7-21-2017

Engineering Electron Superpositions Using a Magnetic Field

Zoe A. Rowley
Ursinus College, zorowley@ursinus.edu

Bianca R. Gualtieri
Ursinus College, bigualtieri@ursinus.edu

Recommended Citation

Rowley, Zoe A. and Gualtieri, Bianca R., "Engineering Electron Superpositions Using a Magnetic Field" (2017). *Physics and Astronomy Summer Fellows*. 15.
https://digitalcommons.ursinus.edu/physics_astro_sum/15

Engineering Electron Superpositions Using a Magnetic Field

By: Bianca Gualtieri and Zoe Rowley
Mentor: Dr. Thomas Carroll

Abstract

A Rydberg atom has a highly excited valence electron which is weakly bound and far from the nucleus. These atoms have exaggerated properties that make them attractive candidates for quantum computation and studies of fundamental quantum mechanics. The discrete energy levels of Rydberg atoms are shifted in the presence of an electric field by the Stark effect and are similarly shifted due to a magnetic field by the Zeeman effect. These effects couple the energy levels together, creating avoiding crossings. At these avoided crossings, an electron in one energy level can jump to the other.

Our goal is to be able to use these avoided crossings to put the electron in a superposition state of both energy levels. In order to achieve this we created new software that enables us to calculate the energy levels of an electron in both a magnetic and an electric field. We present energy level maps visualizing the results of the Stark and Zeeman effects.

Introduction

The most common method of studying Rydberg atoms is to use an electric field pulse to ionize the outer electron and send it to a detector. This electric field creates a Stark Map which shows all of the possible energy paths that the electron can take to ionization. Due to quantum mechanical properties, energy levels that have different j quantum numbers, but the same m_j states can be coupled together into what we call an avoided crossing. At this avoided crossing, the probability amplitude of the electron can do one of three things: it can stay in its original energy state, it can jump over to the next energy state, or it’s amplitude can split into a superposition of both states, creating an extremely complicated map. When a magnetic field is added to this set-up, it creates extra couplings between energies that have the same j quantum number and m_j states that differ in a magnitude of 1 added significantly more avoided crossings to the Stark map. We explore both parallel and perpendicular magnetic field effects by using a software that we coded in C++. This new software takes in parameters for the minimum and maximum electric field wanted, the minimum and maximum principle quantum numbers (n) , the value of the wanted perpendicular field (B_0) and the wanted parallel field (B_{perp}). It then calculates how large of a matrix needed, calculates the matrix elements, finds the eigenvalues, and produces files that contain the eigenenergy associated with each field. We then read in these .dat files that were created on the super computer and graphed the resulting Stark maps.

Our Project

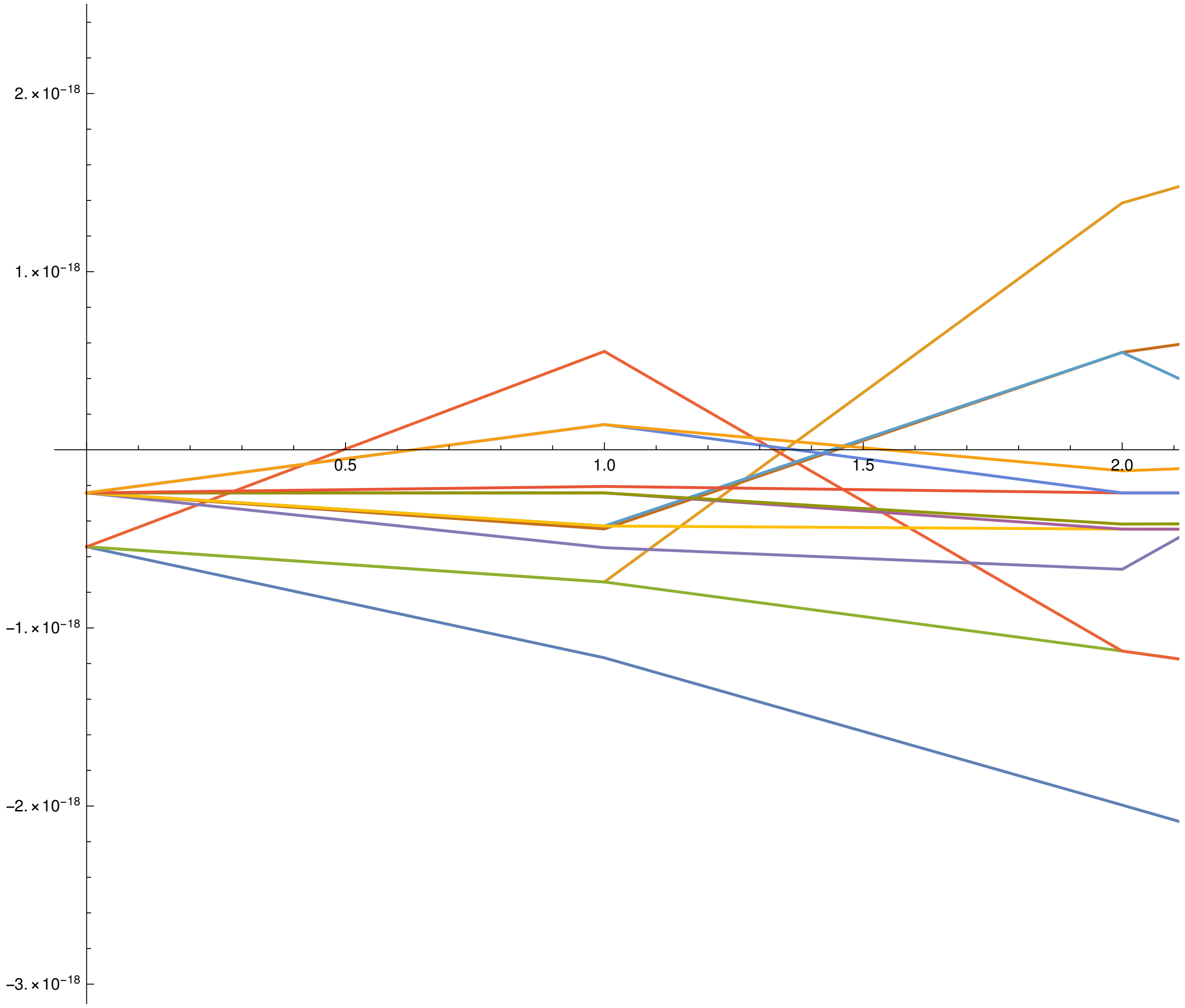
Calculating Stark Map of Hydrogen with just an Electric

Field

```
This .nb has the working code that solves the energy levels of hydrogen and creates the

(*User given n-levels *)

ℰ = 1000;
ℏ = $1.05457 \times 10^{-34}$;
me = $9.1094 \times 10^{-31}$;
mp =  $1.6726 \times 10^{-27}$;
μ = $\frac{\text{me} \times \text{mp}}{\text{me} + \text{mp}}$;
e = $1.602 \times 10^{-19}$;
ϵ0 = $8.854 * 10^{-12}$;
a0 = $\frac{\hbar^2}{\mu\, e^2}$;
A[n_, l_] := $\sqrt{\frac{((n - l - 1)!)}{2 * n * (((n + l)!)^3)}}$;

rwf[n_, l_, ρ_] := $(2 * \kappa[n])^{3/2} \rho^{l}\, e^{-\rho/2} \sum_{i=0}^{n-l-1} \frac{(-1)^{i}\, ((n + l)!)^2 \rho^{i}}{i!\, (n - l - 1 - i)!\, (2\, l + 1 + i)!}$;

z = 1;
κ[n_] := $\frac{z}{\text{a0}\, n}$;



Nmax = 3;
Nmin = 2;
(*Finds the size of the matrix needed given the above n-levels*)
Size = $\sum_{i=\text{Nmin}}^{\text{Nmax}} i^2$;
starkmap = {};
ℰmax = 3;
ℰstep = 1;
For[ℰ = 0, ℰ <= ℰmax, ℰ += ℰstep,
  Print[ℰ];
  (*Creates the matrix of said size*)
  H = ConstantArray[0, {Size, Size}];
  (*Initial postion in the matrix is the top left corner,
  Mathematica starts at 1 not 0*)
  i = 1;
  j = 1;
  (*Specifies the allowed range of n*)
For[n = Nmin, n ≤ Nmax, n++,
   (*Specifies the allowed range of l*)
   For[l = 0, l < n, l++,
     (*Specifies the allowed range of ml*)
     For[ml = -l, ml ≤ l, ml++,
       (*Specifies the allowed range of n prime*)
       For[np = Nmin, np ≤ Nmax, np++,
        (*Specifies the allowed range of l prime*)
        For[lp = 0, lp < np, lp++,
          (*Specifies the allowed range of ml prime *)
          For[mlp = -lp, mlp ≤ lp, mlp++,
            (*Instructs the matrix element to be zero unless the ml
             and ml prime are equal, and l and l prime differ by 1 and
             only 1. In which case we fill that element with 1. The next
             step is to get the 1 spots to perform the integral needed*)
            If[ml == mlp && Abs[l - lp] == 1, H[[i, j]] = ℰ * e * A[n, l] * A[np,
                 lp] ∫_0^∞ ∫_0^π ∫_0^(2 π) ρ^3/(2 κ[n])^4 rwf[n, l, ρ] * Conjugate[rwf[np, lp, ρ]] *
                    SphericalHarmonicY[l, ml, θ, ϕ] * Conjugate[
                     SphericalHarmonicY[lp, mlp, θ, ϕ]] Sin[θ] Cos[θ] ⅆϕ ⅆθ ⅆρ];
            If[ml == mlp && n == np && l == lp,
             H[[i, j]] = -μ (z e^2)^2/(32 π^2 ℏ^2 ϵ0^2 n^2);
            ]; (*close diagonal*)
            (*Increment j each time*)
            j++;
           ];
         ];
       ];
       (*Increment i after the first i has had all its j's done,
       then start at j=1 again*)
       i++;
       j = 1;
      ];
    ];
  ];
  ev = Eigenvalues[H];
  AppendTo[starkmap, Table[{ℰ, ev[[i]]}, {i, 1, Length[ev]}]];
 ];
```

```
0

1

2

3
```

```
ListLinePlot[Transpose[starkmap], PlotRange → All]
```



This above map is incorrect because the eigenenergies are unordered. This problem only occurs with Hydrogen, however, since it has no avoided crossings whatsoever.

```
MatrixForm[H]
```

$$\left(\begin{array}{cccccc}
-5.44457\times10^{-19} & 0 & -6.86242\times10^{-19} & 0 & 0 & \\
0 & -5.44457\times10^{-19} & 0 & 0 & 0 & \\
-6.86242\times10^{-19} & 0 & -5.44457\times10^{-19} & 0 & 2.49029\times10^{-19} & \\
0 & 0 & 0 & -5.44457\times10^{-19} & 0 & \\
0 & 0 & 1.26071\times10^{-18} & 0 & -2.41981\times10^{-19} & \\
0 & 0 & 0 & 0 & 0 & -2.4198 \\
2.05873\times10^{-18} & 0 & 0 & 0 & -1.68094\times10^{-18} & \\
0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & \\
0 & 1.54405\times10^{-18} & 0 & 0 & 0 & -1.0293 \\
0 & 0 & 1.78291\times10^{-18} & 0 & 0 & \\
0 & 0 & 0 & 1.54405\times10^{-18} & 0 & \\
0 & 0 & 0 & 0 & 0 &
\end{array}\right.$$

```
Transpose[starkmap];
```

$$-\mu\frac{(z\,e^2)^2}{32\,\pi^2\,\hbar^2\,\epsilon0^2\,n^2}$$

$$-6.04952\times10^{-20}$$

$$-\mu\frac{(z\,e^2)^2}{32\,\pi^2\,\hbar^2\,\epsilon0^2\,1^2}\Big/(1.6*10^{-19})$$

$$-13.6114$$

$$R=\frac{\hbar^2}{2\,\mu\,a0^2}$$

$$2.69602\times10^{-38}$$

$$-R/n^2$$

$$-7.48893\times10^{-40}$$

$$\mu$$

$$9.10444\times10^{-31}$$

Counting States for the Stark Map

With the inclusion of a magnetic field, the Hamiltonian matrix that is used to calculate the eigenenergies significantly increases to account for all the new coupled energies. Below shows what else needs to be included in the matrix.

Making the matrix including a Magnetic field

Counting over *l*, *j*, *mj* for a specific n

For a given n value, there are 18 additional states associated with it. These additional states depend on the quantum numbers *l*, *j*, and *mj* associated with n.

○ The quantum number *l* ranges from 0 to n-1
○ For each *l*, j is calculated by $l + \frac{1}{2}$ in steps of 1
○ Each *mj* is calculated by ±j in steps of 1

For further clarification, see the table below for an example of counting the *l*, *j*, and *mj* states for an arbitrary n value.

n	ℓ	j	mj
n	0	$\frac{1}{2}$	$+\frac{1}{2}$
	0	$\frac{1}{2}$	$-\frac{1}{2}$
	1	$\frac{1}{2}$	$+\frac{1}{2}$
	1	$\frac{1}{2}$	$-\frac{1}{2}$
	1	$\frac{3}{2}$	$+\frac{3}{2}$
	1	$\frac{3}{2}$	$+\frac{1}{2}$
	1	$\frac{3}{2}$	$-\frac{1}{2}$
	1	$\frac{3}{2}$	$-\frac{3}{2}$
	2	$\frac{3}{2}$	$+\frac{3}{2}$
	2	$\frac{3}{2}$	$+\frac{1}{2}$
	2	$\frac{3}{2}$	$\frac{-1}{2}$
	2	$\frac{3}{2}$	$-\frac{3}{2}$
	2	$\frac{5}{2}$	$+\frac{5}{2}$
	2	$\frac{5}{2}$	$+\frac{3}{2}$
	2	$\frac{5}{2}$	$+\frac{1}{2}$
	2	$\frac{5}{2}$	$-\frac{1}{2}$
	2	$\frac{5}{2}$	$-\frac{3}{2}$
	2	$\frac{5}{2}$	$-\frac{5}{2}$
	3	$\frac{5}{2}$	$+\frac{5}{2}$

Matrix Size

To get the size of the matrix, we need to sum over all the added states. We have discovered a pattern for each increment of *l* which resulted in the following summation:

$$\sum_{l=0}^{n-1}(4*l)+2$$

To account for every state from nmin to nmax, we would then need a double summation to correctly get the size of the matrix. It would look like:

$$\left(\sum_{nmin}^{nmax}\sum_{l=0}^{n-1}(4*l+2)\right) + 18*(\text{added states})$$

This file finds the size of the Stark Matrix that accounts for magnetic fields

Now that we know how many more energies we need, this file will generate the size of the matrix needed based on the minimum and maximum principle quantum numbers that we want to look at.

```
n = 2;
```

$$\sum_{l=0}^{n-1} 4*l+2$$

```
6
```

```
nmin = 34;
nmax = 35;
addedStates = 3;
```

$$\left(\sum_{n=nmin}^{nmax}\left(\sum_{l=0}^{n-1}(4*l+2)\right)\right)+(18*addedStates)$$

```
4816
```

```
(*Check the double summation
  Sum[Sum[4*l+2,{l,0,n-1}],{n,nmin,nmax}]+18*3
*)
```

```
q = 0;
For[n = nmin, n <= nmax, n++,
 For[l = 0, l <= n - 1, l++,
  q += (4 * l + 2)
 ]
]
q += 18 * addedStates
```

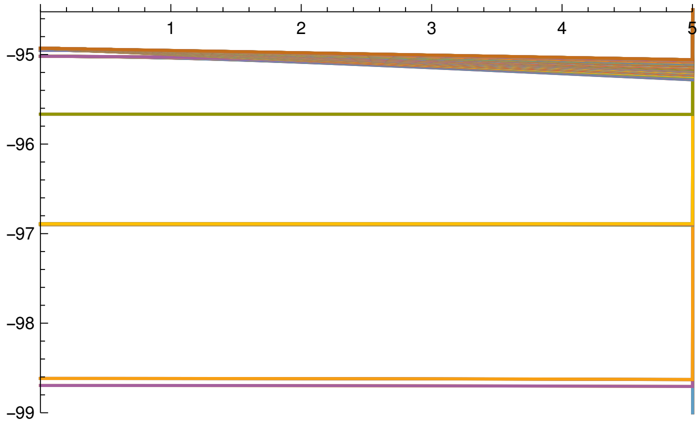
This file looks at the Stark Maps affected by combinations of parallel and perpendicular magnetic fields with an electric field

Here is where we ran all of our calculations and analyzed the different effects of parallel and perpendicular magnetic fields combined with a constant electric field.

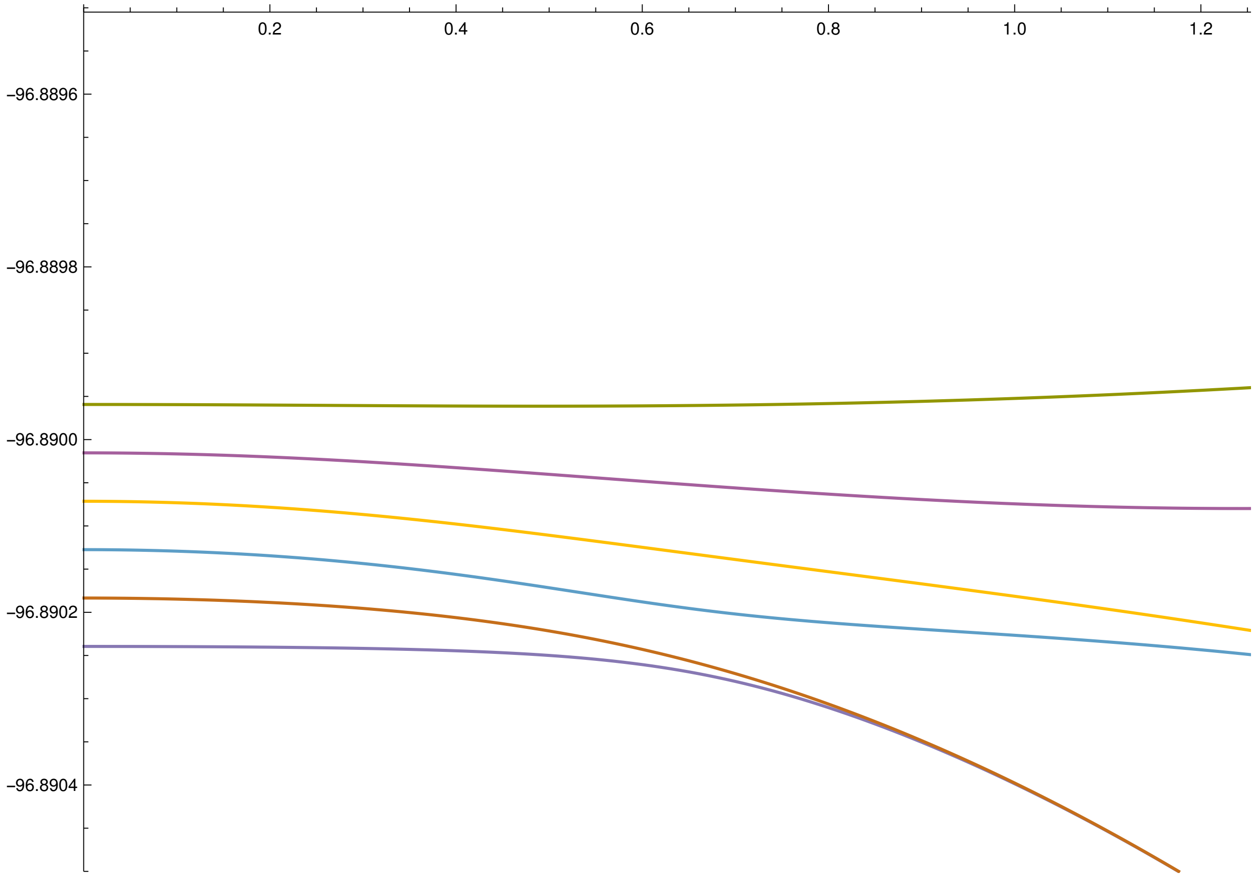
This study looks at the map with 0.0 Gauss Parallel Magnetic Field and a 1.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
noParPerp1 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/NoPar_Perp1_Res.01/NoPar_Perp1_Res
     .01_n_33_34_f_0_5_res_0.01_Bo_1_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp1[[iterate++]]];
numFields = noParPerp1[[iterate++]];
nMin = noParPerp1[[iterate++]];
nMax = noParPerp1[[iterate++]];
fMin = noParPerp1[[iterate++]];
fMax = noParPerp1[[iterate++]] ;
res = noParPerp1[[iterate++]] ;
Bo = noParPerp1[[iterate++]];
Bp = noParPerp1[[iterate++]];
numEnergies = noParPerp1[[iterate++]];
(*Print out the header information
 Print[
  "Header size: ",headerLength,  "\n"
   "Number of fields: ",numFields, "\n"
   "nMin: ",nMin, "\n"
   "nMax: ",nMax,  "\n"
   "Min field: ", fMin,"\n"
   "Max field: ", fMax, "\n"
   "Perpendicular mag field: ", Bo, "\n"
   "Parrallell mag field: ", Bp, "\n"
   "Field resolution: ",res, "\n"
   "Number of Energies: ", numEnergies, "\n"
 ]*)

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp1[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



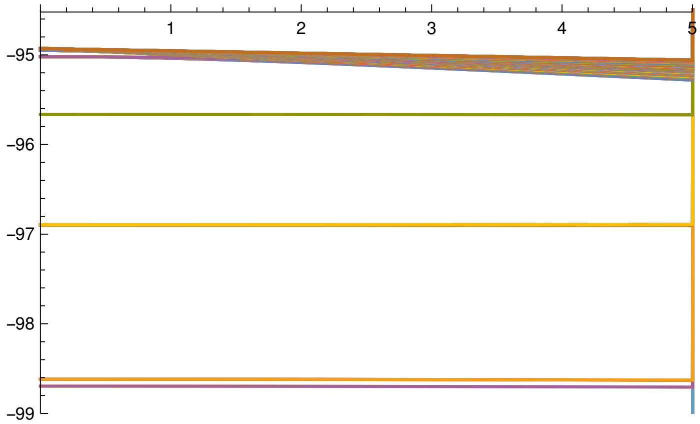
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8895, -96.8905}}]
```



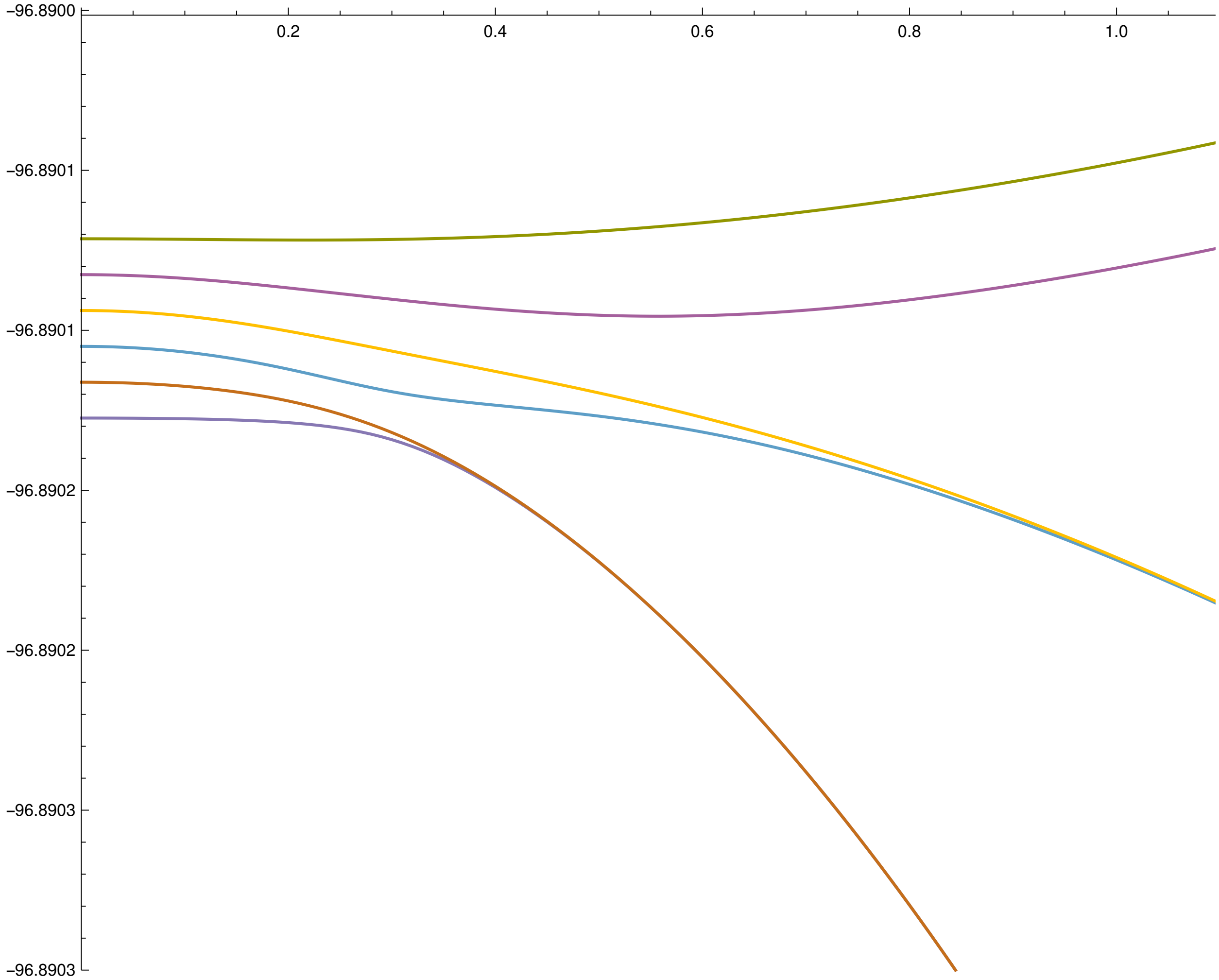
This study looks at the map with 0.0 Gauss Parallel Magnetic Field and a 0.2 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging

0 to 5 V/cm

```
noParPerp02 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/noPar_Perp.2/noPar_Perp.2_n_33_34_f
     _0_5_res_0.01_Bo_0.2_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp02[[iterate++]]];
numFields = noParPerp02[[iterate++]];
nMin = noParPerp02[[iterate++]];
nMax = noParPerp02[[iterate++]];
fMin = noParPerp02[[iterate++]];
fMax = noParPerp02[[iterate++]] ;
res = noParPerp02[[iterate++]] ;
Bo = noParPerp02[[iterate++]];
Bp = noParPerp02[[iterate++]];
numEnergies = noParPerp02[[iterate++]];
(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp02[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];

fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



```
testOneNoPar =
 ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.890, -96.8903}}]
```

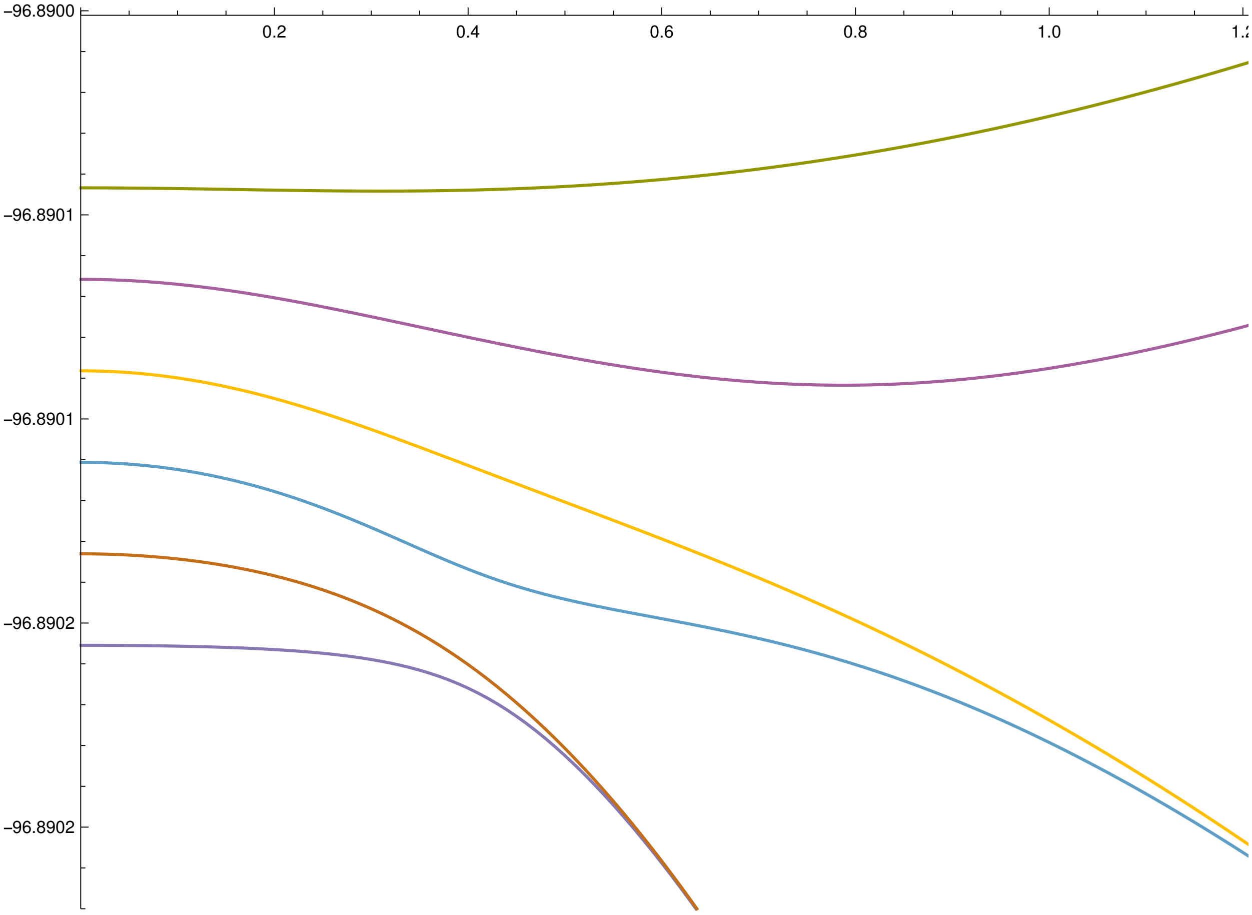


```
Export["noPar_Perp.2.pdf", testOneNoPar];
```

This study looks at the map with 0.0 Gauss Parallel Magnetic Field and a 0.4 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

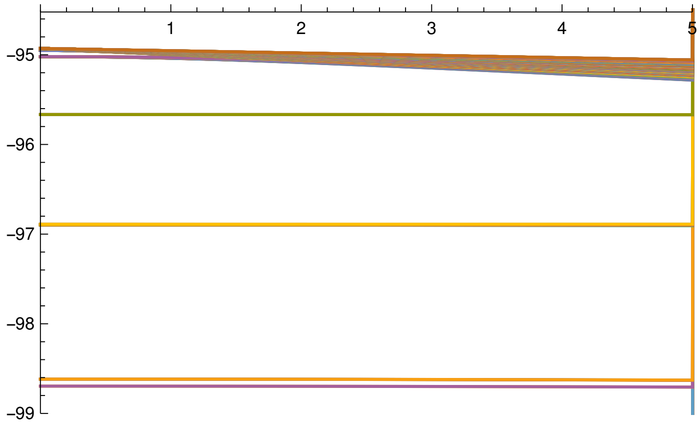
```
noParPerp04 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/noPar_Perp.4/noPar_Perp.4_n_33_34_f
     _0_5_res_0.01_Bo_0.4_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp04[[iterate++]]];
numFields = noParPerp04[[iterate++]];
nMin = noParPerp04[[iterate++]];
nMax = noParPerp04[[iterate++]];
fMin = noParPerp04[[iterate++]];
fMax = noParPerp04[[iterate++]] ;
res = noParPerp04[[iterate++]] ;
Bo = noParPerp04[[iterate++]];
Bp = noParPerp04[[iterate++]];
numEnergies = noParPerp04[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp04[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.89, -96.89022}}]
```

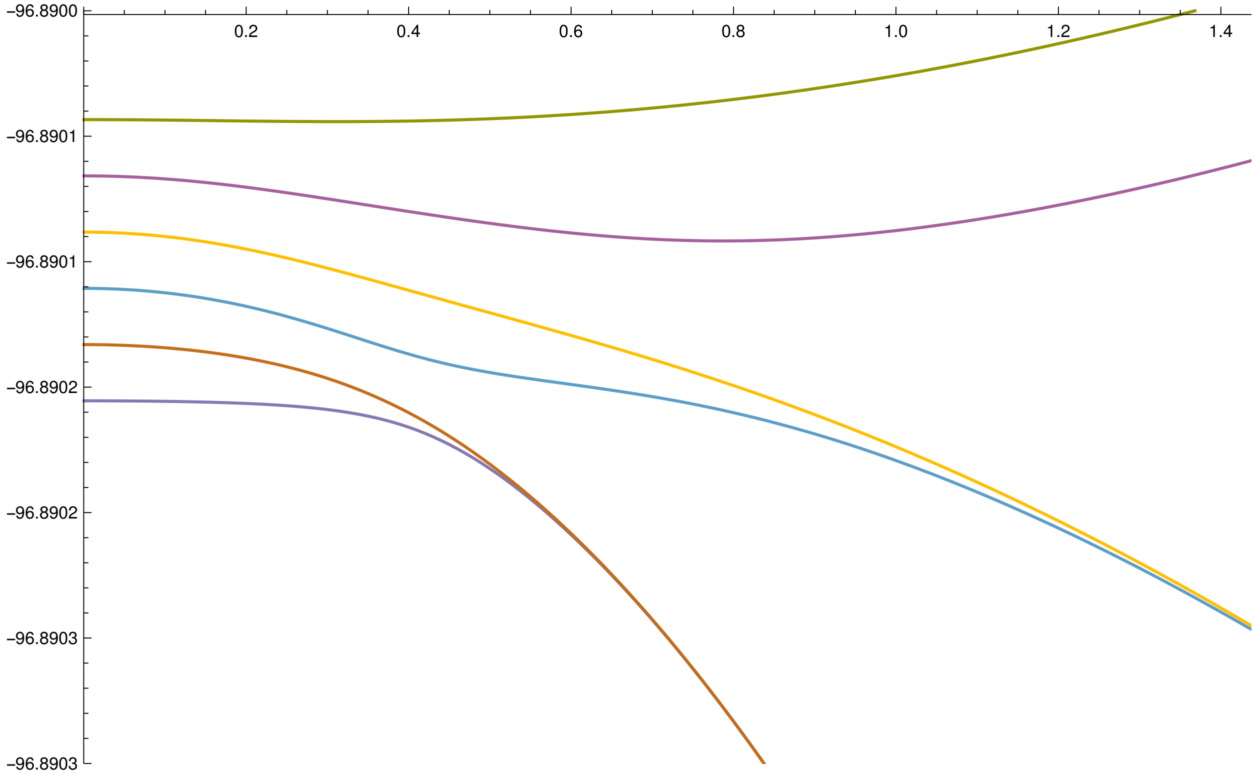


no perp par .6

```
noParPerp06 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/noPar_Perp.6/noPar_Perp.6_n_33_34_f
     _0_5_res_0.01_Bo_0.6_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp06[[iterate++]]];
numFields = noParPerp06[[iterate++]];
nMin = noParPerp06[[iterate++]];
nMax = noParPerp06[[iterate++]];
fMin = noParPerp06[[iterate++]];
fMax = noParPerp06[[iterate++]] ;
res = noParPerp06[[iterate++]] ;
Bo = noParPerp06[[iterate++]];
Bp = noParPerp06[[iterate++]];
numEnergies = noParPerp06[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp06[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];

fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



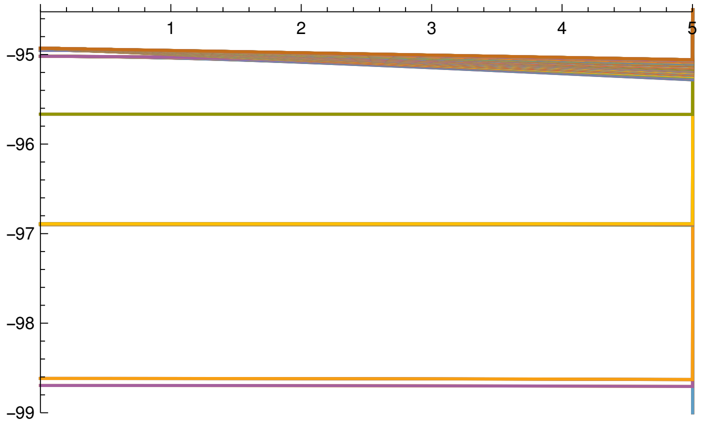
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.89, -96.8903}}]
```



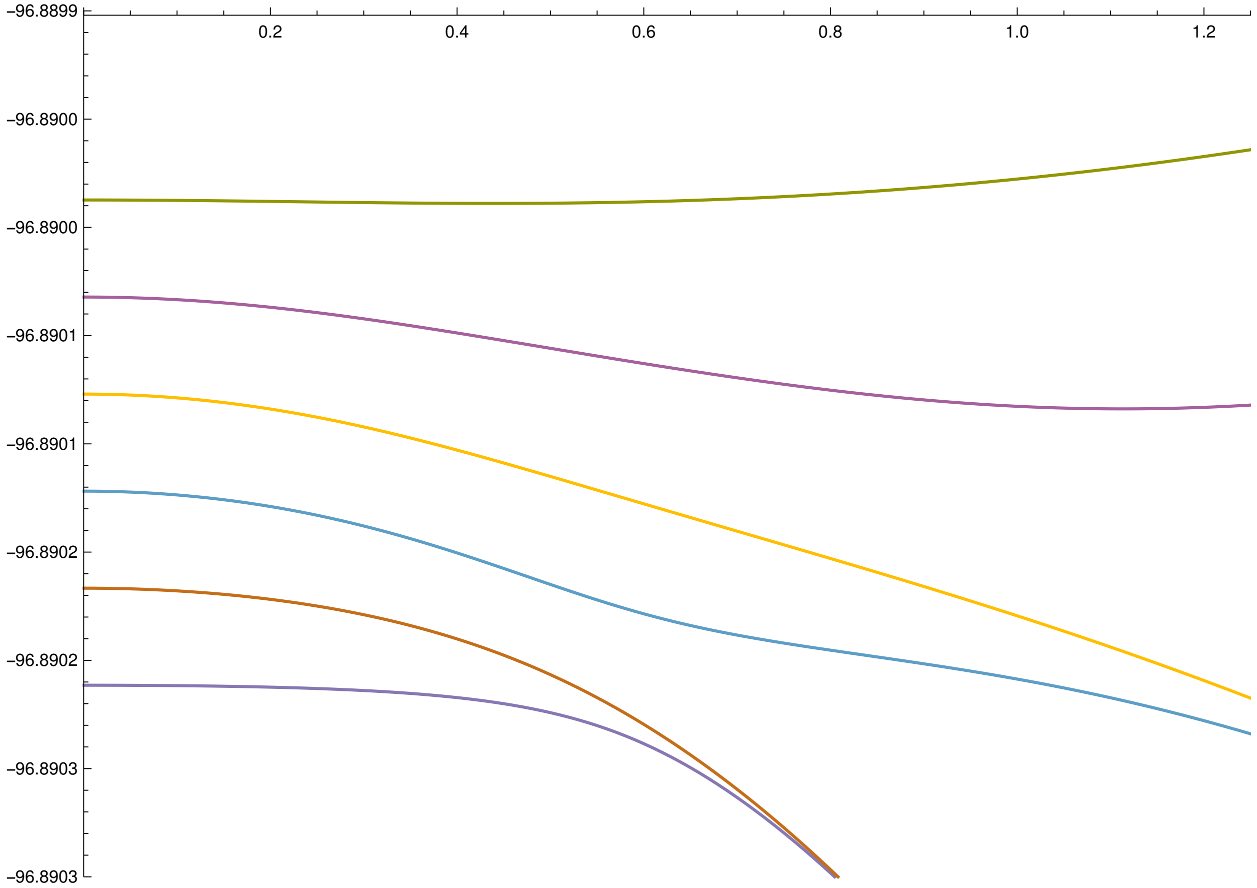
This study looks at the map with 0.0 Gauss Parallel Magnetic Field and a 0.8 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
noParPerp08 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/noPar_Perp.8/noPar_Perp.8_n_33_34_f
     _0_5_res_0.01_Bo_0.8_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp08[[iterate++]]];
numFields = noParPerp08[[iterate++]];
nMin = noParPerp08[[iterate++]];
nMax = noParPerp08[[iterate++]];
fMin = noParPerp08[[iterate++]];
fMax = noParPerp08[[iterate++]] ;
res = noParPerp08[[iterate++]] ;
Bo = noParPerp08[[iterate++]];
Bp = noParPerp08[[iterate++]];
numEnergies = noParPerp08[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp08[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



```
testTwoNoPar =
 ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8899, -96.8903}}]
```



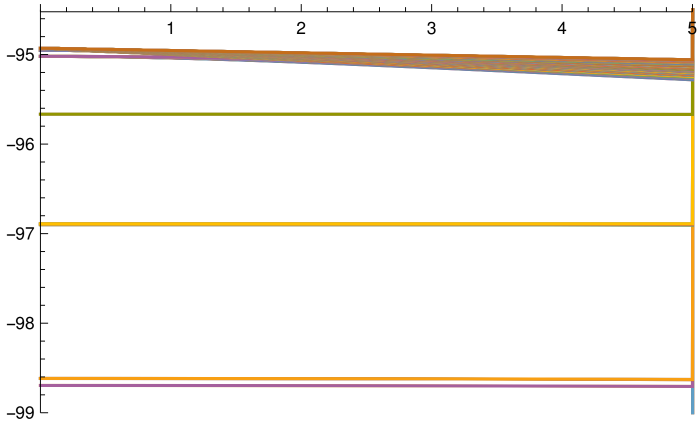
```
Export["noPar_Perp.8.pdf", testTwoNoPar];
```

This study looks at the map with 1.0 Gauss Parallel

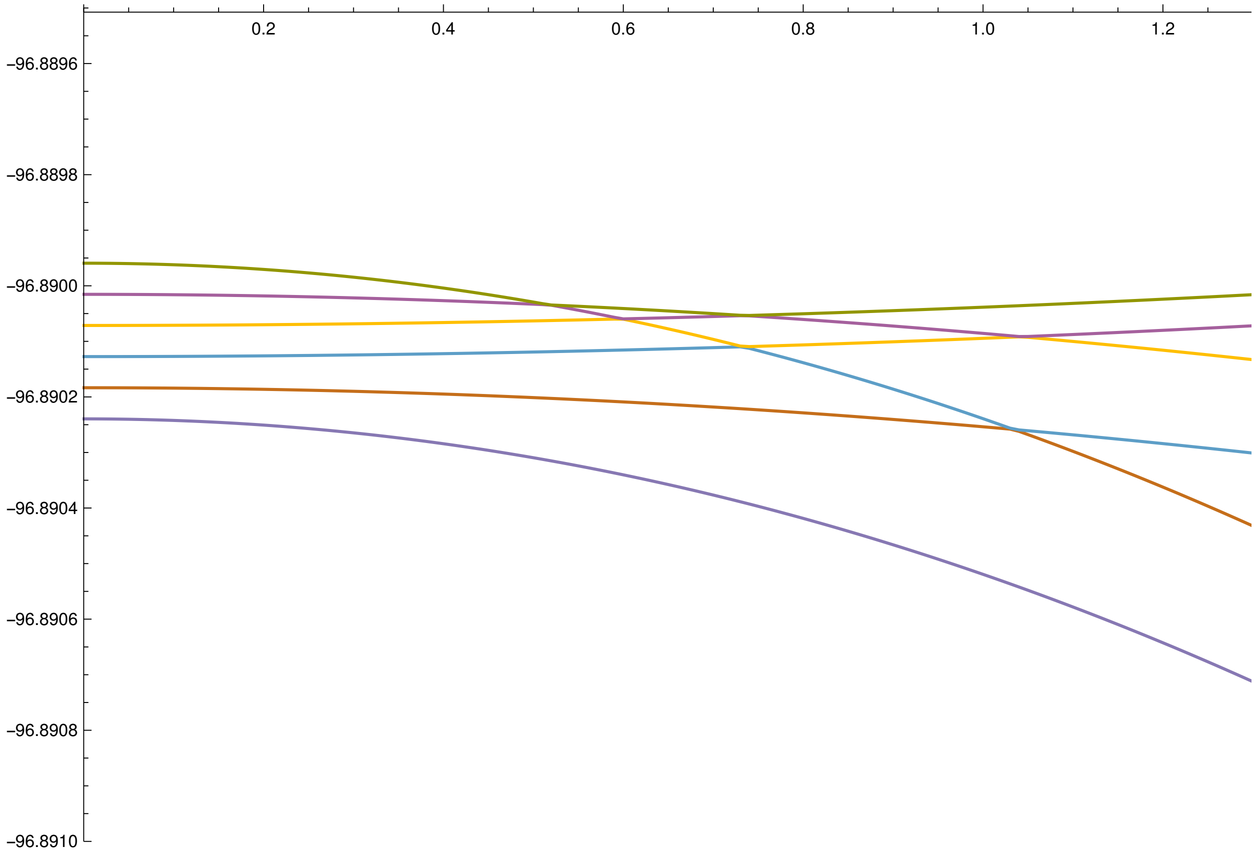
Magnetic Field and a 0.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
Par1NoPerp = BinaryReadList[
   "/home/zorowley/Documents/summerFellows2017/Par1_NoPerp_n_33_34_f_0_5_res_0.01_Bo_
     0_Bp_1.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[Par1NoPerp[[iterate++]]];
numFields = Par1NoPerp[[iterate++]];
nMin = Par1NoPerp[[iterate++]];
nMax = Par1NoPerp[[iterate++]];
fMin = Par1NoPerp[[iterate++]];
fMax = Par1NoPerp[[iterate++]] ;
res = Par1NoPerp[[iterate++]] ;
Bo = Par1NoPerp[[iterate++]];
Bp = Par1NoPerp[[iterate++]];
numEnergies = Par1NoPerp[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[Par1NoPerp[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



```
more = ListLinePlot[starkMap[[2213 ;; 2223]],
  PlotRange → {{0, 1.5}, {-96.8895, -96.891}}]
```



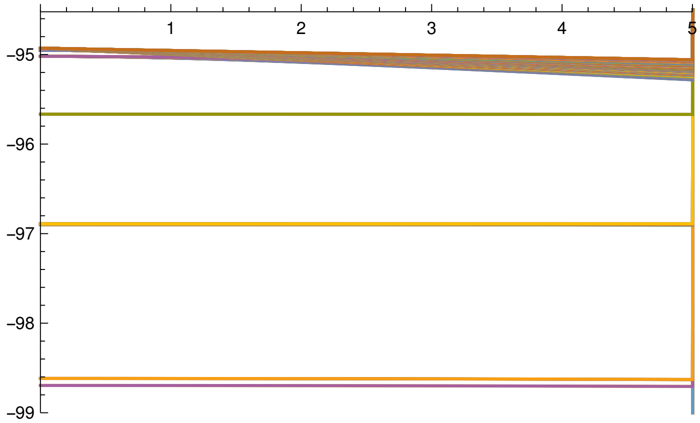
```
Export[
```

This study looks at the map with 0.2 Gauss Parallel

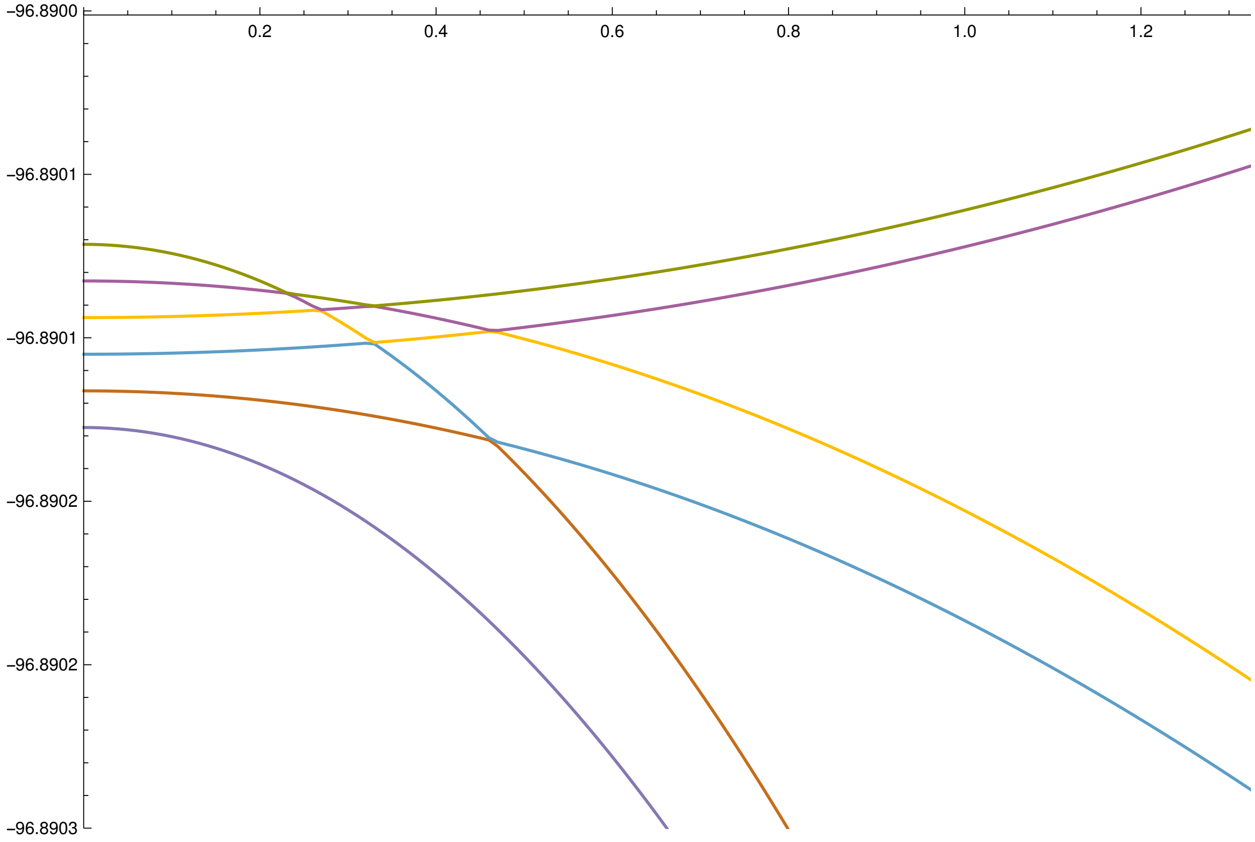
Magnetic Field and a 0.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
Par02NoPerp = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/Par.2_NoPerp/Par.2_NoPerp_n_33_34_f
     _0_5_res_0.01_Bo_0_Bp_0.2.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[Par02NoPerp[[iterate++]]];
numFields = Par02NoPerp[[iterate++]];
nMin = Par02NoPerp[[iterate++]];
nMax = Par02NoPerp[[iterate++]];
fMin = Par02NoPerp[[iterate++]];
fMax = Par02NoPerp[[iterate++]] ;
res = Par02NoPerp[[iterate++]] ;
Bo = Par02NoPerp[[iterate++]];
Bp = Par02NoPerp[[iterate++]];
numEnergies = Par02NoPerp[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[Par02NoPerp[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



```
testpar2NoPerp =
 ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8900, -96.89025}}]
```

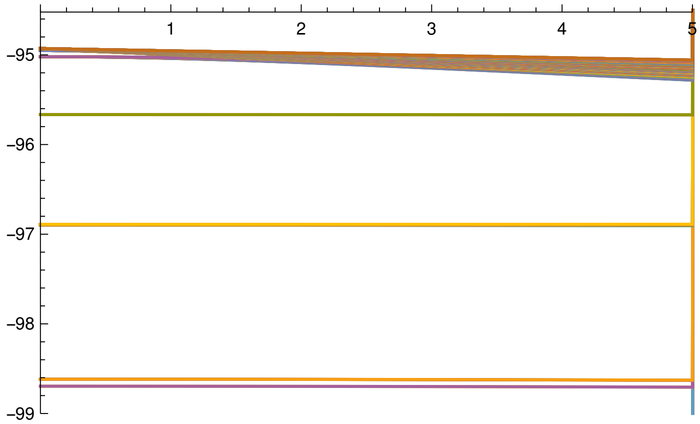


```
Export["par.2_noPerp.pdf", testpar2NoPerp];
```

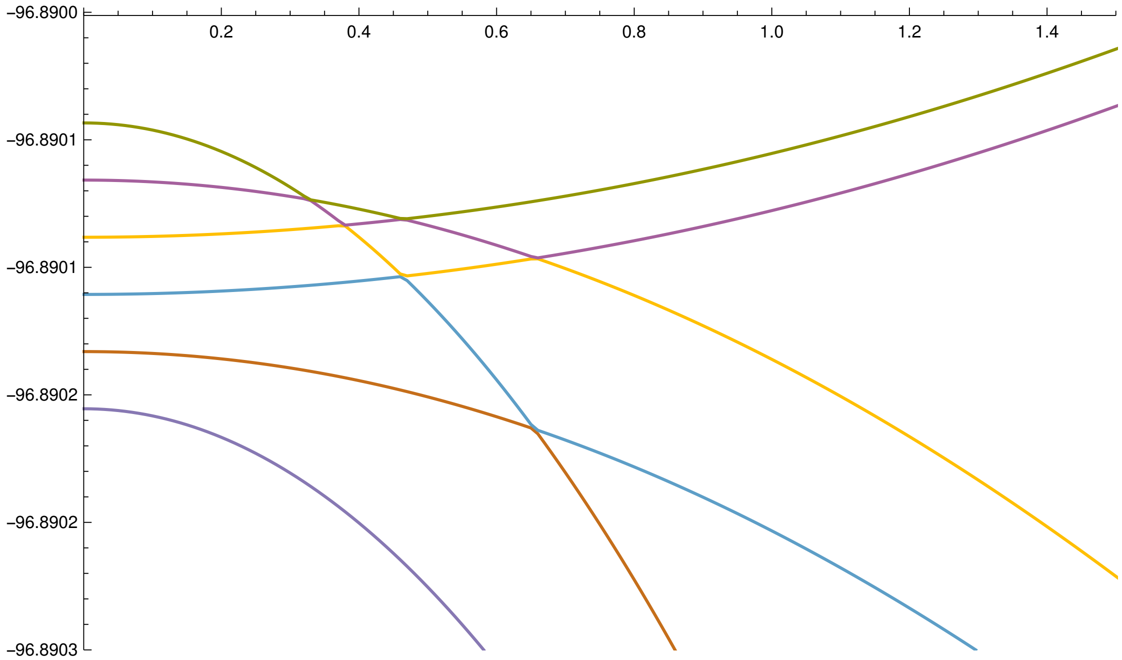
This study looks at the map with 0.4 Gauss Parallel Magnetic Field and a 0.0 Gauss Perpendicular Magnetic

Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
Par04NoPerp = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/Par.4_NoPerp/Par.4_NoPerp_n_33_34_f
     _0_5_res_0.01_Bo_0_Bp_0.4.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[Par04NoPerp[[iterate++]]];
numFields = Par04NoPerp[[iterate++]];
nMin = Par04NoPerp[[iterate++]];
nMax = Par04NoPerp[[iterate++]];
fMin = Par04NoPerp[[iterate++]];
fMax = Par04NoPerp[[iterate++]] ;
res = Par04NoPerp[[iterate++]] ;
Bo = Par04NoPerp[[iterate++]];
Bp = Par04NoPerp[[iterate++]];
numEnergies = Par04NoPerp[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[Par04NoPerp[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];

fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



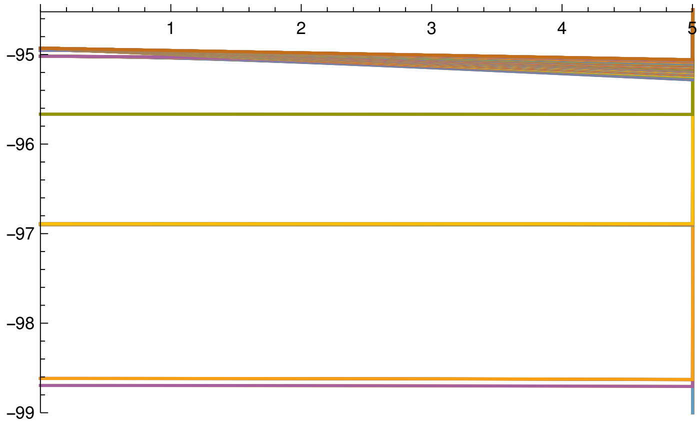
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8900, -96.89025}}]
```



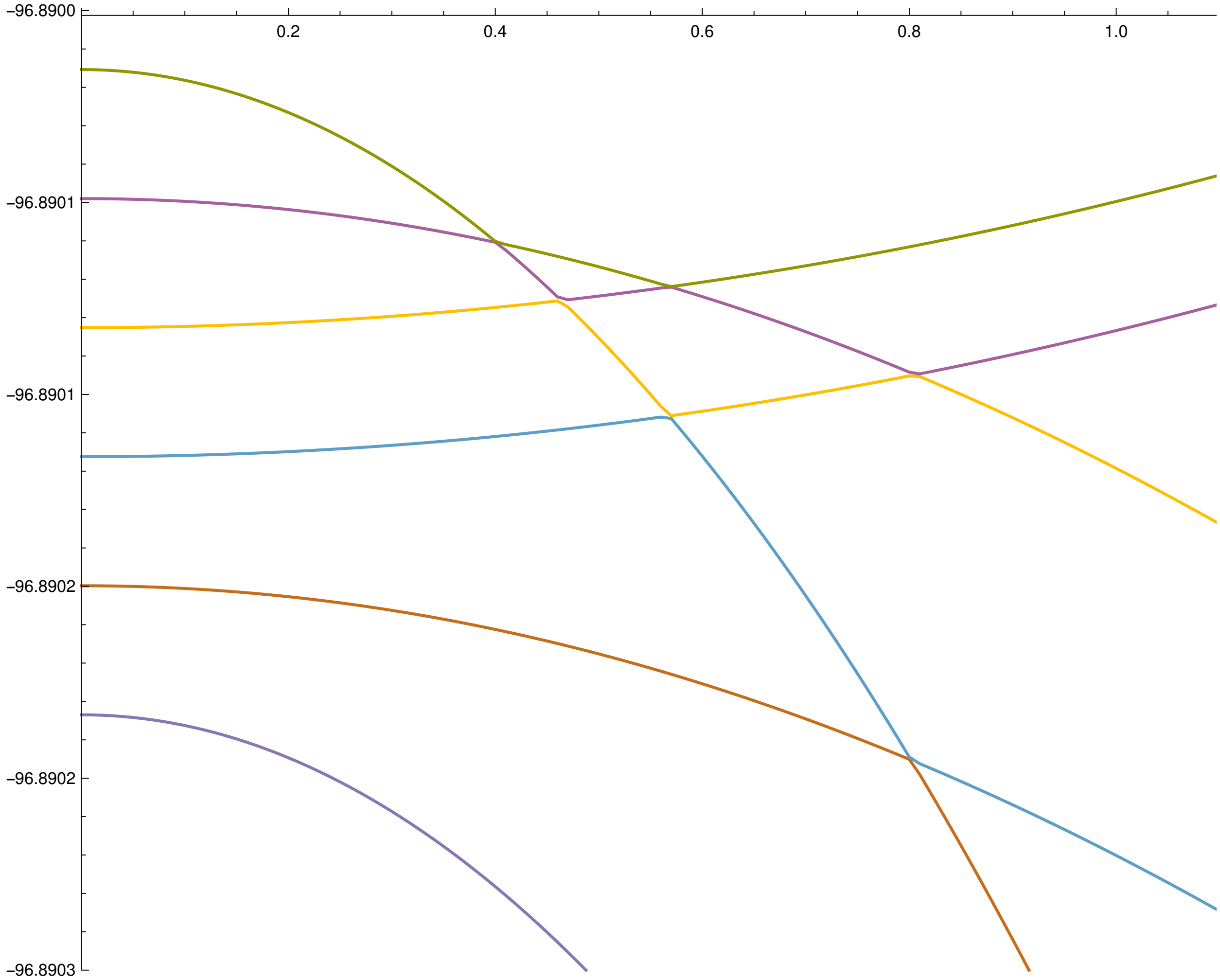
This study looks at the map with 0.6 Gauss Parallel Magnetic Field and a 0.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
Par06NoPerp = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/Par.6_NoPerp/Par.6_NoPerp_n_33_34_f
     _0_5_res_0.01_Bo_0_Bp_0.6.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[Par06NoPerp[[iterate++]]];
numFields = Par06NoPerp[[iterate++]];
nMin = Par06NoPerp[[iterate++]];
nMax = Par06NoPerp[[iterate++]];
fMin = Par06NoPerp[[iterate++]];
fMax = Par06NoPerp[[iterate++]] ;
res = Par06NoPerp[[iterate++]] ;
Bo = Par06NoPerp[[iterate++]];
Bp = Par06NoPerp[[iterate++]];
numEnergies = Par06NoPerp[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[Par06NoPerp[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
     {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



```
testperp6nopar =
 ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8900, -96.89025}}]
```



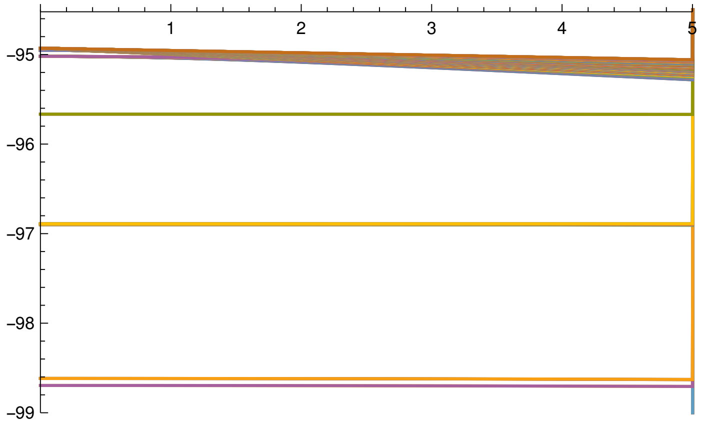
```
Export["perp.6_noPar.pdf", testperp6nopar];
```

This study looks at the map with 0.8 Gauss Parallel

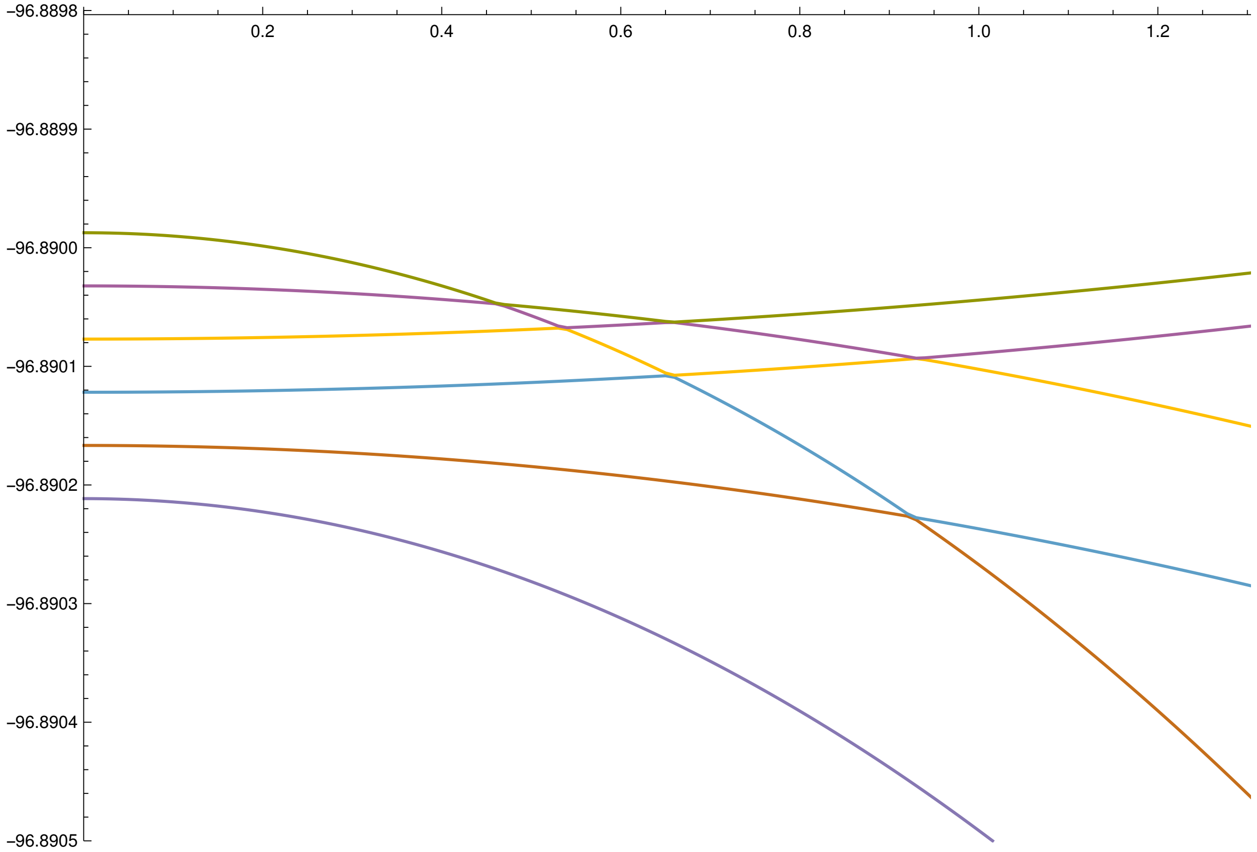
Magnetic Field and a 0.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
Par08NoPerp = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/Par.8_NoPerp/Par.8_NoPerp_n_33_34_f
     _0_5_res_0.01_Bo_0_Bp_0.8.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[Par08NoPerp[[iterate++]]];
numFields = Par08NoPerp[[iterate++]];
nMin = Par08NoPerp[[iterate++]];
nMax = Par08NoPerp[[iterate++]];
fMin = Par08NoPerp[[iterate++]];
fMax = Par08NoPerp[[iterate++]] ;
res = Par08NoPerp[[iterate++]] ;
Bo = Par08NoPerp[[iterate++]];
Bp = Par08NoPerp[[iterate++]];
numEnergies = Par08NoPerp[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[Par08NoPerp[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



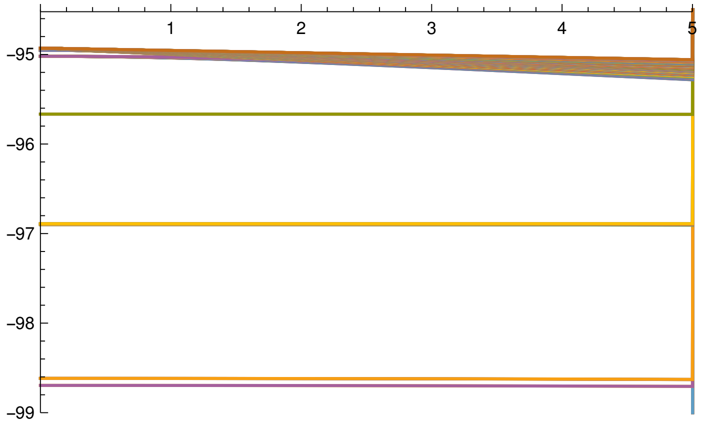
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8898, -96.8905}}]
```



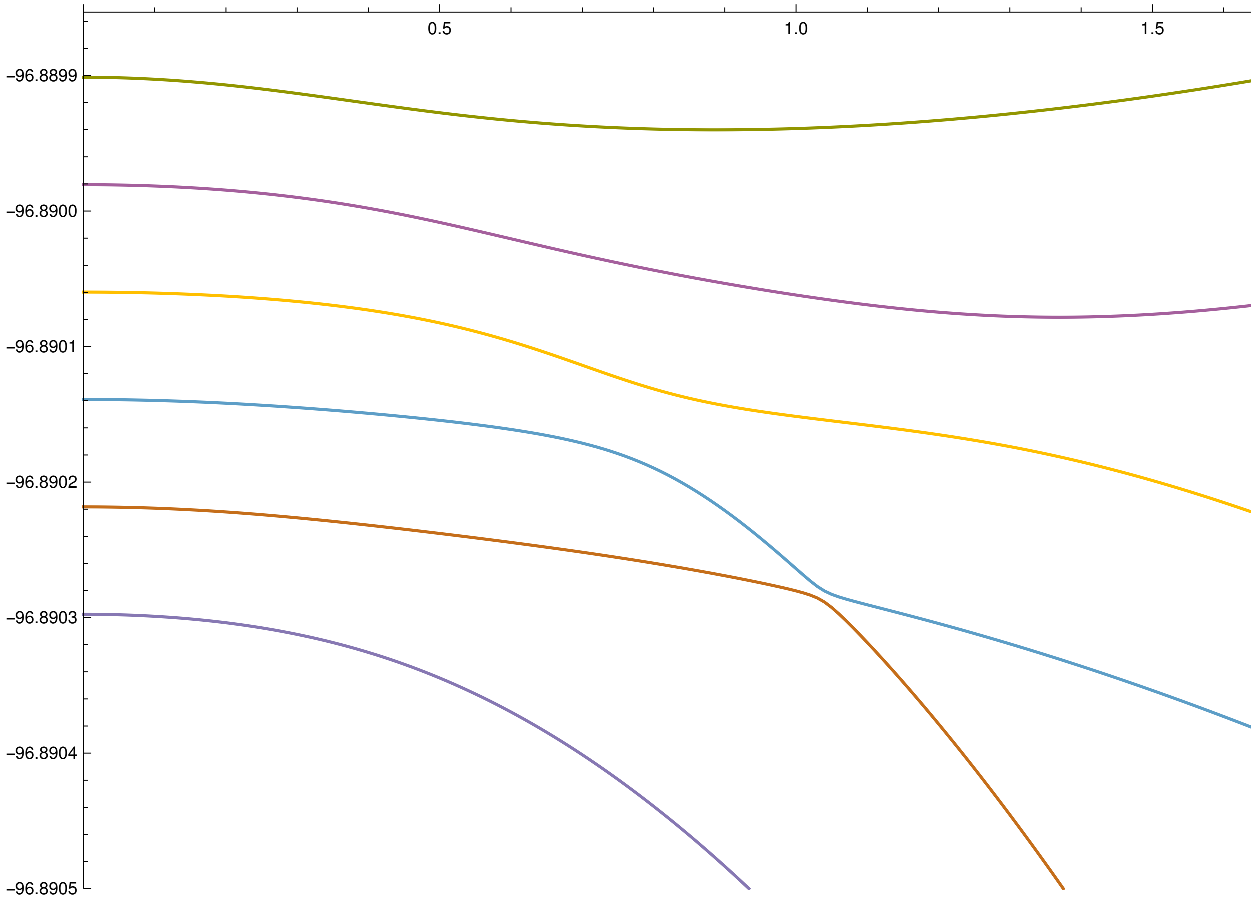
This study looks at the map with 1.0 Gauss Parallel Magnetic Field and a 1.0 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging

0 to 5 V/cm

```
PaPerp1 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/ParPerp1/ParPerp1_n_33_34_f_0_5_res
     _0.01_Bo_1_Bp_1.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[PaPerp1[[iterate++]]];
numFields = PaPerp1[[iterate++]];
nMin = PaPerp1[[iterate++]];
nMax = PaPerp1[[iterate++]];
fMin = PaPerp1[[iterate++]];
fMax = PaPerp1[[iterate++]] ;
res = PaPerp1[[iterate++]] ;
Bo = PaPerp1[[iterate++]];
Bp = PaPerp1[[iterate++]];
numEnergies = PaPerp1[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[PaPerp1[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];

fullImage = ListLinePlot[starkMap[[2200 ;; 2700]], PlotRange → {{0, 5}, {-94.5, -99}}]
```



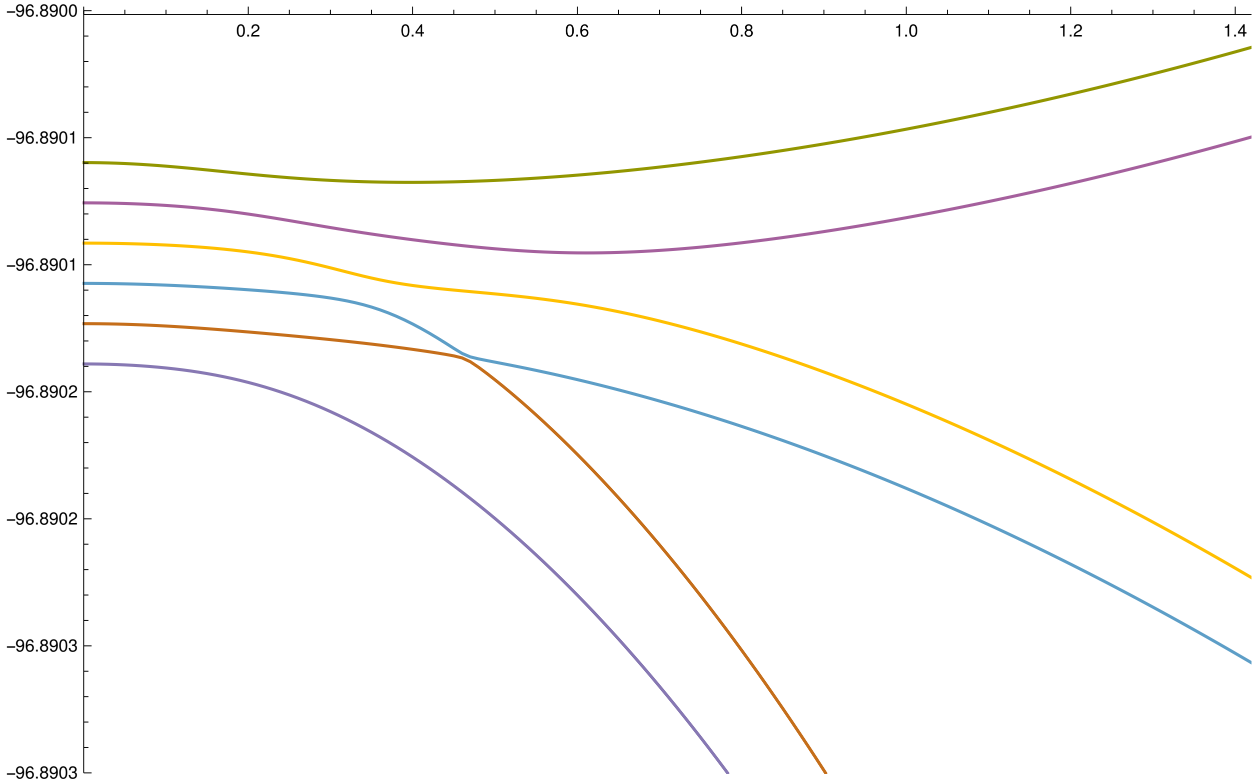
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 2}, {-96.88985, -96.8905}}]
```



This study looks at the map with 0.2 Gauss Parallel Magnetic Field and a 0.2 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
ParPerp02 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/ParPerp.2/ParPerp.2_n_33_34_f_0_5
     _res_0.01_Bo_0.2_Bp_0.2.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[ParPerp02[[iterate++]]];
numFields = ParPerp02[[iterate++]];
nMin = ParPerp02[[iterate++]];
nMax = ParPerp02[[iterate++]];
fMin = ParPerp02[[iterate++]];
fMax = ParPerp02[[iterate++]] ;
res = ParPerp02[[iterate++]] ;
Bo = ParPerp02[[iterate++]];
Bp = ParPerp02[[iterate++]];
numEnergies = ParPerp02[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[ParPerp02[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

```
parperp2 =
 ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8900, -96.8903}}]
```

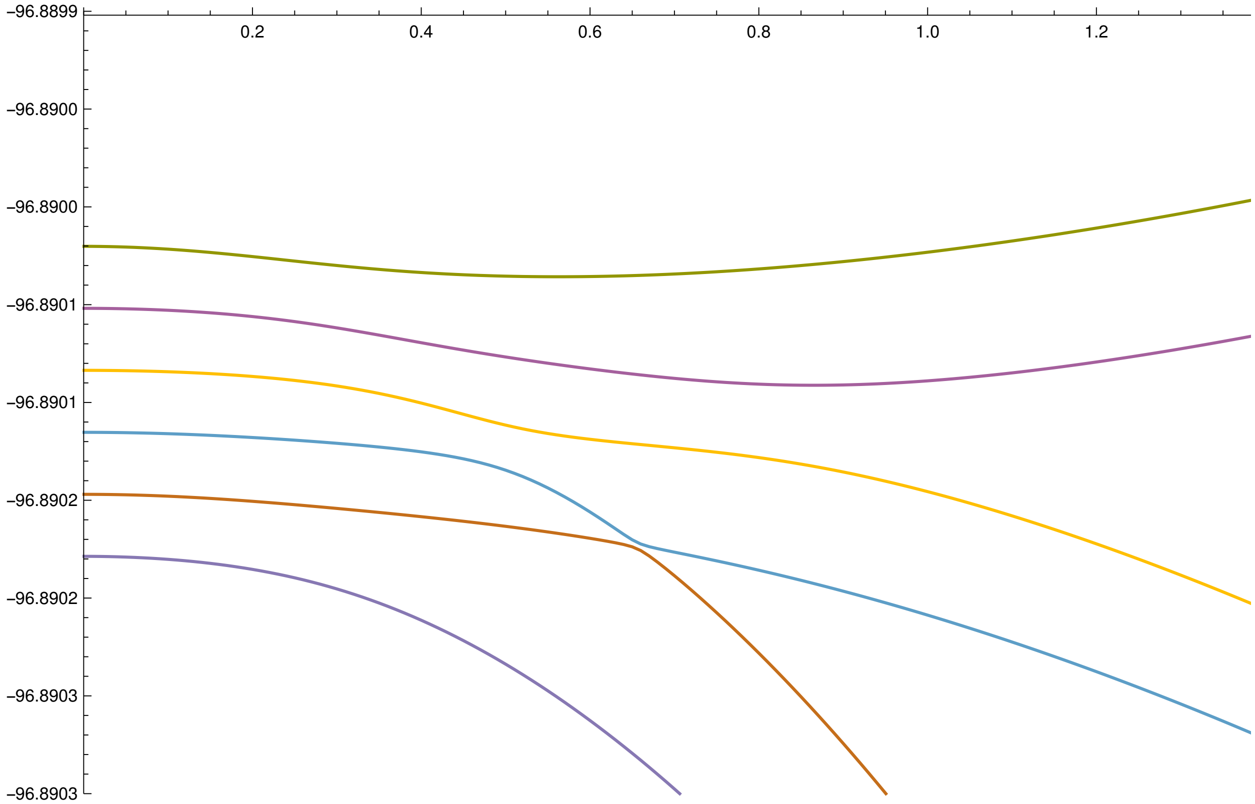


```
Export["parPerp.2.pdf", parperp2];
```

This study looks at the map with 0.4 Gauss Parallel Magnetic Field and a 0.4 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
ParPerp04 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/ParPerp.4/ParPerp.4_n_33_34_f_0_5
     _res_0.01_Bo_0.4_Bp_0.4.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[ParPerp04[[iterate++]]];
numFields = ParPerp04[[iterate++]];
nMin = ParPerp04[[iterate++]];
nMax = ParPerp04[[iterate++]];
fMin = ParPerp04[[iterate++]];
fMax = ParPerp04[[iterate++]] ;
res = ParPerp04[[iterate++]] ;
Bo = ParPerp04[[iterate++]];
Bp = ParPerp04[[iterate++]];
numEnergies = ParPerp04[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[ParPerp04[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

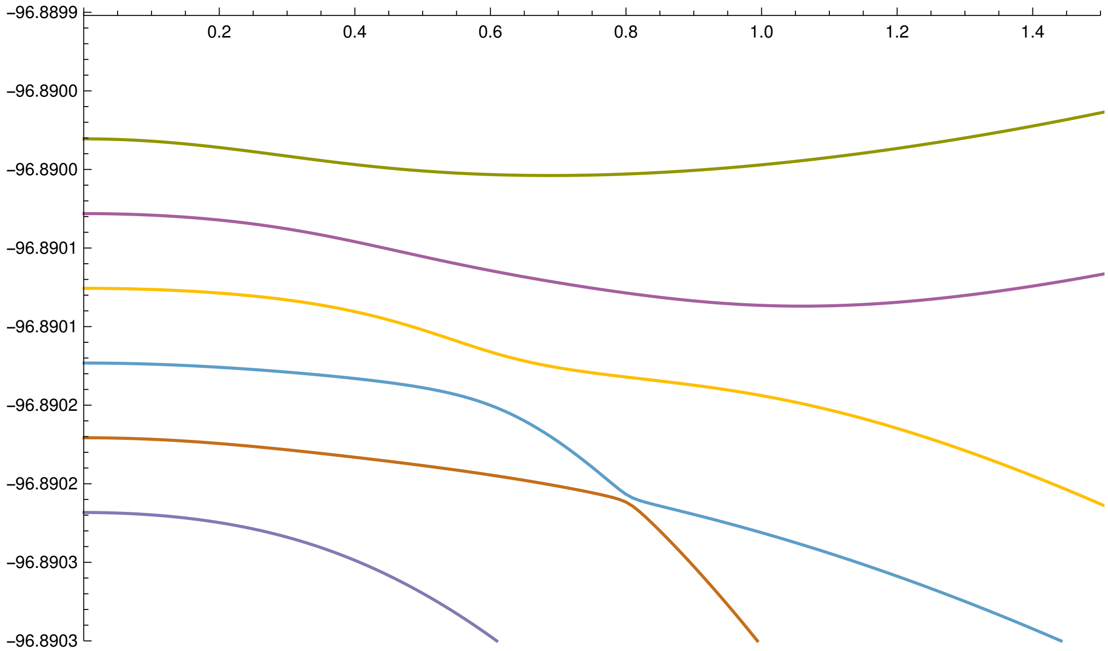
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8899, -96.8903}}]
```



This study looks at the map with 0.6 Gauss Parallel Magnetic Field and a 0.6 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
ParPerp06 = BinaryReadList[
   "/home/zorowley/multivac/experiment/Magno/maps/ParPerp.6/ParPerp.6_n_33_34_f_0_5
     _res_0.01_Bo_0.6_Bp_0.6.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[ParPerp06[[iterate++]]];
numFields = ParPerp06[[iterate++]];
nMin = ParPerp06[[iterate++]];
nMax = ParPerp06[[iterate++]];
fMin = ParPerp06[[iterate++]];
fMax = ParPerp06[[iterate++]] ;
res = ParPerp06[[iterate++]] ;
Bo = ParPerp06[[iterate++]];
Bp = ParPerp06[[iterate++]];
numEnergies = ParPerp06[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[ParPerp06[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
```

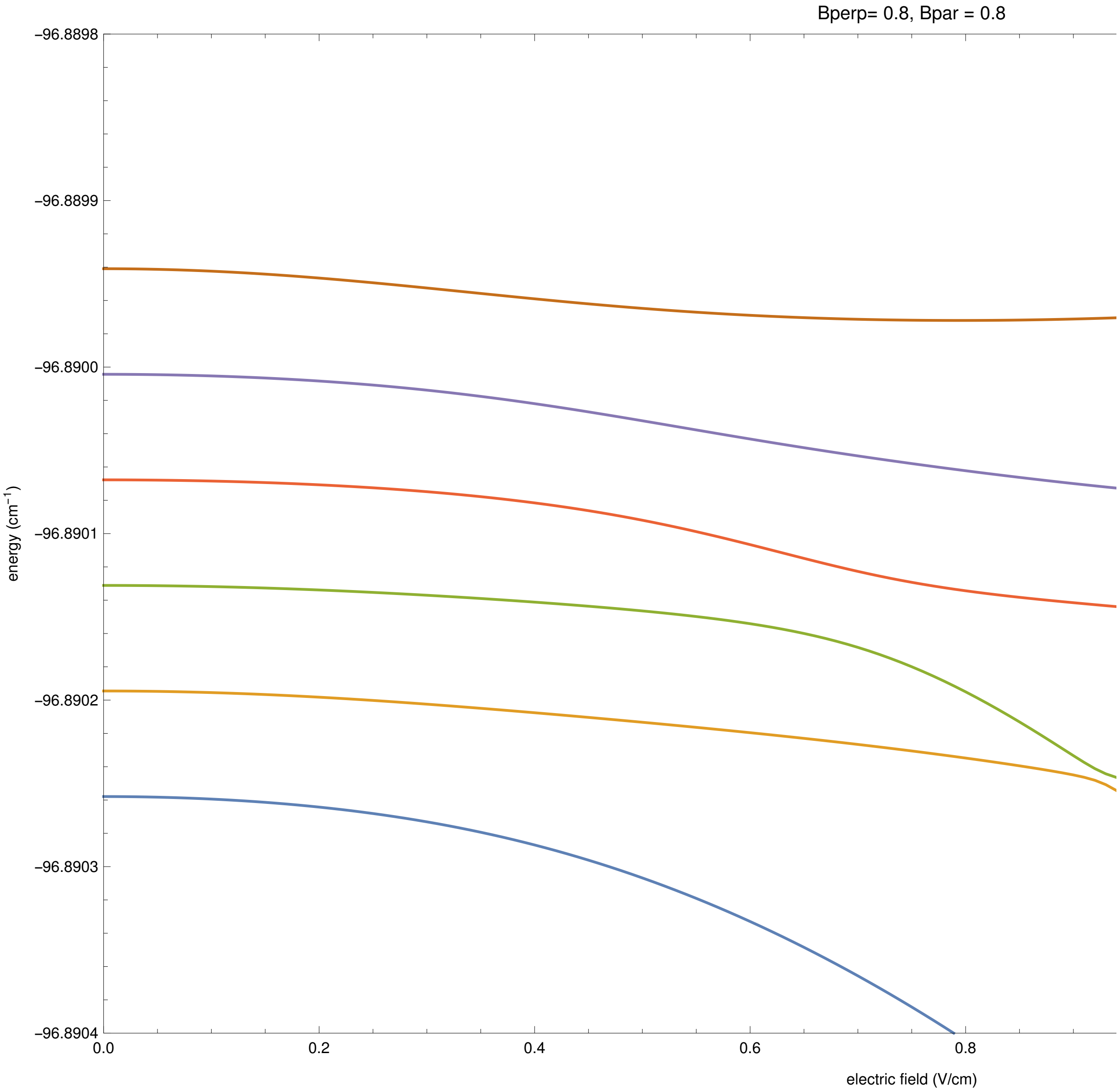
```
ListLinePlot[starkMap[[2213 ;; 2223]], PlotRange → {{0, 1.5}, {-96.8899, -96.8903}}]
```



This study looks at the map with 0.8 Gauss Parallel Magnetic Field and a 0.8 Gauss Perpendicular Magnetic Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

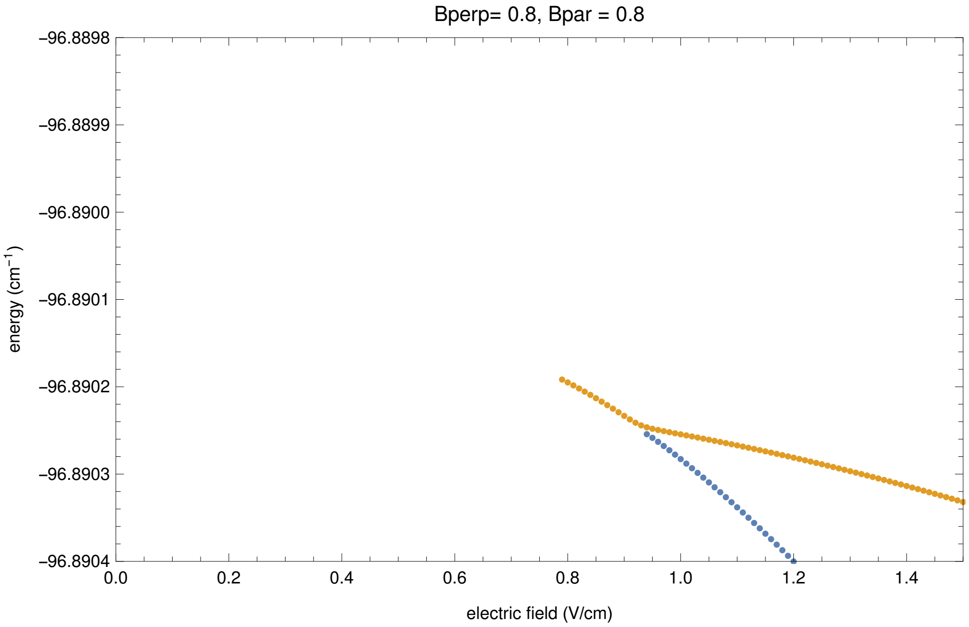
```
ParPerp08 = BinaryReadList[
   "/home/zorowley/Documents/summerFellows2017/ParPerp.8_n_33_34_f_0_5_res_0.01_Bo_
     0.8_Bp_0.8.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[ParPerp08[[iterate++]]];
numFields = ParPerp08[[iterate++]];
nMin = ParPerp08[[iterate++]];
nMax = ParPerp08[[iterate++]];
fMin = ParPerp08[[iterate++]];
fMax = ParPerp08[[iterate++]] ;
res = ParPerp08[[iterate++]] ;
Bo = ParPerp08[[iterate++]];
Bp = ParPerp08[[iterate++]];
numEnergies = ParPerp08[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[ParPerp08[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-30 000 * (energyPar[[i]] - 96.89),
      fMin + (i - 1) * res, {1, -2, 2}], 2], {i, 1, Length[energyPar]}] ];

manif = starkMap[[2217 ;; 2222]];

temp = manif[[3]];
For[i = 96, i ≤ Length[manif[[3]]], i++,
  manif[[3, i]] = manif[[2, i]];
  manif[[2, i]] = temp[[i]];
 ];
```

```
emin = -96.8898;
emax = -96.8904;
fmin = 0;
fmax = 1.5;
parPerp8 = ListLinePlot[manif, PlotRange → {{fmin, fmax}, {emin, emax}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm⁻¹)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000]
```

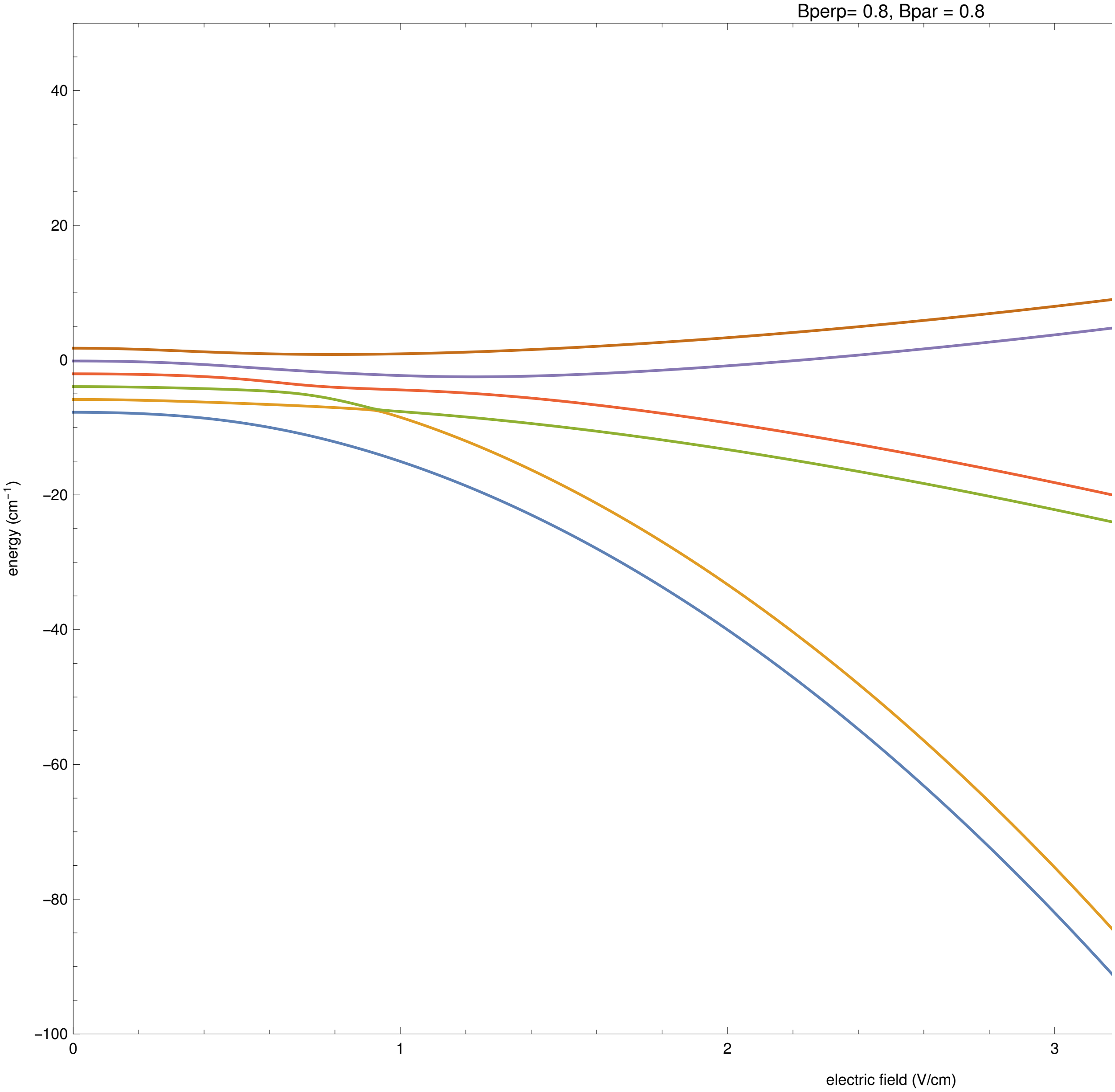


```
Export["parPerp.8.pdf", parPerp8];
```

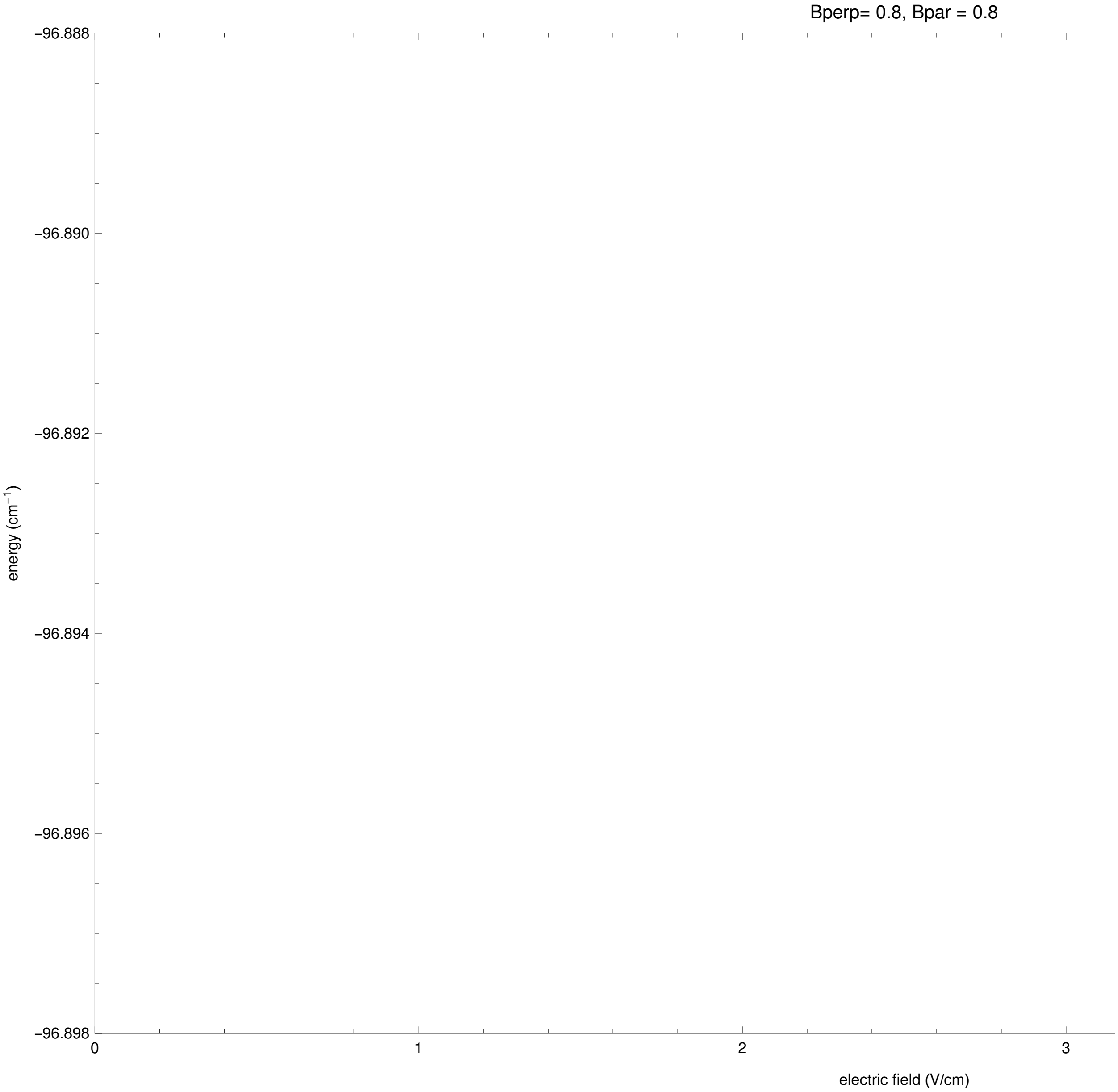
```
parPerp8 = ListPlot[{manif[[2, 95 ;;]], manif[[3, 80 ;;]]},
  PlotRange → {{fmin, fmax}, {emin, emax}}, Frame → True,
  FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 500]
```



```
parPerp8Zoom = ListLinePlot[manif, PlotRange → {{0, 5}, {-100, 50}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp],
  ImageSize → 1000, Axes → False]
```

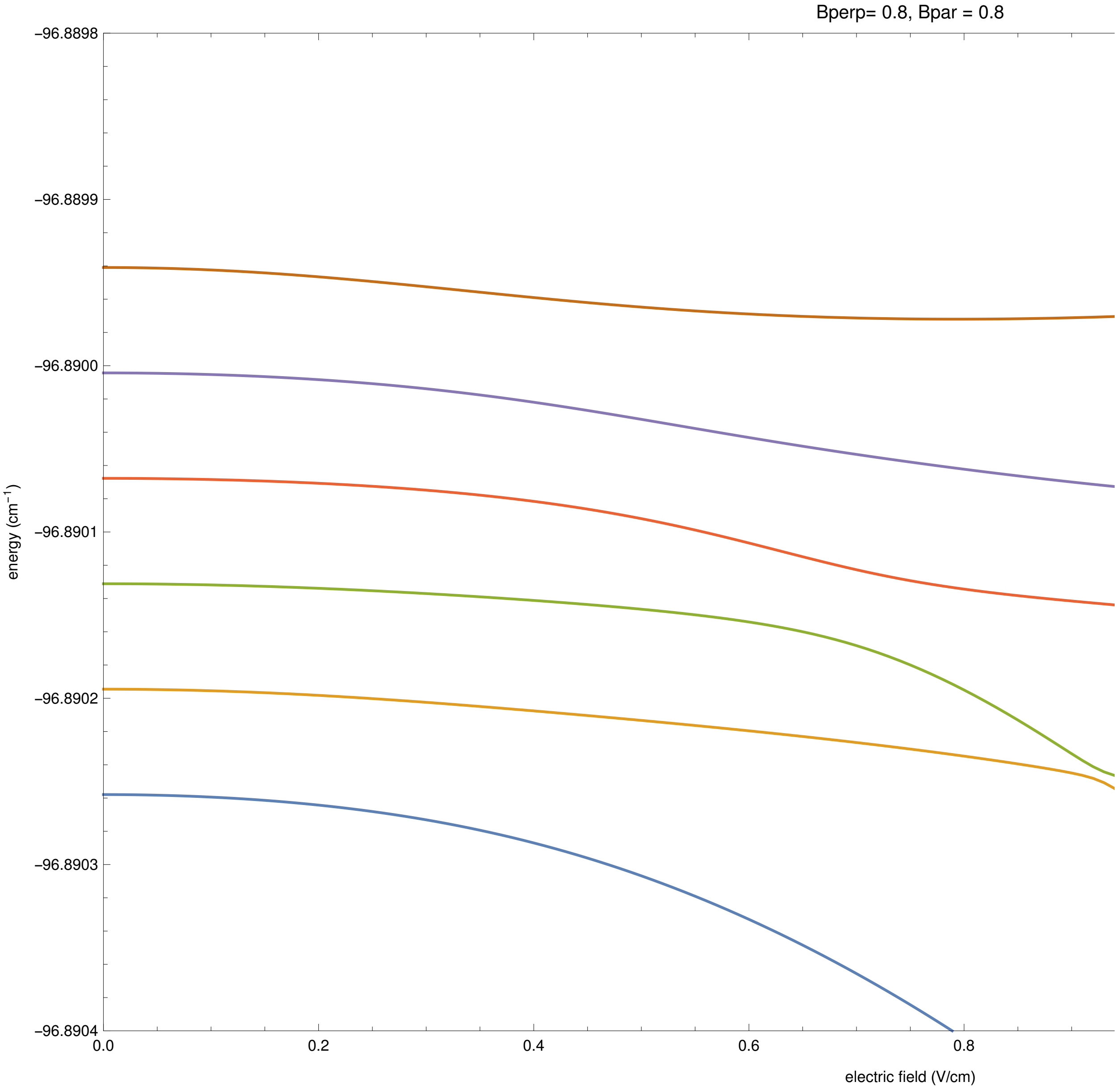


```
parPerp8Zoom = ListLinePlot[manif, PlotRange → {{0, 5}, {-96.888, -96.898}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000]
```

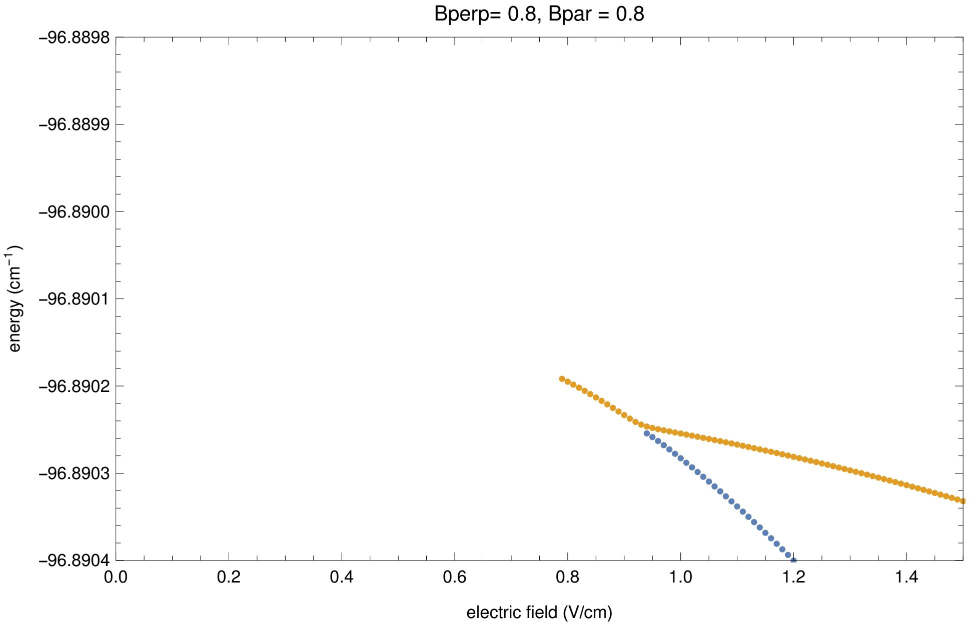


```
This study looks at the map with 0.8 Gauss Parallel
 Magnetic Field and a 0.8 Gauss Perpendicular Magnetic Field at
 a Resolution of .01 and an Electric field ranging 0 to 5 V / cm
ParPerp08 = BinaryReadList[
   "/home/zorowley/Documents/summerFellows2017/ParPerp.8_n_33_34_f_0_5_res_0.01_Bo_
     0.8_Bp_0.8.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
```

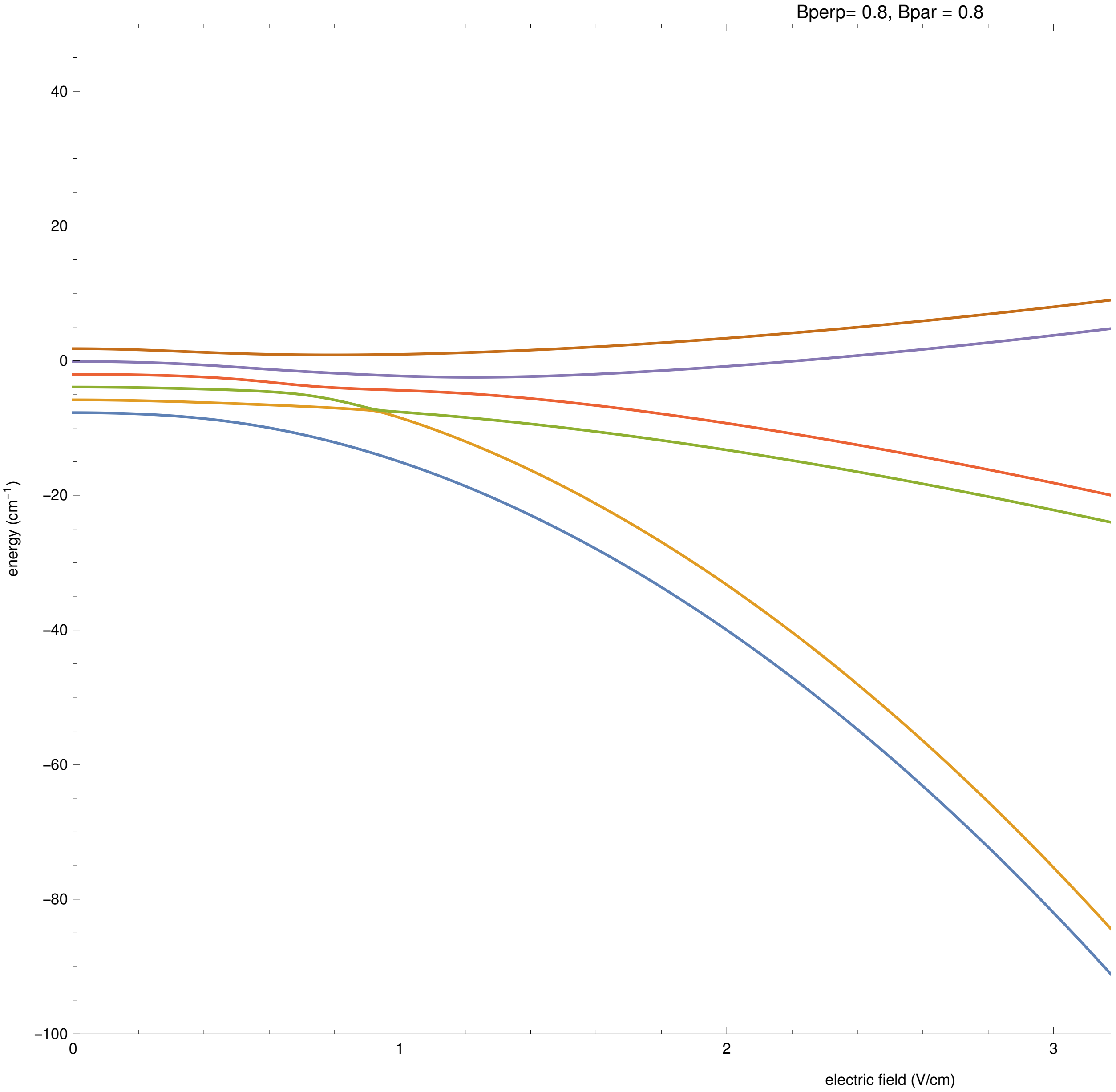
```
headerLength = Round[ParPerp08[[iterate++]]];
numFields = ParPerp08[[iterate++]];
nMin = ParPerp08[[iterate++]];
nMax = ParPerp08[[iterate++]];
fMin = ParPerp08[[iterate++]];
fMax = ParPerp08[[iterate++]] ;
res = ParPerp08[[iterate++]] ;
Bo = ParPerp08[[iterate++]];
Bp = ParPerp08[[iterate++]];
numEnergies = ParPerp08[[iterate++]];

(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[ParPerp08[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-30 000 * (energyPar[[i]] - 96.89),
      fMin + (i - 1) * res, {1, -2, 2}], 2], {i, 1, Length[energyPar]}] ];
manif = starkMap[[2217 ;; 2222]];
temp = manif[[3]];
For[i = 96, i ≤ Length[manif[[3]]], i++,
  manif[[3, i]] = manif[[2, i]];
  manif[[2, i]] = temp[[i]];
 ];
emin = -96.8898;
emax = -96.8904;
fmin = 0;
fmax = 1.5;
parPerp8 = ListLinePlot[manif, PlotRange → {{fmin, fmax}, {emin, emax}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000]
```



```
Export["parPerp.8.pdf", parPerp8];
parPerp8 = ListPlot[{manif[[2, 95 ;;]], manif[[3, 80 ;;]]},
  PlotRange → {{fmin, fmax}, {emin, emax}}, Frame → True,
  FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 500]
```



```
parPerp8Zoom = ListLinePlot[manif, PlotRange → {{0, 5}, {-100, 50}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp],
  ImageSize → 1000, Axes → False]
```



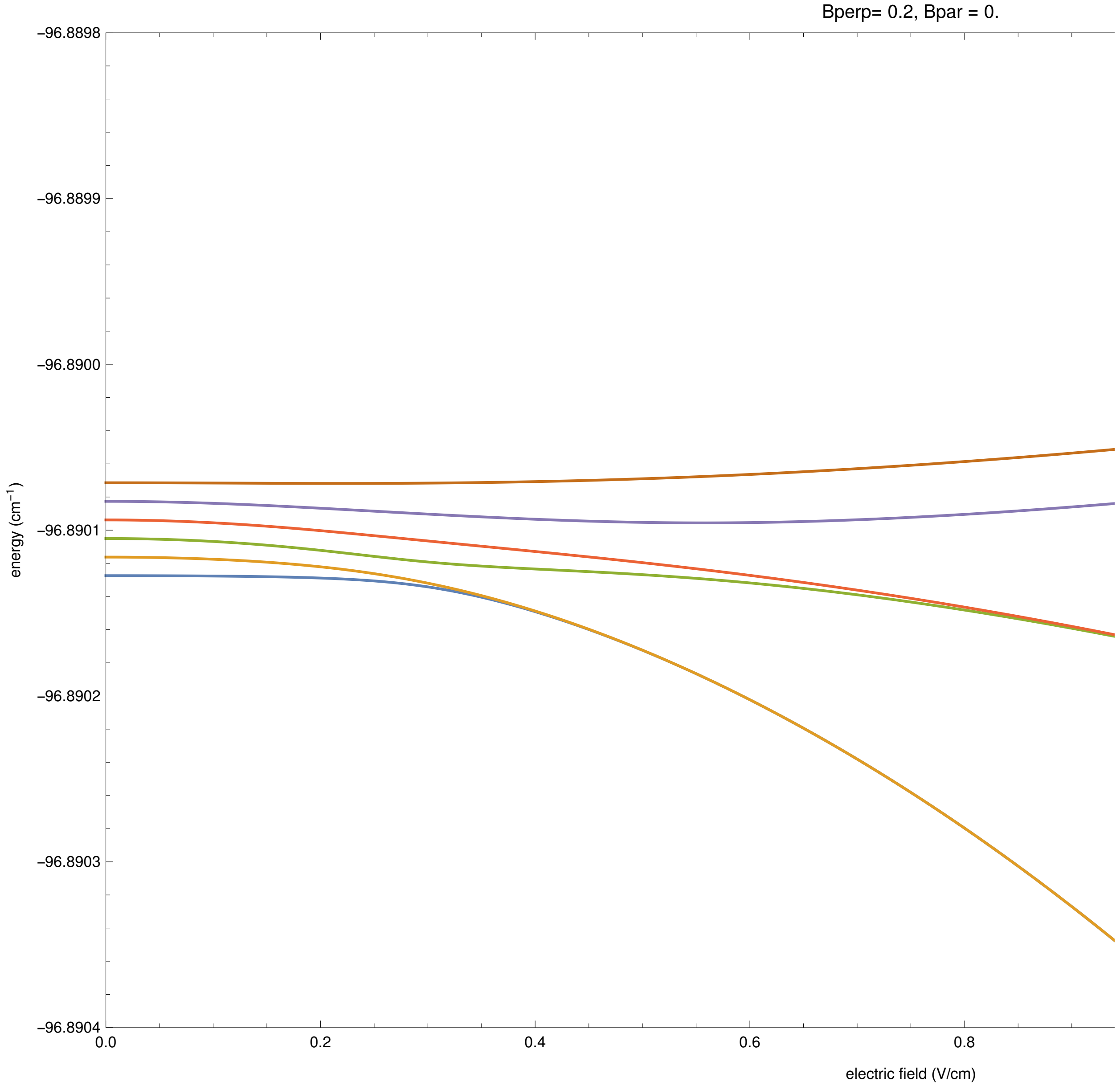
```
parPerp8Zoom = ListLinePlot[manif, PlotRange → {{0, 5}, {-96.888, -96.898}},
  Frame → True, FrameLabel → {"electric field (V/cm)", "energy (cm^-1)"},
  PlotLabel → "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000]
```

This study looks at the map with 0.0 Gauss Parallel Magnetic Field and a 0.2 Gauss Perpendicular Magnetic

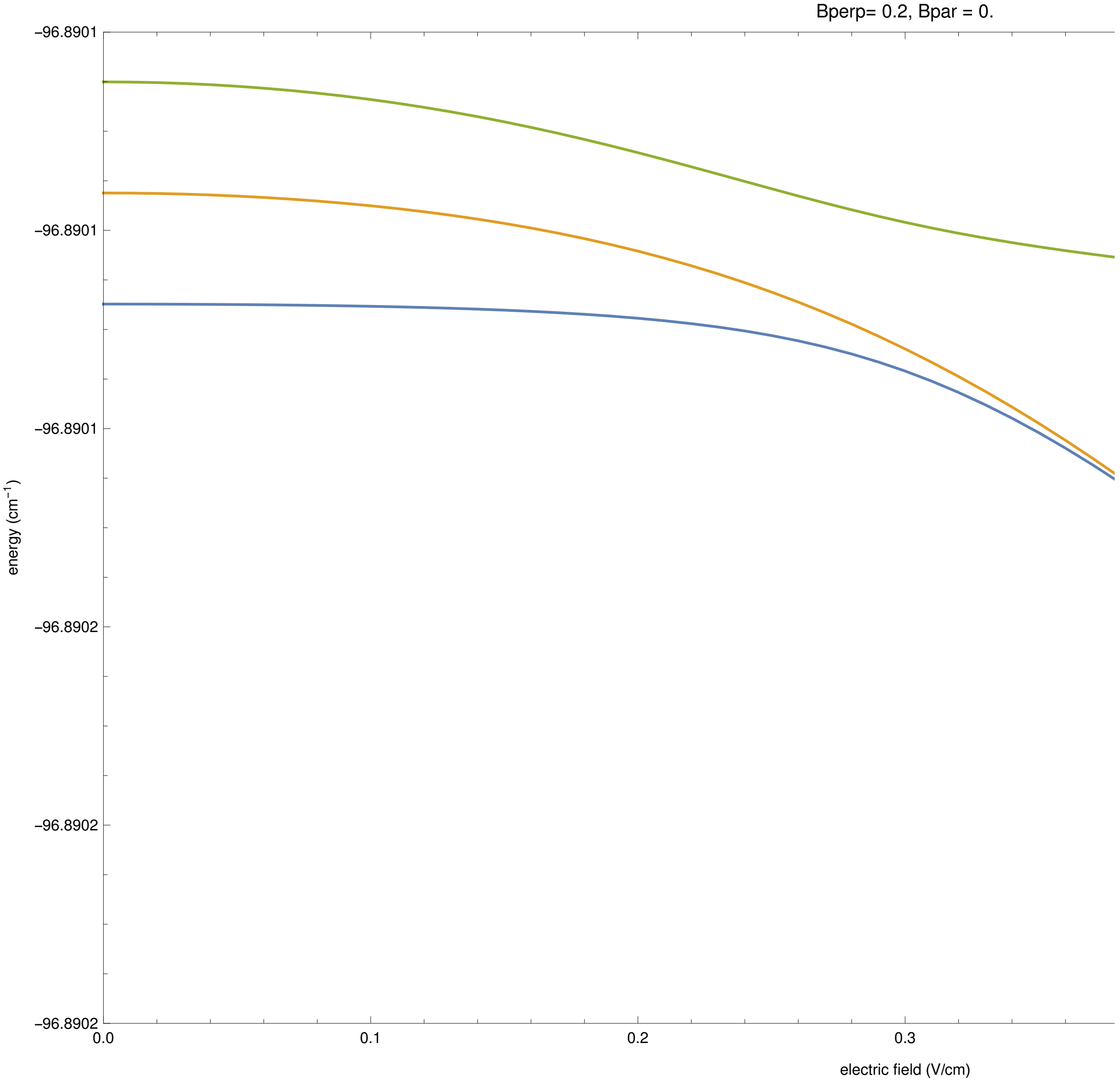
Field at a Resolution of .01 and an Electric field ranging 0 to 5 V/cm

```
noParPerp02 = BinaryReadList[
   "/home/zorowley/Documents/summerFellows2017/noPar_Perp.2_n_33_34_f_0_5_res_0.01_Bo
     _0.2_Bp_0.dat", "Real64" ];
(*Gather and define all information from the header within the .dat file*)
iterate = 1;
headerLength = Round[noParPerp02[[iterate++]]];
numFields = noParPerp02[[iterate++]];
nMin = noParPerp02[[iterate++]];
nMax = noParPerp02[[iterate++]];
fMin = noParPerp02[[iterate++]];
fMax = noParPerp02[[iterate++]] ;
res = noParPerp02[[iterate++]] ;
Bo = noParPerp02[[iterate++]];
Bp = noParPerp02[[iterate++]];
numEnergies = noParPerp02[[iterate++]];
(*Partition through the file (readIn) and remove the entries that
 make up the header, just leaving the energy fields in the file.*)
energyPar = Partition[noParPerp02[[(headerLength + 1) ;;]], Round[numEnergies] ];
(*This table uses Riffle (1,-2,2) which takes a line of the energy
 values that was read in (energyPar[[i]]) and places its respecitve
 electric field value infront of it. Partition, 2 then puts the
 electric field and its corresponding energy level into an ordered pair.
 *)
starkMap = Transpose[
   Table[Partition[Riffle[-1 * energyPar[[i]], fMin + (i - 1) * res, {1, -2, 2}], 2],
    {i, 1, Length[energyPar]}] ];
manif = starkMap[[2217 ;; 2222]];
emin = -96.8898;
emax = -96.8904;
fmin = 0;
fmax = 1.5;
```

```
avoidedPar0Perp02 = ListLinePlot[
  manif, PlotRange → {{fmin, fmax}, {emin, emax}}, Frame -> True,
  FrameLabel → {"electric field (V/cm)", "energy (cm-1)"}, PlotLabel →
   "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000
 ]
```



```
zoom = ListLinePlot[
  starkMap[[2217 ;; 2219]], PlotRange → {{0, .6}, {-96.8901, -96.8902}}, Frame -> True,
  FrameLabel → {"electric field (V/cm)", "energy (cm-1)"}, PlotLabel →
   "Bperp= " <> ToString[Bo] <> ", Bpar = " <> ToString[Bp], ImageSize → 1000
 ]
```



Conclusion

Through out this research, we have consistently found that whenever a perpendicular magnetic field is introduced, it will always take over and by the end, couple together like m_j states. We have also found that whenever a perpendicular field is applied, that we will always see a coupling between the m_j= 3/2 and m_j= 5/2 characteristic states, which is consistent with the population transfer between these two states that has been observed in the laboratory set up at Bryn Mawr College.

Acknowledgements

We would like to thank our Mentor, Dr. Thomas Carroll and his super computer, Slurm. We would also like to thank Dr. Michael Noel at Bryn Mawr College for allowing us to work in his lab with him. Special shout out to Jason Bennet and Kevin Choice for helping out too.

All of this work is supported by the NSF Grants No. 1607335 and No. 1607377.