DIII-D ECH with linear Slab Profiles

Open Additional files:

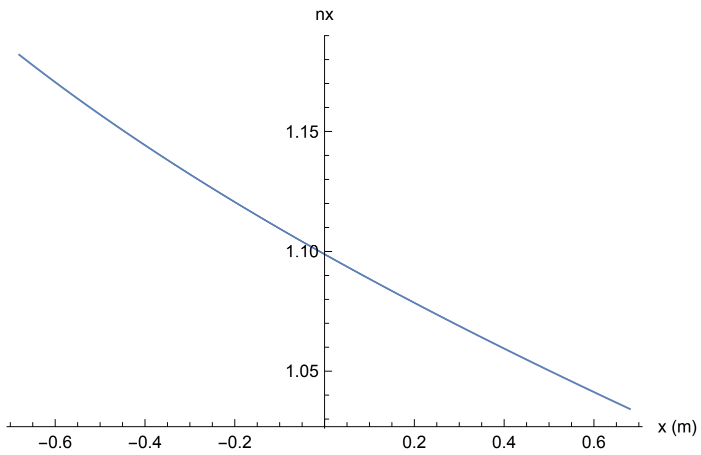
Get dispersion routines by evaluating Disper.nb
Get plotting and printing routines by evaluating PlotPack.nb
Set Parameters by opening a Parameter Window

Note: Slab profile models defined in initialization cells at the bottom of this notebook.

First Do Cold Plasma

Plot Real and Imaginary parts of nx from 2nd order warm plasma dispersion relation (i.e. $E_{\|}\equiv 0$)

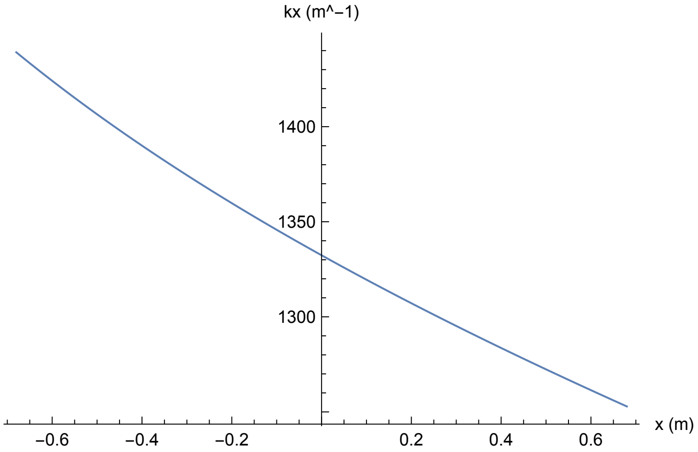
```
nPerpCold[x_] := Module[{ne, b, x0}, x0 = x;
  ne = nprof[x0];
  b = bprof[x0];
  ColdDis0[freq, ne, b, nz, etaList]]
                                            xmax - xmin
nt = Table[{x, nPerpCold[x]}, {x, xmin, xmax, ───────────}];
                                            nPoints - 1
ComplexListPlot[nt, "x (m)", "nx"]
paramPrint[{dataSet, xProfileMin, xProfileMax,
   nXmin, nXmax, BXmin, BXmax, freq, nz, etaList, xmin, xmax}];
```



```
dataSet=DIII-D slab
xProfileMin=-0.68
xProfileMax=0.68
nXmin=2.5 × 10^19
nXmax=3.5 × 10^19
BXmin=1.9
BXmax=2.1
freq=55 990
nz=0.1
etaList={1., 0., 0., 0., 0.}
xmin=-0.68
xmax=0.68
```

Plot Real and Imaginary parts of kx from 2nd order cold plasma dispersion relation (i.e. $E_{\|} \equiv 0$)

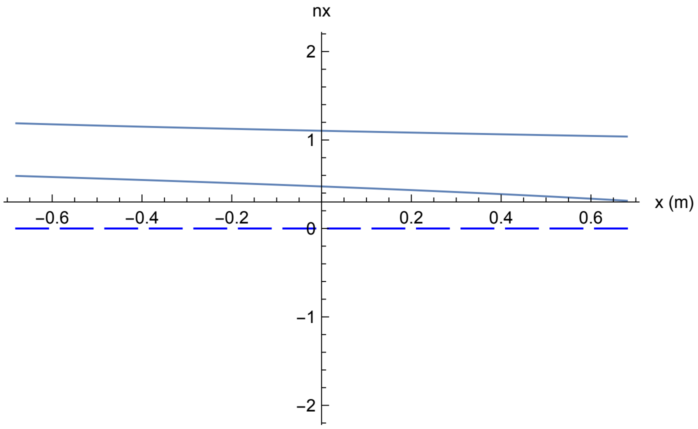
```
nPerpCold[x_] := Module[{ne, b, x0}, x0 = x;
  ne = nprof[x0];
  b = bprof[x0];
  ColdDis0[freq, ne, b, nz, etaList]]
nt = Table[{x, k0 nPerpCold[x]}, {x, xmin, xmax, (xmax - xmin)/(nPoints - 1)}];
ComplexListPlot[nt, "x  (m)", "kx  (m^-1)"]
paramPrint[{dataSet, xProfileMin, xProfileMax,
   nXmin, nXmax, BXmin, BXmax, freq, nz, etaList, xmin, xmax}];
```



```
dataSet=DIII-D slab

xProfileMin=-0.68

xProfileMax=0.68

nXmin=2.5 × 10^19

nXmax=3.5 × 10^19

BXmin=1.9

BXmax=2.1

freq=56 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

xmin=-0.68

xmax=0.68
```

Plot Real and Imaginary parts of nx from 4nd order cold plasma dispersion relation (i.e. fast and slow)

```
nPerp2FS[x_] := Module[{ne, b, x0}, x0 = x;
  ne = nprof[x0];
  b = bprof[x0];
        ColdDis2FS[freq, ne, b, nz, etaList]]
nt2FS = Table[Flatten[{x, nPerp2FS[x]}], {x, xmin, xmax, (xmax - xmin)/(nPoints - 1)}];
nF = Transpose[{Transpose[nt2FS]⟦1⟧, Transpose[nt2FS]⟦2⟧}];
nS = Transpose[{Transpose[nt2FS]⟦1⟧, Transpose[nt2FS]⟦3⟧}];
g1 = ComplexListPlot[nF, "x (m)", "nx"];
g2 = ComplexListPlot[nS, "x (m)", "nx"];
Show[{g1, g2}, PlotRange → {-2., 2.}]
paramPrint[{dataSet, xProfileMin, xProfileMax,
   nXmin, nXmax, BXmin, BXmax, freq, nz, etaList, xmin, xmin}];
```



```
dataSet=DIII-D slab

xProfileMin=-0.68

xProfileMax=0.68

nXmin=2.5 × 10^19

nXmax=3.5 × 10^19

BXmin=1.9

BXmax=2.1

freq=55 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

xmin=-0.68

xmin=-0.68
```

Now Warm Plasma Stuff

Plot Real and Imaginary parts of nx from 6th order warm plasma dispersion relation (expanded to 2nd order in $k_{L}\ \rho$)

```
nPerpWarm6[x_] := Module[{ne, te, b, x0, TL},
        x0 = x;
  ne = nprof[x0];
  b = bprof[x0];
     TL = tprof[x0] * TList;
     WarmDis6[freq, ne, b, nz, etaList, TL]]

nxwarm = Table[Flatten[{x, nPerpWarm6[x]}], {x, xmin, xmax, (xmax - xmin)/(nPoints - 1)}];
roots = rootSort[nxwarm];
rootsRe = Table[Flatten[{roots[[i]][[1]],
      Table[Re[roots[[i]][[j]]], {j, 2, Length[roots[[i]]]}]}], {i, Length[roots]}];

rootsIm = Table[Flatten[{roots[[i]][[1]],
      Table[Im[roots[[i]][[j]]], {j, 2, Length[roots[[i]]]}]}], {i, Length[roots]}];

g6 = ComplexVectorListPlot[roots, "x  (m)", "nx"];
paramPrint[{dataSet, xProfileMin, xProfileMax, nXmin, nXmax,
    BXmin, BXmax, freq, nz, etaList, TList, modelList, xmin, xmax}];
Show[g6, PlotRange → All]
Show[g6, PlotRange → {-1.5, 1.5}]
ComplexVectorListPlot[rootsRe, "x", "Re[kx]", PlotRange → {-2., 2.}]
ComplexVectorListPlot[rootsIm, "x", "Re[kx]"]
```

dataSet=DIII-D slab

xProfileMin=-0.68

xProfileMax=0.68

nXmin=2.5×10^{19}

nXmax=3.5×10^{19}

BXmin=1.9

BXmax=2.1

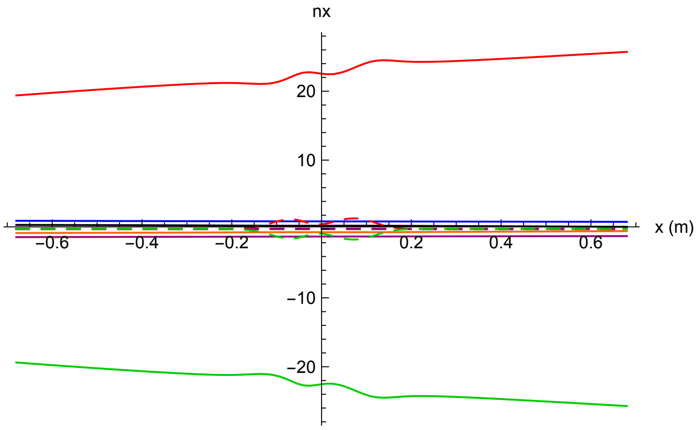
freq=55 990

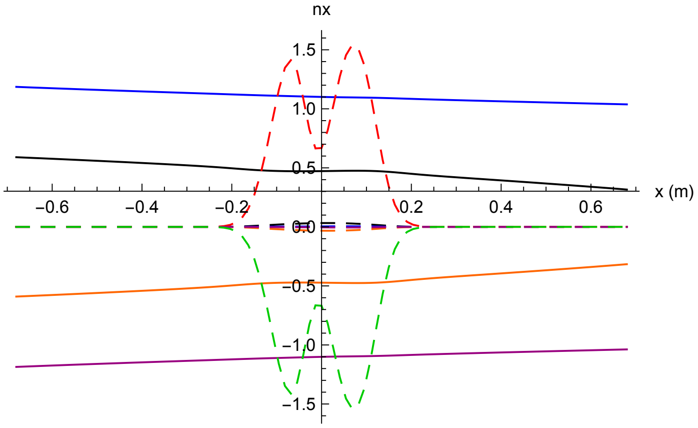
nz=0.1

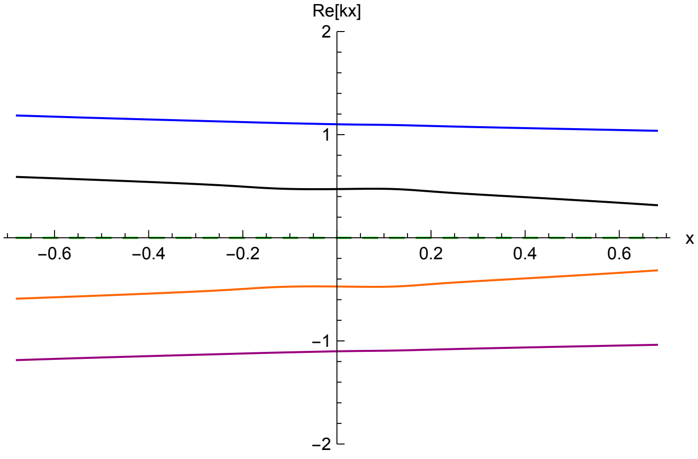
etaList={1., 0., 0., 0., 0.}

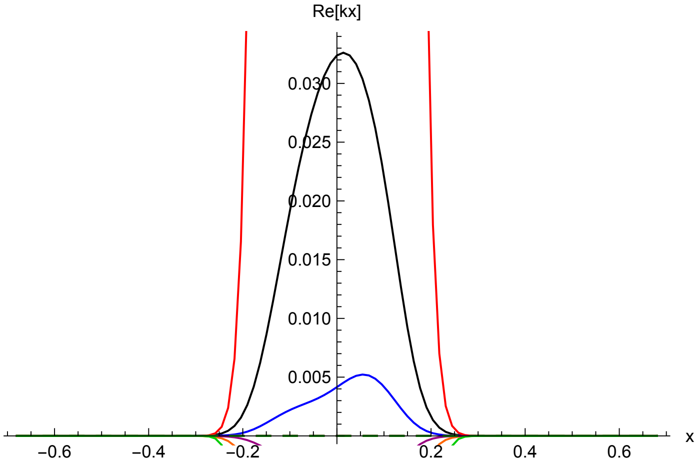
TList={1., 1., 0., 0., 0., 0.}

```
modelList={1, 1, 0, 0, 0, 0}

xmin=-0.68

xmax=0.68
```









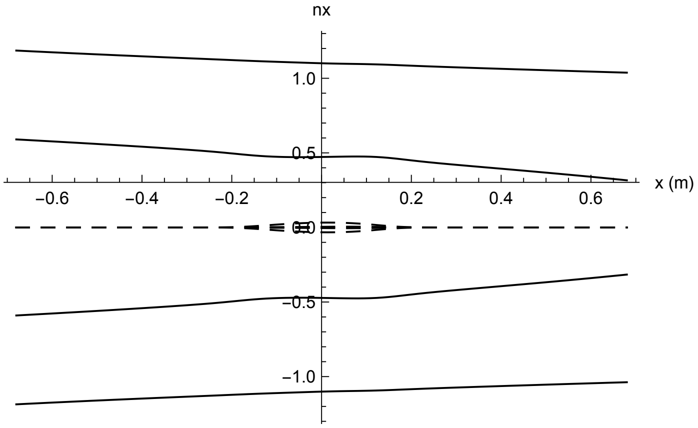
Now Try with Hot Plasma Dispersion using root finder

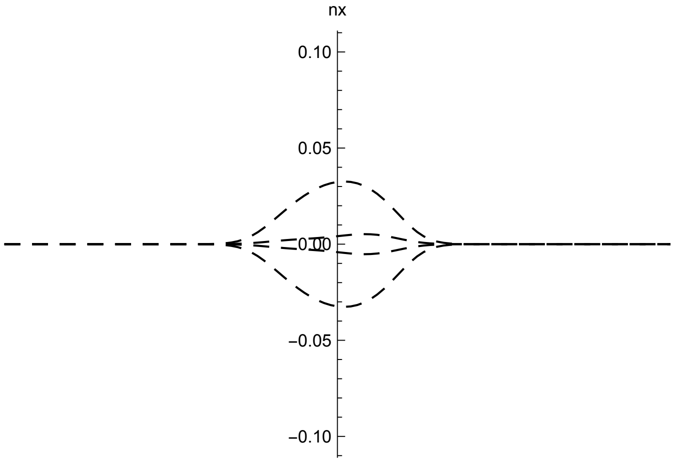
```
nPerpHot[x_, nxGuess_] := Module[{ne, b, x0, TL, nx0},
        x0 = x;
  nx0 = nxGuess;
  ne = nprof[x0];
  b = bprof[x0];
  TL = tprof[x0] * TList;
rootRule = FindRoot[DisFuncGeneral[freq, ne, b, nz, nx, etaList, TL,
     nminList, nmaxList, modelList], {nx, nx0}, MaxIterations -> 30];
  nx /. rootRule]
```

First try root finding on warm plasma dispersion rel (model = 1)

```
modelList = Table[1, {i, 1, 6}];
```

```
nxhot[iRoot_] := Module[{iRoot0, nxWarm, rootsWarm, nxH, x0, ne, b, t, TL},
   nxWarm = Table[Flatten[{x, nPerpWarm6[x]}], {x, xmin, xmax, (xmax - xmin)/(nPoints - 1)}];
   iRoot0 = iRoot;
   rootsWarm = rootSort[nxWarm];
   nxH = Table[0., {i, 1, nPoints}];
   Do[
    (x0 = xmin + (i - 1) (xmax - xmin)/(nPoints - 1);
     nxGuess = rootsWarm[[i]][[iRoot0 + 1]];
     (*Print["x0 = ", x0,"  nxGuess= ",nxGuess];*)
     nxH[[i]] = {x0, nPerpHot[x0, nxGuess]};), {i, 1, nPoints}];
   nxH];
g7 = ComplexVectorListPlot[nxhot[1], "x (m)", "nx"];
g8 = ComplexVectorListPlot[nxhot[2], "x (m)", "nx"];
g9 = ComplexVectorListPlot[nxhot[3], "x (m)", "nx"];
g10 = ComplexVectorListPlot[nxhot[4], "x (m)", "nx"];
g11 = ComplexVectorListPlot[nxhot[5], "x (m)", "nx"];
g12 = ComplexVectorListPlot[nxhot[6], "x (m)", "nx"];
Show[{g7, g8, g10, g11}, PlotRange → All]
Show[{g7, g8, g10, g11}, PlotRange → {-.1, .1}]
paramPrint[{dataSet, ne0, nsep, B, freq, nz, etaList,
   TList, modelList, rmaj, rmin, rsep, sol, alphan, alphaT}];
```





```
dataSet=DIII-D slab

ne0=ne0

nsep=nsep

B=B

freq=55 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

TList={1., 1., 0., 0., 0., 0.}

modelList={1, 1, 1, 1, 1, 1}

rmaj=rmaj

rmin=rmin

rsep=rsep

sol=sol

alphan=alphan

alphaT=alphaT
```

Now try with hot plasma (model = 2) for all species.   Can I find the warm plasma roots with the full dispersion relation?

```
xPoint = 0.;
guesses = nPerpWarm6[xPoint] (* Warm plasma roots at x=0  *)
```

```
{0.472427 + 0.0323801 i, 1.10079 + 0.00415331 i, 22.5351 + 0.667857 i,
 -0.472427 - 0.0323801 i, -1.10079 - 0.00415331 i, -22.5351 - 0.667857 i}
```

- Change model to 2

```
modelList = Table[2, {i, 1, 6}];
```

```
{nPerpHot[xPoint, guesses[[1]]],
 nPerpHot[xPoint, guesses[[2]]],
 nPerpHot[xPoint, guesses[[3]]],
 nPerpHot[xPoint, guesses[[4]]],
 nPerpHot[xPoint, guesses[[5]]],
 nPerpHot[xPoint, guesses[[6]]]}
paramPrint[{dataSet, ne0, nsep, B, freq, nz,
   etaList, TList, modelList, rmaj, rmin, rsep, sol, alphan, alphaT}];
```

```
{0.590246 + 1.7869 × 10^-27 ⅈ, 1.18681 + 3.41307 × 10^-28 ⅈ, 1.18681 + 7.67968 × 10^-26 ⅈ,
 -0.590246 - 1.7869 × 10^-27 ⅈ, -1.18681 - 3.41307 × 10^-28 ⅈ, -1.18681 - 7.67968 × 10^-26 ⅈ}
```

```
dataSet=DIII-D slab

ne0=ne0

nsep=nsep

B=B

freq=55 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

TList={1., 1., 0., 0., 0., 0.}

modelList={2, 2, 2, 2, 2, 2}

rmaj=rmaj

rmin=rmin

rsep=rsep

sol=sol

alphan=alphan

alphaT=alphaT
```

- Hot plasma root finder gets all roots. What about at xmin?

```
xPoint = xmin;
guesses = nPerpWarm6[xPoint] (* Warm plasma roots at x=0  *)
```

```
{0.590188, 1.18597, 19.3865, -0.590188, -1.18597, -19.3865}
```

```
{nPerpHot[xPoint, guesses[[1]]],
 nPerpHot[xPoint, guesses[[2]]],
 nPerpHot[xPoint, guesses[[3]]],
 nPerpHot[xPoint, guesses[[4]]],
 nPerpHot[xPoint, guesses[[5]]],
 nPerpHot[xPoint, guesses[[6]]]}
paramPrint[{dataSet, ne0, nsep, B, freq, nz,
   etaList, TList, modelList, rmaj, rmin, rsep, sol, alphan, alphaT}];
```

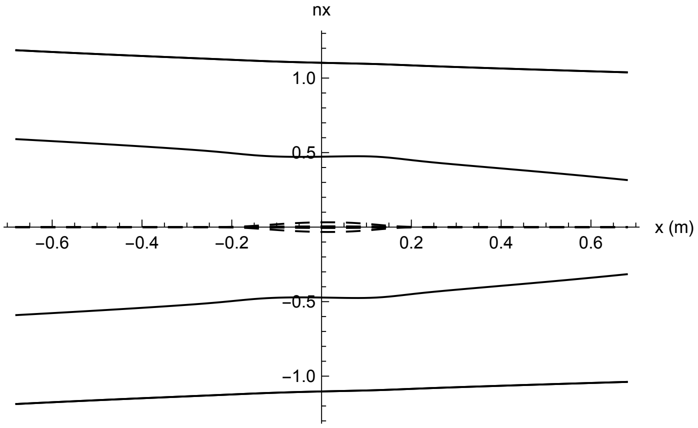
$$\{0.590246 + 1.7869\times10^{-27}\,i,\ 1.18681 + 3.41307\times10^{-28}\,i,\ 1.18681 + 7.67968\times10^{-26}\,i,\ -0.590246 - 1.7869\times10^{-27}\,i,\ -1.18681 - 3.41307\times10^{-28}\,i,\ -1.18681 - 7.67968\times10^{-26}\,i\}$$

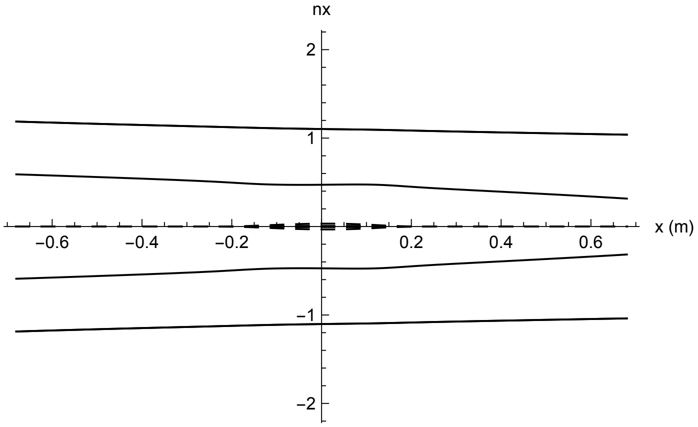
```
dataSet=DIII-D slab

ne0=ne0

nsep=nsep

B=B

freq=55 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

TList={1., 1., 0., 0., 0., 0.}

modelList={2, 2, 0, 0, 0, 0}

rmaj=rmaj

rmin=rmin

rsep=rsep

sol=sol

alphan=alphan

alphaT=alphaT
```

- Hot plasma root finder gets 2 of the roots

Plot hot plasma vs x

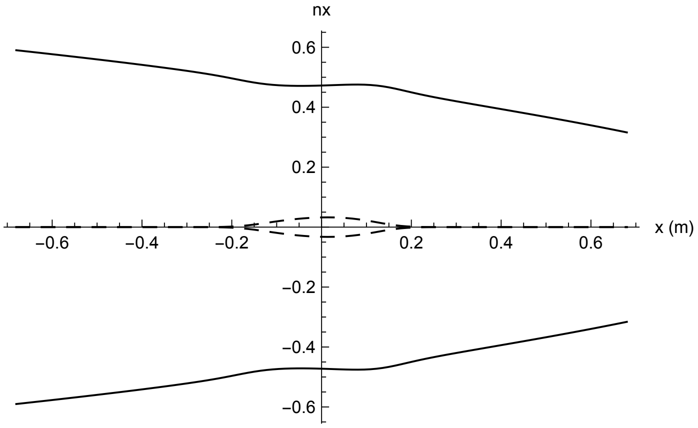
```
g7 = ComplexVectorListPlot[nxhot[1], "x (m)", "nx"];
g8 = ComplexVectorListPlot[nxhot[2], "x (m)", "nx"];
g9 = ComplexVectorListPlot[nxhot[3], "x (m)", "nx"];
g10 = ComplexVectorListPlot[nxhot[4], "x (m)", "nx"];
g11 = ComplexVectorListPlot[nxhot[5], "x (m)", "nx"];
g12 = ComplexVectorListPlot[nxhot[6], "x (m)", "nx"];
Show[{g7, g8, g9, g10, g11, g12}, PlotRange → All, AxesOrigin → {0., 0.}]
Show[{g7, g8, g9, g10, g11, g12}, PlotRange → {-2., 2.}, AxesOrigin → {0., 0.}]
paramPrint[{dataSet, ne0, nsep, B, freq, nz, etaList,
   TList, modelList, rmaj, rmin, rsep, sol, alphan, alphaT}];
```

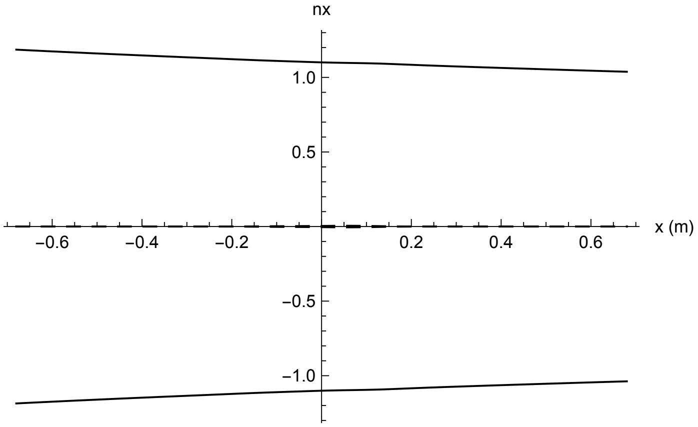


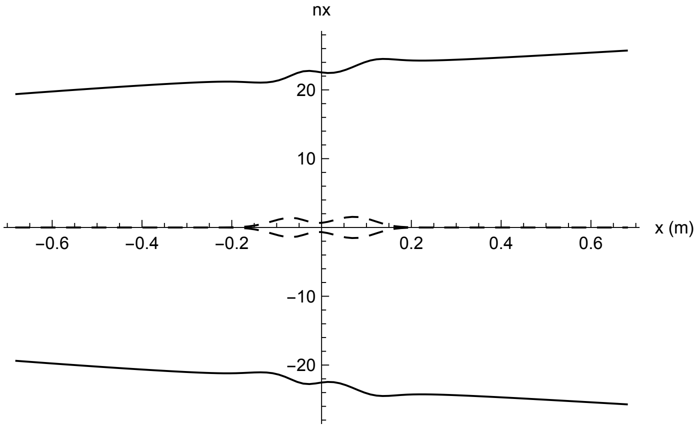


```
dataSet=DIII-D slab

ne0=ne0

nsep=nsep

B=B

freq=55 990

nz=0.1

etaList={1., 0., 0., 0., 0.}

TList={1., 1., 0., 0., 0., 0.}

modelList={2, 2, 0, 0, 0, 0}

rmaj=rmaj

rmin=rmin

rsep=rsep

sol=sol

alphan=alphan

alphaT=alphaT
```

```
Show[{g7, g10}, PlotRange → All, AxesOrigin → {0., 0.}]
Show[{g8, g11}, PlotRange → All, AxesOrigin → {0., 0.}]
Show[{g9, g12}, PlotRange → All, AxesOrigin → {0., 0.}]
```





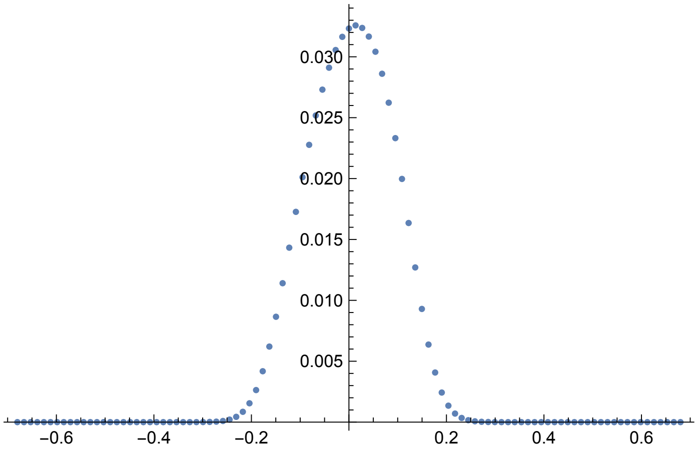


Focus on the damping

```
nxhotIm[iRoot_] := Module[{iRoot0, nxWarm, rootsWarm, nxH, x0, ne, b, t, TL},
   nxWarm = Table[Flatten[{x, nPerpWarm6[x]}], {x, xmin, xmax, (xmax - xmin)/(nPoints - 1)}];
   iRoot0 = iRoot;
   rootsWarm = rootSort[nxWarm];
   nxH = Table[0., {i, 1, nPoints}];
   Do[
    (x0 = xmin + (i - 1) (xmax - xmin)/(nPoints - 1);
     nxGuess = rootsWarm[[i]][[iRoot0 + 1]];
     (*Print["x0 = ", x0,"  nxGuess= ",nxGuess];*)
     nxH[[i]] = {x0, Im[nPerpHot[x0, nxGuess]]};), {i, 1, nPoints}];
   nxH];
```

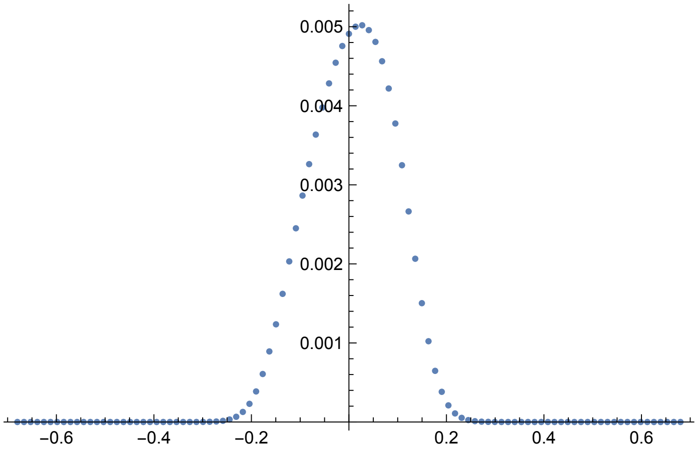
X - mode

```
ListPlot[nxhotIm[1], PlotRange → All]
```



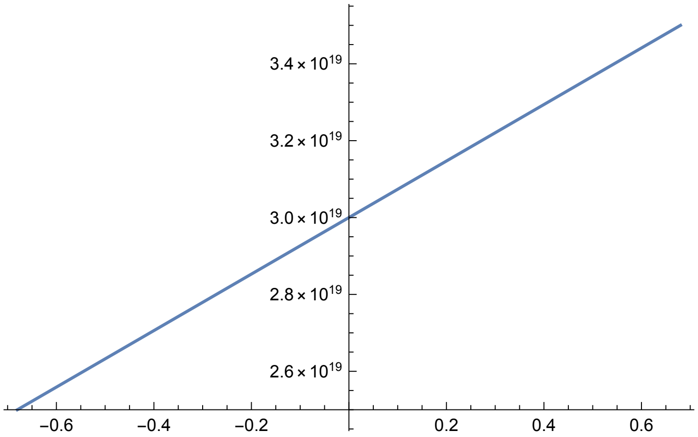
O - mode

```
ListPlot[nxhotIm[2], PlotRange → All]
```

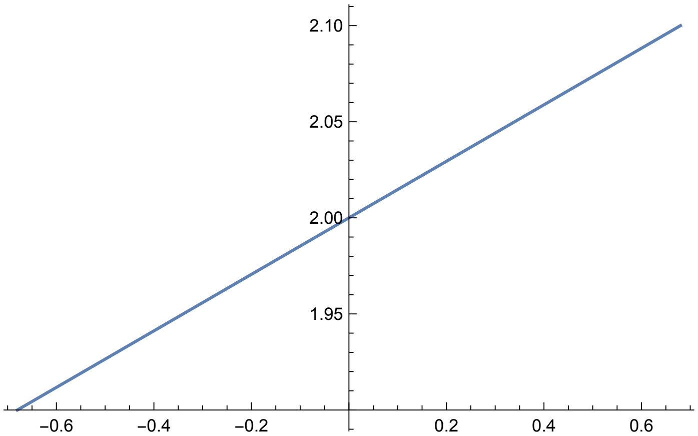


Plot Profiles

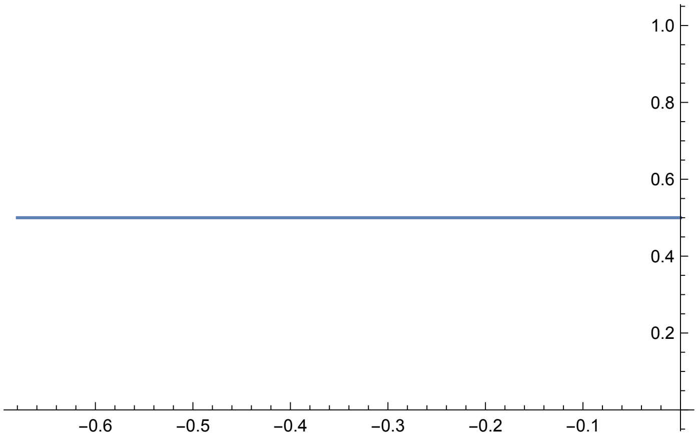
```
Plot[nprof[x], {x, xmin, xmax}]
```



```
Plot[bprof[x], {x, xmin, xmax}]
```



```
Plot[tprof[x], {x, xmin, xmax}]
```



Initialization

Magnetic field,Density and Temperature Profiles

```
bprof[x_] := If[Abs[(BXmax - BXmin) / BXmax] > 10^-6,
    BXmin + (x - xProfileMin) / (xProfileMax - xProfileMin) (BXmax - BXmin), BXmin];
```

```
nprof[x_] := If[Abs[(nXmax - nXmin) / nXmax] > 10^-6,
    nXmin + (x - xProfileMin) / (xProfileMax - xProfileMin) (nXmax - nXmin), nXmin];
```

```
tprof[x_] := If[Abs[(TXmax - TXmin) / TXmax] > 10^-6,
    TXmin + (x - xProfileMin) / (xProfileMax - xProfileMin) (TXmax - TXmin), TXmin];
```