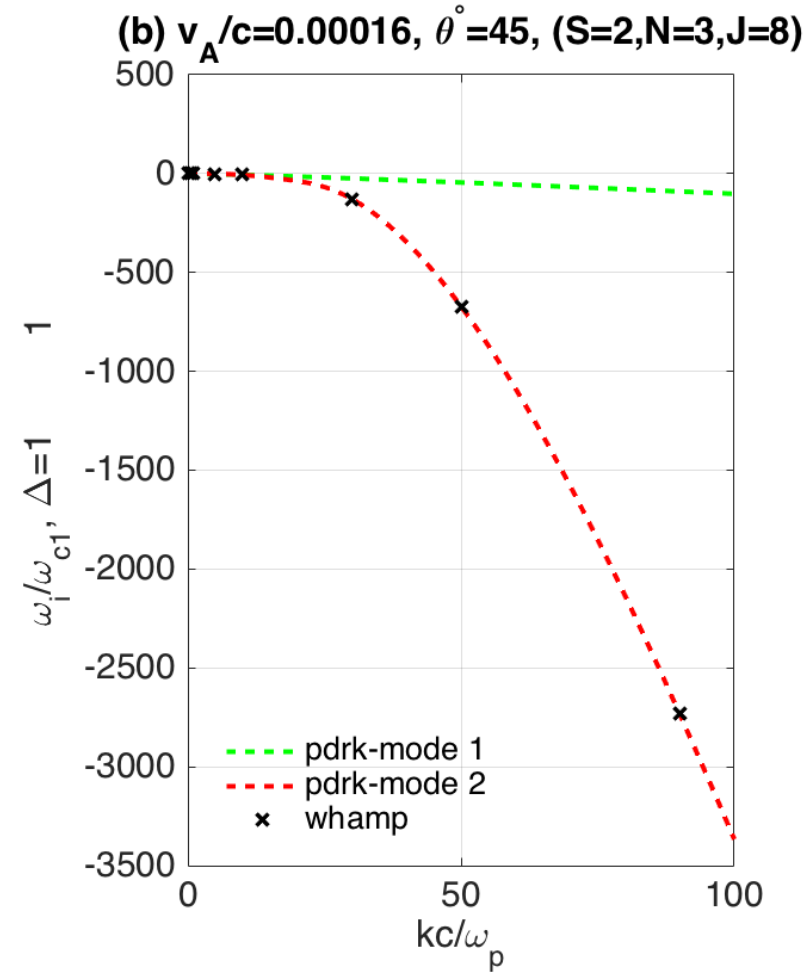
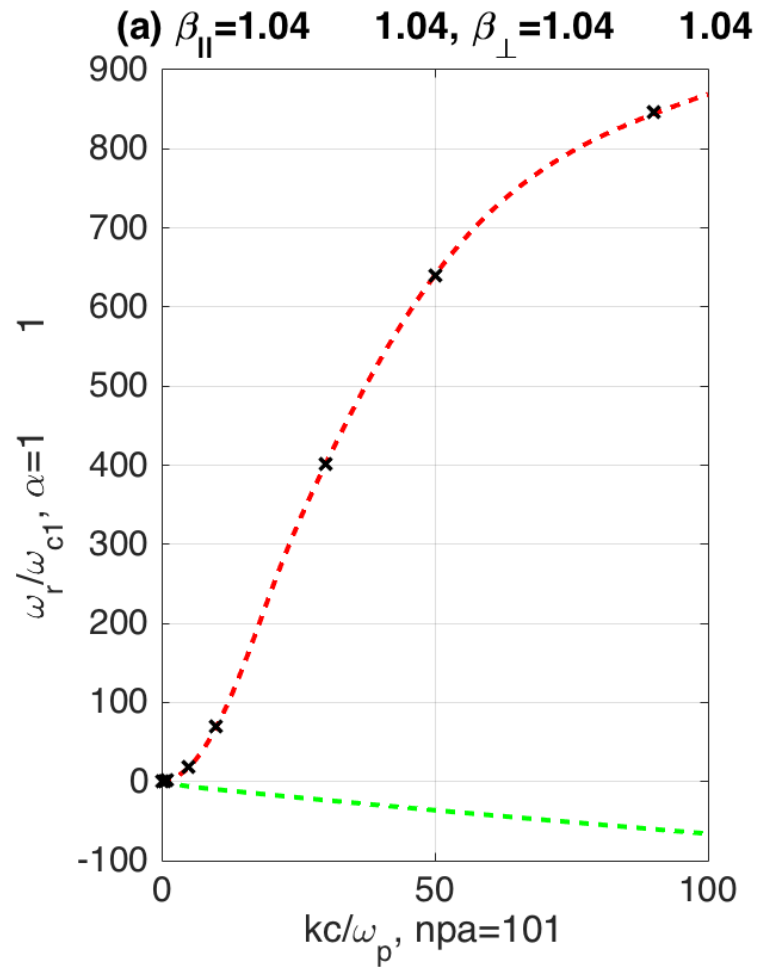
```
B0=5.0E-9; N=3; J=8; iem=1;
```

```
(ipa,ipb) = (1,2) scan 2D (k, theta)
```

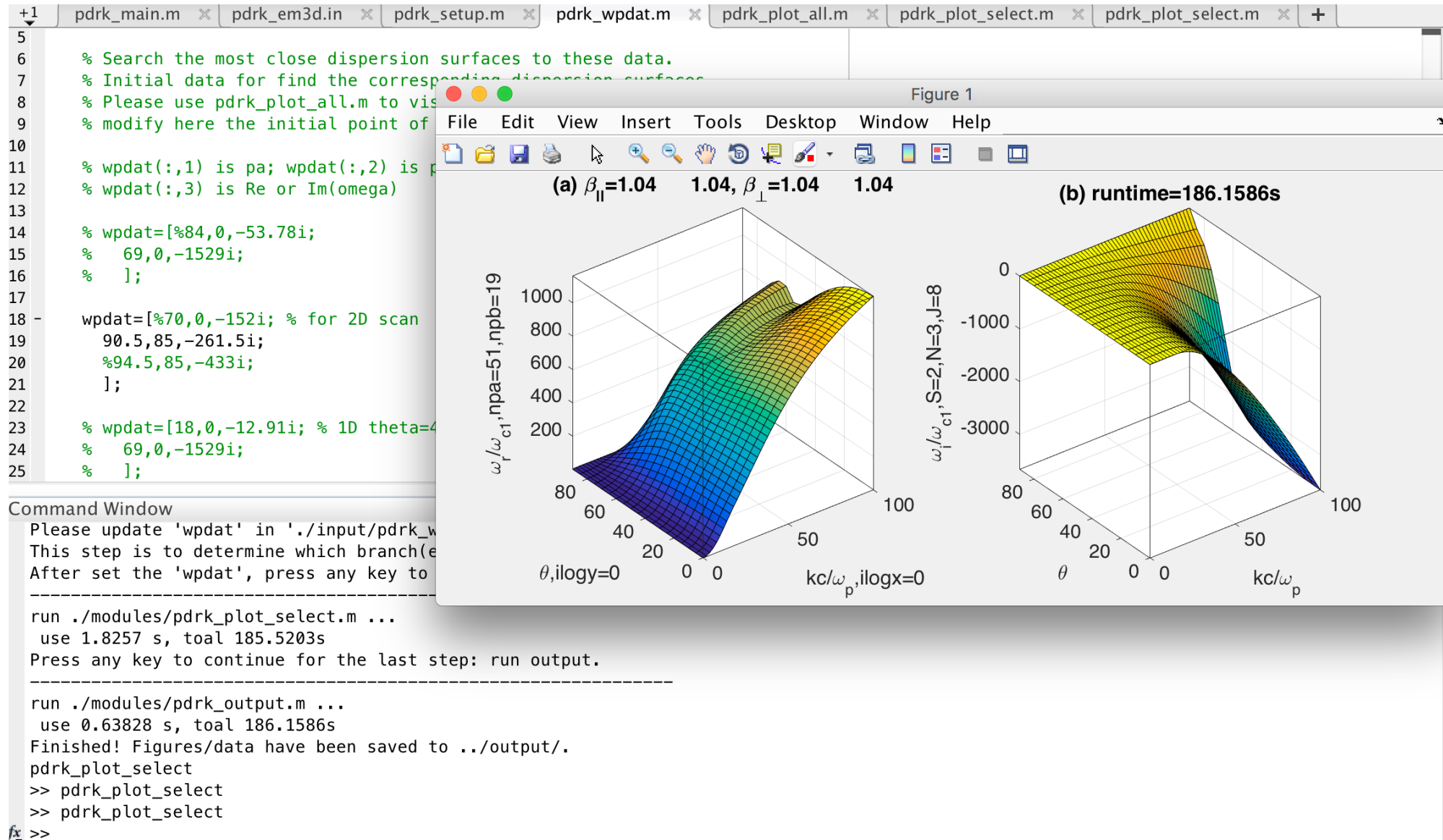
```
pa=0:2:100;pb=0:5:90;
```

```
iout=1;
```

Firstly, $\theta=45$, 1D scan k agrees with whamp.



The 2D scan with proper 'wpdat' gives a nice whistler wave dispersion surface. Give multi 'wpdat' can also plot other dispersion surfaces.



Typical cases 5: ES3D loss cone instability

pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
-1	5.447e-4	1.e6	1.	1.	1.0	1.0	0.0
-1	5.447e-4	1.e6	5.e2	5.e2	0.005	0.0	0.0

```
B0= 143.5E-9; N=1; J=8; iem=0;
```

```
(ipa,ipb) = (1,1) scan k, fixed theta=88.5
```

```
pa=41:10:1700
```

```
iout=1;
```

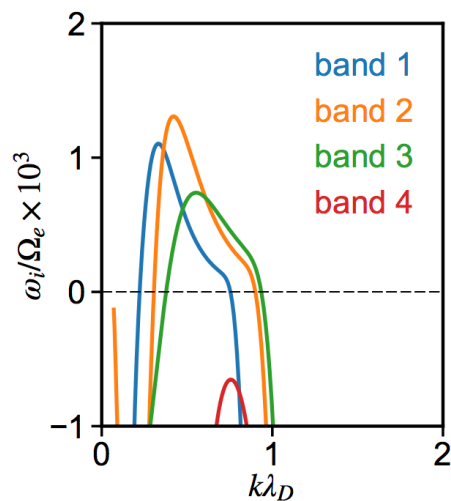
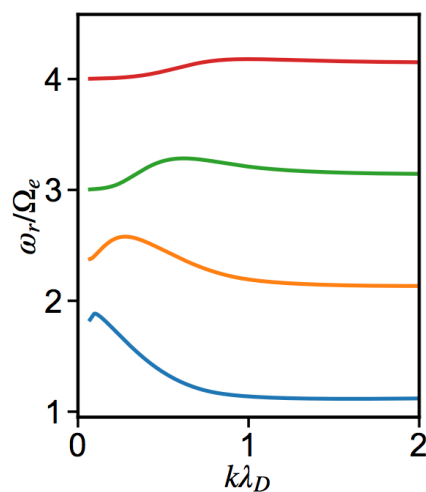
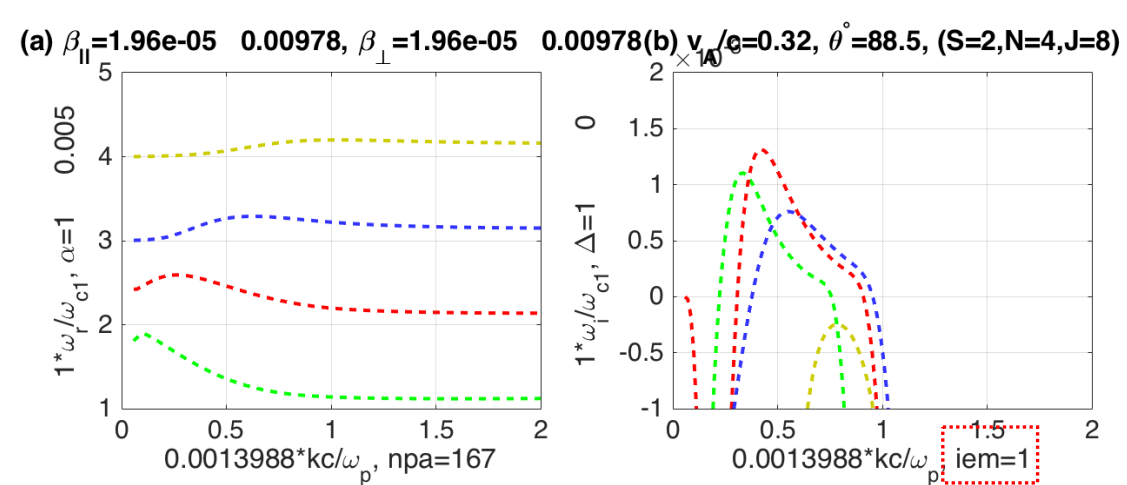
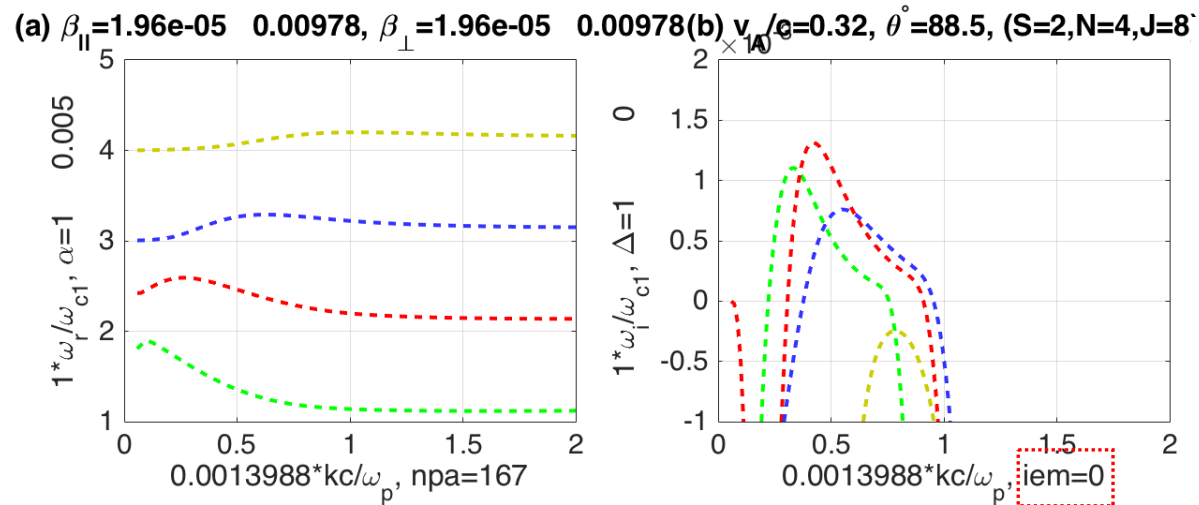
To normalized to cold λ_{De} ,
we have rescaled the k-axis.

```
rex=abs(lambdaDs(1)/cwp)=0.001398  
8;
```

Left: pdrk electrostatic run, iem=0.

Right: pdrk electromagnetic run, iem=1.

Down: HOTRAY electrostatic result from X. Tao et al, 2018 (submitted).



HOTRAY agrees with PDRK very well. And the agreement between iem=0 and iem=1 imply that the mode is indeed electrostatic mode.

Typical cases 6: ES1D beam

pdrk.in

qs (e)	ms (mp)	ns (m ⁻³)	Tzs (eV)	Tps (eV)	alphas	Deltas	vds/c
-1	5.447e-4	0.9e6	1.	1.	1.0	1.0	0.0
-1	5.447e-4	0.1e6	1.	1.	1.0	1.0	9.8913e-3

```
B0= 143.5E-9; N=1; J=8; iem=0;
```

```
(ipa,ipb) = (1,1) scan k, fixed theta=0
```

```
pa=0.1:2:400
```

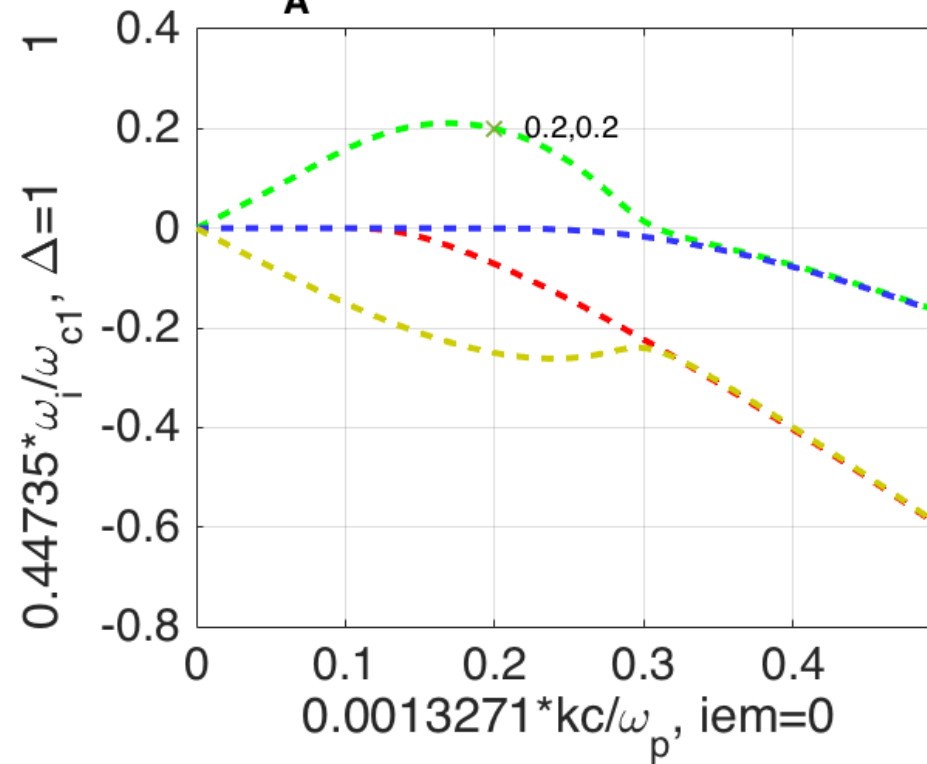
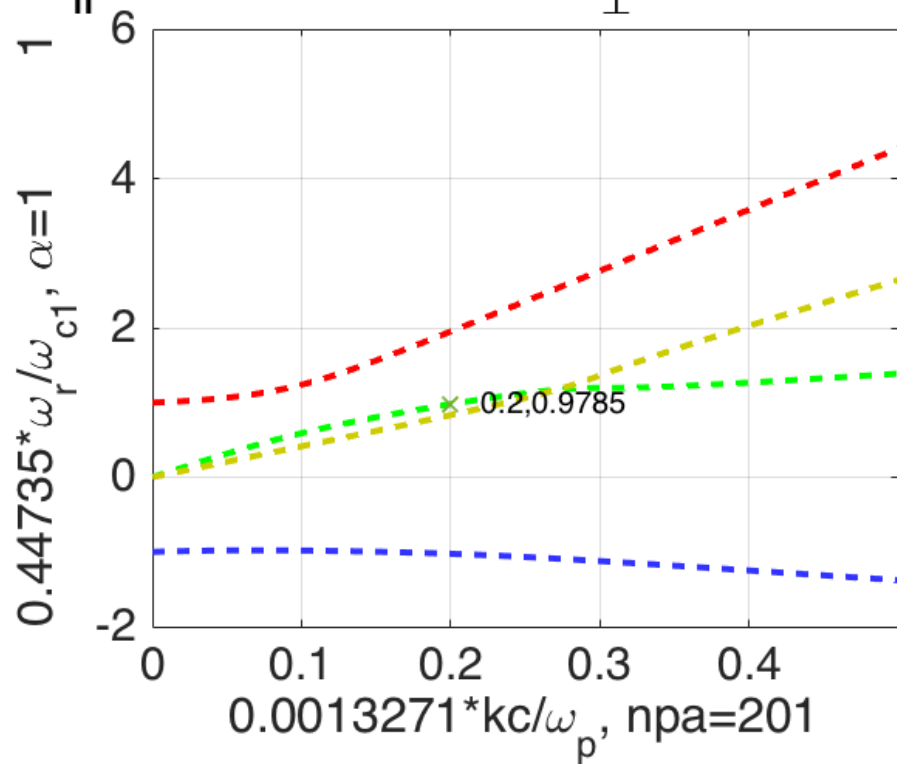
```
iout=1;
```

This is default beam test case in Xie2016PST PDRK paper for ES1D version, i.e., $k_{\text{perp}}=0$, $n_b=0.1n_0$, $n_e=n_0-n_b$ and $v_{ds}(2)/v_{tzs}(2)=5.0$.

To normalized to ω_{pe} and λ_{De} , we have rescaled the k and omega-axis.
 $rex=abs(sqrt(1/sum(1./lambdaDs.^2)))/cwp)=0.0013271$;
 $rez=abs(wcs(1)/sqrt(sum(wps2)))=0.44735$;

Agree well with the original paper, i.e., $k \cdot \lambda_{De} = 0.2$, the most unstable mode $\omega = 0.9785 + 0.2000i$

(a) $\beta_{\parallel} = 1.76e-05$ $1.96e-06$, $\beta_{\perp} = 1.76e-05$ $1.96e-06$ (b) $v_A/c = 0.45$, $\theta^\circ = 0$, (S=2, N=1, J=8)



Enjoy!

If you meet any problems or find pdrk does not agree some benchmarks, please not hesitate to email me (huashengxie@gmail.com), I will long term support this code. Thanks! The suggestions to improve this code are appreciated.

Download: <http://hsxie.me/codes/pdrk/>

You are welcome to rewrite PDRK to other versions or other languages.

If you use this code, please cite:

[Xie2016] Huasheng Xie and Yong Xiao, PDRK: A General Kinetic Dispersion Relation Solver for Magnetized Plasma, Plasma Science and Technology, 18, 2, 97 (2016). Update/Bugs fixed at <http://hsxie.me/codes/pdrk/> or <https://github.com/hsxie/pdrk>.