Van der Waals Interactions

```
Clear[]
Off[General::"spell1"]
<< "Units`"
$MaxPrecision = 150; $MachinePrecision;
$MaxExtraPrecision = 150; $MaxMachineNumber;
```

Rb^{87} data and other constants

Natural constants

```
hbar = 1.054571596 * 10^-34; (*Js, ref. Steck*)
Head[hbar];
SetPrecision[hbar, 15]; (*to show hbar with 15 numbers*)
Precision[hbar] (*hbar is given in Machine Precision*);
h = hbar * 2. * π;
μ0 = 4. * π * 10^-7 ; (*N/A^2, exact, ref. Steck*)
c = 2.99792458 * 10^8; (*m/s, exact, ref. Steck*)
ϵ0 = (μ0 c^2)^-1; (*F/m, exact, ref. Steck*)
kB = 1.3806503 * 10^-23 ;(*J/K, ref. Steck*)
me = 9.10938188 * 10^-31 (*ElectronMass [kg], ref. Steck*);
eSI = 1.602176462 * 10^-19 (*ElectronCharge [C], ref. Steck*);
ecgs = eSI / (4. π ϵ0)^(1/2);
a0 = 0.5291772083 * 10^-10; (*m , Bohr Radius, ref. Steck*)
hbar^2 / (me * ecgs ^2) (*a0 calculated*);
(ecgs^2 / a0) / eSI (*atomic energy unit given in [eV=eSIV]*)
μB = 9.27400899 * 10^-24 ;(*J/T , Bohr Magneton, ref. Steck*)
u = 1.66053873 * 10^-27 ;(*kg, Atomic mass unit , ref. Steck*)
α = eSI^2 / (hbar c 4. π ϵ0);
(SetPrecision[α, 9];)
R∞ = α^2 me c / (4. π hbar) ;(* Rydberg constant: Wikipedia R∞=
 1.0973731568525(73)*10^7 m^-1, slightly different from
 value calculated with constants from ref. Steck *)

SetPrecision[R∞, 13] (*to show R∞ with 13 numbers*)

(*Rubidium*)

mRb87 = 86.909180520 * u; (*kg, Atomic Mass 87Rb, ref. Steck*)
RRb87 = R∞ * (1 + me / mRb87)^-1 ;
(* RRb=109736.605 cm^-1 from PRA 67 052502 (2003) Gallagher,
"where RRb is the Rydberg constant for the reduced
   electron mass in Rb" → obviously for Rb85 !! *)
SetPrecision[RRb87, 9] (*to show R∞ with 9 numbers*)

mRb85 = 84.909180520 * u; (*kg, Atomic Mass 85Rb*)
RRb85 = R∞ * (1 + me / mRb85)^-1 ;
1 - (1 + me / mRb85)^-1 (*relative effect of reduced mass
   (replacing R∞ by RRb87) on Energy is in the order of 6E-6*)
```

```
27.2114
```

$1.097373156880 \times 10^{7}$

1.09736623×10^{7}

6.46074×10^{-6}

Define Functions for calculation of energy-levels without interaction

Quantum defects taken from Publication T. Gallagher et al. PRA 67, 052502 (2003), n>20.
nf_j and ng_j states are found in Han, Gallagher et al. PRA 74 × 054 502 (2006)

```
δ0 = {3.1311804, 2.6548849, 2.6416737,
  1.34809171, 1.34646572, 0.0165192, 0.0165437, 0}
δ2 = {0.1784, 0.2900, 0.2950, -0.60286, -0.59600, -0.085, -0.086, 0};


(*list of quantum defects for Rb85 et Rb87 from PRA 67,
052502 (2003) 1st entry ns1/2: lj=1, 2nd entry np1/2:lj=2,
3nd entry np3/2:lj=3, 4th entry nd3/2: lj=4, 5th entry nd5/2:lj=5,
6th entry nf5/2:lj=6, 7th entry nf7/2:lj=7,n>20,
8th entry for levels with l>f quantum defect is set to zero*)
```

$$\delta[lj\_, n\_] := \delta_0[[lj]] + \frac{\delta_2[[lj]]}{(n - \delta_0[[lj]])^2};$$

```
(*Quantum defect for given lj and n for n>20,
see T. Gallagher "Rydberg Atoms"*)
δ[7, 41]

(*Energy of Rydberg levels with quantum defect (fine structure)*)
```

$$En[lj\_, n\_] := -\frac{R_{Rb87}\, c}{(n - \delta[lj, n])^2}$$ (*[Hz], Energy of Level |n, l,j> with quantum effect and mass correction *)

$$ERadinte[lj\_, n\_] := -\frac{0.5 * (1 + m_e / m_{Rb87})^{-1}}{(n - \delta[lj, n])^2}$$ (*[Unit=Atomic Energy Unit = $e_{cgs}{}^2/a_0$ [J] = 2 h R_∞c [J] = 2*$E_{Hydrogen\ n=1}$], "Energy" as needed by "radinte.exe" of Level |n,l, j> with quantum defect and with reduced mass effect, n>20 *)

(*Effect of mass on Energylevels for Rb^{87} and Rb^{85}*)

$$Enw85[lj\_, n\_] := -\frac{R_{Rb85}\, c}{n^2}$$ (*[Hz], Energy of Level |n, l,j> without quantum effect for Rb85 *)

$SetPrecision[(Enw85[1, 51] - Enw85[1, 50]) * 10^{-9}, 15]$
 (*Transition Frequency in GHz for Rb85 to control*);

```
Enw87[lj_, n_] := - (R_Rb87 c)/n^2 (*[Hz], Energy of Level |n,
 l,j> without quantum effect for Rb87 *);
SetPrecision[(Enw87[1, 51] - Enw87[1, 50]) * 10^-9, 15]
 (*Transition Frequency in GHz for Rb87 to control*);
SetPrecision[(Enw87[1, 51] - Enw87[1, 50]) - (Enw85[1, 51] - Enw85[1, 50]),
 15] (* Difference for Transition Frequencies in Hz for Rb87 and Rb85,
effect in the order of 10kHz *)

(*Define Function Choose to choose lj
 for quantum defect as a function of l and j*)
Chooselj[l_, j_] := If[j == 1 / 2, lj = 1] /; l == 0 (*s states*)
Chooselj[l_, j_] := If[j == 1 / 2, lj = 2, lj = 3] /; l == 1 (*p states*)
Chooselj[l_, j_] := If[j == 3 / 2, lj = 4, lj = 5] /; l == 2 (*d states*)
Chooselj[l_, j_] := If[j == 5 / 2, lj = 6, lj = 7] /; l == 3 (*f states*)
Chooselj[l_, j_] := lj = 8;
```

```
{3.13118, 2.65488, 2.64167, 1.34809, 1.34647, 0.0165192, 0.0165437, 0}
```

```
0.0164925
```

```
7597.31420898438
```

Integration of C-Code for Numerov Integration to calculate radial matrix elements, Tests of accuracy by comparison to Hydrogen

Install the external program "radinte.exe" which will be used to calculate the radial matrix element <E1,l1!R**l!E2,l2> by Numerov Integration in atomic length unit $[a_0{}^l]$ with the following *Mathematica* syntax: "RadIntE[E1_Real,E2_Real,l1_Integer,l2_Integer,l_Integer]". l1 and l2 are magnetic quantum numbers. E1 and E2 are the Energies of the levels |n,l,j> calculated with quantum defect and with mass correction in atomic energy units $[e_{cgs}{}^2/a_0]$ done by ERadinte[lj_,n_].

Attention: The Real numbers E1 and E2 have to be in floating point representation, otherwise Mathlink (communication between radinte.exe and *Mathematica*) will hang up!! The calculation in radinte.exe is done in double precision.

```
Install["/media/2kome/DATA/Works/Lab/Simulation/Mathematica/radinte/./radint
?RadIntE

(*Test: Comparison with existing .exe file *)
RadIntE[
  ERadinte[1,50],-0.0453211,0,1,1]
RadIntE[-0.0453211,ERadinte[1,50],1,0,1]
RadIntE[-0.03661927,ERadinte[1,50],1,0,1]
Timing[Do[RadIntE[-0.03661928,-0.0002276161,1,0,1],{10000}]]
Timing[RadIntE[-0.03661928,-0.0002276161,1,0,1]]
RadIntE[ERadinte[8,60],ERadinte[8,60],10,9,1]
```

```
LinkObject[
 /media/2kome/DATA/Works/Lab/Simulation/Mathematica/radinte/radinte_log,
 738, 12]
```

RadIntE[E1_Real,E2_Real,l1_Integer,l2_Integer,l_Integer]
calculates radial matrix element <E1,l1!R**l!E2,l2> by Numerov Integration.

```
-0.0267742
```

```
-0.0267742
```

```
0.00740212
```

```
{0.292000, Null}
```

```
{0., 0.00741686}
```

```
5323.98
```

Definition of Zeeman Shift energies

```
(*Landé g-factor*)
g[l_, j_] := 3 / 2 + (3 / 4 - l * (l + 1)) / (2 * j * (j + 1));

(* Zeeman shift of the levels in Hz; magnetic field in Tesla *)
ZeemanShift[l_, j_, m_, Bfield_] :=  Bfield * g[l, j] * m * μB / h;

(* 60s Zeeman Shift for 1 Gauss = 10^(-4) Tesla*)
ZeemanShift[0, 1 / 2, 1 / 2, 10^(-4)]
```

1.39962×10^{6}

Definition of Interaction Matrix Elements

```
(*Define Function RInt to calculate
 radial part of VdW Interaction matrix element
<A,B|V_vdW(R)|A',
B'> in [atomic units a_0^2] as a function of atom-atom distance R[m]*)
Off[ClebschGordan::phy];
Off[SixJSymbol::tri];
Off[ClebschGordan::tri];

RInt[R_, {nA_, lA_, jA_, nB_, lB_, jB_},
   {nAp_, lAp_, jAp_, nBp_, lBp_, jBp_}] := (*1/R^3*)
  RadIntE[ERadinte[Chooselj[lA, jA], nA], ERadinte[Chooselj[lAp, jAp],
     nAp], lA, lAp, 1] * RadIntE[ERadinte[Chooselj[lB, jB], nB],
    ERadinte[Chooselj[lBp, jBp], nBp], lB, lBp, 1];

(*Define Function AInt to calculate angular part of VdW Interaction
 matrix element, this gives automatically selection rules *)

AInt[{lA_, jA_, mA_, lB_, jB_, mB_},
   {lAp_, jAp_, mAp_, lBp_, jBp_, mBp_}, θ_] :=
  (- (ThreeJSymbol[{jA, -mA}, {1, 1}, {jAp, mAp}] *
           ThreeJSymbol[{jB, -mB}, {1, -1}, {jBp, mBp}]
          + ThreeJSymbol[{jA, -mA}, {1, -1}, {jAp, mAp}] *
           ThreeJSymbol[{jB, -mB}, {1, 1}, {jBp, mBp}]) * (2 - 3 (Sin[θ]) ^2) *
      0.5 - (ThreeJSymbol[{jA, -mA}, {1, 1}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, 1}, {jBp, mBp}]
         + ThreeJSymbol[{jA, -mA}, {1, -1}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, -1}, {jBp, mBp}]) * (Sin[θ]) ^2 * 1.5 +
     ThreeJSymbol[{jA, -mA}, {1, 0}, {jAp, mAp}] *
      ThreeJSymbol[{jB, -mB}, {1, 0}, {jBp, mBp}]
      * (1 - 3 (Cos[θ]) ^2) - 1.5 * Sin[θ] * Cos[θ] * Sqrt[2] *
      (ThreeJSymbol[{jA, -mA}, {1, -1}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, 0}, {jBp, mBp}] -
        ThreeJSymbol[{jA, -mA}, {1, 1}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, 0}, {jBp, mBp}] +
        ThreeJSymbol[{jA, -mA}, {1, 0}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, -1}, {jBp, mBp}] -
        ThreeJSymbol[{jA, -mA}, {1, 0}, {jAp, mAp}] *
          ThreeJSymbol[{jB, -mB}, {1, 1}, {jBp, mBp}]
      )) * (-1) ^(jA+mA) * (-1) ^(-1/2+jAp+1) Sqrt[(2 jA + 1) * (2 jAp + 1)]
   SixJSymbol[{jA, 1, jAp}, {lAp, 1 / 2, lA}] * Sqrt[(2 lA + 1) * (2 lAp + 1)] *
   ThreeJSymbol[{lA, 0}, {1, 0}, {lAp, 0}] * (-1) ^(jB+mB) * (-1) ^(-1/2+jBp+1)
   Sqrt[(2 jB + 1) * (2 jBp + 1)] SixJSymbol[{jB, 1, jBp}, {lBp, 1 / 2, lB}] *
   Sqrt[(2 lB + 1) * (2 lBp + 1)] * ThreeJSymbol[{lB, 0}, {1, 0}, {lBp, 0}];
```

Calculation of Interaction Matrix Elements $< A, B \mid V_{vdW} |A',B'>$

text

```
(*Define Initial levels*)
(*Atom1, |n1,l1,s1,j1,m1>*)
n1 = 60; l1 = 59; s1 = 1 / 2; j1 = l1 + 1 / 2; m1 = j1;
(*Atom2, |n2,l2,s2,j2,m2>*)
n2 = 59; l2 = 58; s2 = 1 / 2; j2 = l2 + 1 / 2; m2 = j2;
(*VdW coupling between <n1,l1,j1,m1;n2,l2,j2,m2| and |np1,
lp1,jp1,mp1;np2,lp2,jp2,mp2> only possible for M=
 m1+m2=mp1+mp2 and same parity of states (-1)^(l1+l2)=(-1)^(lp1+lp2) *)

(*Set up magnetic field in Tesla. 1 Gauss = 10^(-4) Tesla *)
Bfield = 8.71 * 10 ^ (-4);
(* Define angle b/w diatom and quantization axis, Bfield *)
θ = 0 * ArcCos[1 / Sqrt[3]];

(* Calculate Energy E12=E1+E2 of initial level of product state
<n1,l1,s1,j1,m1|<n2,l2,s2,j2,
m2| and make a list of all other product states
|n1',l1',s1',j1',m1'>|n2',l2',s2',j2',m2'>
 (from n= nmin to nmax ad for all lj) that are within E12+-δf [Hz] *)

lj1 = Chooselj[l1, j1];
lj2 = Chooselj[l2, j2];
E12 = En[lj1, n1] + En[lj2, n2];
(*Energy of interested levels without interaction*)
EZeeman12 =
  E12 + ZeemanShift[l1, j1, m1, Bfield] + ZeemanShift[l2, j2, m2, Bfield];
En[lj1, n1];
En[lj2, n2];
Parity = (-1)^(l1+l2);

(*Set up Basis for Hilbertspace*)
nmin = n1 - 4; Δnmax = 6;  lmax = 2;
(*δf=35*10^9; *)
Choix = 10^7; (*Cut off energy for 1st order terms, in Hz*)
Rtest = 10^-6;
Clear[NList, i, k, n, l, ljA, ljB, EA, EB, EAB, EABZeeman]
```

```
(* Search for 1st order coupling terms *)
Clear[NList]
NList = {};
NList2 = {};
(*En[Chooselj[0,1/2],60]
 E12
 EZeeman12*)
```

```
(*ZeemanShift[l1,j1,m1,Bfield]+ZeemanShift[l2,j2,m2,Bfield]*)
Timing[
For[lA = Abs[l1 - 1], lA ≤ l1 + 1, lA += 2,
      For[lB = Abs[l2 - 1], lB ≤ l2 + 1, lB += 2,
           For[ jA = Abs[lA - 1 / 2], jA ≤ lA + 1 / 2, jA ++,
    If[Abs[j1 - 1] ≤ jA ∧ jA ≤ j1 + 1,
             For[ jB = Abs[lB - 1 / 2], jB ≤ lB + 1 / 2, jB ++,
      If[Abs[j2 - 1] ≤ jB ∧ jB ≤ j2 + 1,
           (*     If[(-1)^(l1+l2)==(-1)^(lB+lA),*)
                 ljA = Choosełj[lA, jA];
                 ljB = Choosełj[lB, jB];
                     For[mA = m1 - 1, mA ≤ m1 + 1, mA ++,
        If[-jA ≤ mA ∧ mA ≤ jA,
         For[mB = m2 - 1, mB ≤ m2 + 1, mB ++,
          If[-jB ≤ mB ∧ mB ≤ jB,
           AIntTemp =
            AInt[{lA, jA, mA, lB, jB, mB}, {l1, j1, m1, l2, j2, m2}, θ];
           If[AIntTemp ≠ 0,
            ZeemanShiftAB = ZeemanShift[lA, jA, mA, Bfield] +
              ZeemanShift[lB, jB, mB, Bfield];
            (*Search up levels for A*)
            For[i = 0, i ≤ Δnmax, i ++,
             nA = n1 + i;
             If[nA > lA,
              EA = En[ljA, nA];
              (*Search up levels for B*)
              For[k = 0, k ≤ Δnmax, k ++,
               nB = n2 + k;
               If[nB > lB,
                EB = En[ljB, nB];
                EAB = (EA + EB);
                RIntTemp =
                 RInt[R, {nA, lA, jA, nB, lB, jB}, {n1, l1, j1, n2, l2, j2}];
                VColumnTemp = RIntTemp * AIntTemp;
                (*Print[nA |nB |mA |mB | (VColumnTemp* 1/(10^-6)^3 *a0^2* (eSI^2/(4. π ϵ0))/h)^1 |
                    Choix*Abs[(EAB+ZeemanShiftAB-EZeeman12)]];*)
                If[Abs[(VColumnTemp * 1/(Rtest)^3 * a0^2 * (eSI^2/(4. π ϵ0))/ h)^2] ≥
                  Choix * Abs[(EAB + ZeemanShiftAB - EZeeman12)],
                 NList = Join[NList, {{nA, lA, jA, mA, nB, lB, jB, mB,
                      EAB, mA + mB, (-1)^(lA+lB), EAB + ZeemanShiftAB}}],
```

```
            Break[];
            ]
          ]]; (*End search up levels for B*)
        (*Search down levels for B*)
        For[k = 1, k ≤ Δn_max, k++,
         n_B = n_2 - k;
         If[n_B > l_B,
          EB = En[ljB, n_B];
          EAB = (EA + EB);
          RIntTemp =
           RInt[R, {n_A, l_A, j_A, n_B, l_B, j_B}, {n_1, l_1, j_1, n_2, l_2, j_2}];
          VColumnTemp = RIntTemp * AIntTemp;
          If[Abs[(VColumnTemp * 1/(Rtest)^3 * a_0^2 * e_SI^2/(4. π ϵ_0) / h)^2] ≥
            Choix * Abs[(EAB + ZeemanShift_AB - EZeeman_12)],
           NList = Join[NList, {{n_A, l_A, j_A, m_A, n_B, l_B, j_B, m_B,
                EAB, m_A + m_B, (-1)^(l_A+l_B), EAB + ZeemanShift_AB}}],
           Break[];
          ]
         ]];(*End search down levels for B*)
      ]]; (* End search up for A*)
    (*Search down levels for A*)
    For[i = 1, i ≤ Δn_max, i++,
     n_A = n_1 - i;
     If[n_A > l_A,
      EA = En[ljA, n_A];
      (*Search up levels for B*)
      For[k = 0, k ≤ Δn_max, k++,
        n_B = n_2 + k;
        If[n_B > l_B,
         EB = En[ljB, n_B];
          EAB = (EA + EB);
          RIntTemp =
           RInt[R, {n_A, l_A, j_A, n_B, l_B, j_B}, {n_1, l_1, j_1, n_2, l_2, j_2}];
          VColumnTemp = RIntTemp * AIntTemp;
          (*Print[n_A | n_B | m_A | m_B | (VColumnTemp* 1/(10^-6)^3 *a_0^2* e_SI^2/(4. π ϵ_0) / h)^1 |
             Choix*Abs[(EAB+ZeemanShift_AB-EZeeman_12)]];*)
          If[Abs[(VColumnTemp * 1/(Rtest)^3 * a_0^2 * e_SI^2/(4. π ϵ_0) / h)^2] ≥
            Choix * Abs[(EAB + ZeemanShift_AB - EZeeman_12)],
           NList = Join[NList, {{n_A, l_A, j_A, m_A, n_B, l_B, j_B, m_B,
                EAB, m_A + m_B, (-1)^(l_A+l_B), EAB + ZeemanShift_AB}}],
```

```
                    Break[];
                    ]
                  ]]; (*End search up levels for B*)
                 (*Search down levels for B*)
                 For[k = 1, k ≤ Δn_max, k++,
                  n_B = n_2 - k;
                  If[n_B > l_B,
                   EB = En[ljB, n_B];
                   EAB = (EA + EB);
                   RIntTemp =
                    RInt[R, {n_A, l_A, j_A, n_B, l_B, j_B}, {n_1, l_1, j_1, n_2, l_2, j_2}];
                   VColumnTemp = RIntTemp * AIntTemp;
                   If[Abs[(VColumnTemp * 1/(Rtest)^3 * a_0^2 * e_SI^2/(4. π ϵ_0) / h)^2] ≥
                      Choix * Abs[(EAB + ZeemanShift_AB - EZeeman_12)],
                    NList = Join[NList, {{n_A, l_A, j_A, m_A, n_B, l_B, j_B, m_B,
                         EAB, m_A + m_B, (-1)^(l_A+l_B), EAB + ZeemanShift_AB}}],
                    Break[];
                   ]
                  ]];(*End search down levels for B*)
               ]](* End search down for A*)
              ]
             ]]
           ]]
         ]]
       ]]
     ]
    ]
   ]


(*Old Code for level selection
  NList={};

Timing[For[i=n_min,i≤n_max,i++,
  For[k=n_min,k≤n_max,k++,
       For[l_A=0,l_A≤l_max, l_A++,
        For[l_B=0,l_B≤l_max, l_B++,
             For[ j_A=Abs[l_A-1/2],j_A≤l_A+1/2,j_A++,
              For[ j_B=Abs[l_B-1/2],j_B≤l_B+1/2,j_B++,
                   ljA=Chooselj[l_A,j_A];
                   ljB=Chooselj[l_B,j_B];
```

```
                            EA=En[ljA,i];
                            EB=En[ljB,k];
                            EAB=(EA+EB);
                            If[(E12-δf)<EAB<(E12+δf),
          For[mA=-jA,mA≤jA,mA++,
           For[mB=-jB,mB≤jB,mB++,
            If[(Abs[mB-m2]≤1∧Abs[mA-m1]≤1)∧((Abs[(mA-mB)-(m1-m2)]≤1)∧
                 (mA+mB≠m1+m2))∧((-1)^(l1+l2)==(-1)^(lB+lA)),
              EABZeeman = EAB+ZeemanShift[i,lA,jA,mA,Bfield]+
                ZeemanShift[k,lB,jB,mB,Bfield];
              NList=Join[NList,{{i,lA,jA,mA,k,lB,jB,mB,EAB,mA+
                    mB,(-1)^(lA+lB),EABZeeman}}]]]]]]
        ]]
      ]]
   ]]]*)
NList;
Print["# 1st order terms = ", Length[NList]]

For[u = 1, u ≤ Length[NList], u++,
  (*Print[u];*)
  If[Intersection[NList[[All, 1 ;; 8]],
     {NList[[u, #]] & /@ {5, 6, 7, 8, 1, 2, 3, 4}}] == {},
   NList = Join[NList, {NList[[u, #]] & /@
        {5, 6, 7, 8, 1, 2, 3, 4, 9, 10, 11, 12}}]]
]
(*LengthNList=Length[NList];*)
Print["# 1st order terms = ", Length[NList]]
```

```
{0.388000, Null}
```

```
# 1st order terms = 5

# 1st order terms = 10
```

```
(* Search for 2nd order coupling terms *)
Start = AbsoluteTime[];
Dynamic[Refresh[Round[AbsoluteTime[] - Start], UpdateInterval → 1]]
Dynamic[{nA, lA, jA, mA, nB, lB - 1, jB, mB}]
Dynamic[Length[NList2]]
Choix2 = 1 * Choix;
Rtest2 = 2 * Rtest;
Δnmax2 = 4;
(*Clear[lA,lB,i,k]*)
Timing[
 For[lA = Abs[Abs[l1 - 1] - 1], lA ≤ l1 + 2, lA += 2,
      For[lB = Abs[Abs[l2 - 1] - 1], lB ≤ l2 + 2, lB += 2,
           For[ jA = Abs[lA - 1 / 2], jA ≤ lA + 1 / 2, jA++,
              For[ jB = Abs[lB - 1 / 2], jB ≤ lB + 1 / 2, jB++,
                (*If[(-1)^(l1+l2)==(-1)^(lB+lA),]*)
```

```
ljA = Chooselj[l_A, j_A];
ljB = Chooselj[l_B, j_B];
For[m_A = m_1 - 2, m_A ≤ m_1 + 2, m_A ++,
If[-j_A ≤ m_A ⋀ m_A ≤ j_A,
For[m_B = m_2 - 1, m_B ≤ m_2 + 1, m_B ++,
If[-j_B ≤ m_B ⋀ m_B ≤ j_B,
For[u = 1, u ≤ Length[NList], u ++,
AIntTemp =
 AInt[{l_A, j_A, m_A, l_B, j_B, m_B}, NList[[u, {2, 3, 4, 6, 7, 8}]], θ];
If[AIntTemp ≠ 0,
 ZeemanShift_AB = ZeemanShift[l_A, j_A, m_A, Bfield] +
   ZeemanShift[l_B, j_B, m_B, Bfield];
 (*Search up levels for A*)
 For[i = 0, i ≤ Δn_max2, i ++,
   n_A = NList[[u, 1]] + i;
   If[n_A > l_A,
    (*Search up levels for B*)
    For[k = 0, k ≤ Δn_max2, k ++,
     n_B = NList[[u, 5]] + k;
     If[n_B > l_B,
      If[Intersection[NList[[All, 1 ;; 8]], {{n_A, l_A, j_A, m_A, n_B,
           l_B, j_B, m_B}}] == {} ⋀ Intersection[NList2[[All,
          1 ;; 8]], {{n_A, l_A, j_A, m_A, n_B, l_B, j_B, m_B}}] == {},
       (*Print["True",n_A,l_A,j_A,m_A,n_B,l_B,j_B,m_B]*)
       EB = En[ljB, n_B];
       EA = En[ljA, n_A];
       EAB = (EA + EB);
       RIntTemp = RInt[R, {n_A, l_A, j_A, n_B, l_B, j_B},
         NList[[u, {1, 2, 3, 5, 6, 7}]]];
       VColumnTemp = RIntTemp * AIntTemp;
       (*Print[n_A | n_B | m_A | m_B | (VColumnTemp* 1/(10^-6)^3 *a_0^2* e_SI^2/(4. π ϵ_0) / h)^1 |
          Choix*Abs[(EAB+ZeemanShift_AB-EZeeman_12)]];*)
       If[Abs[(VColumnTemp * 1/(Rtest2)^3 * a_0^2 * e_SI^2/(4. π ϵ_0) / h)^2] ≥
         Choix2 * Abs[(EAB + ZeemanShift_AB - NList[[u, 12]])],
        NList2 = Join[NList2, {{n_A, l_A, j_A, m_A, n_B, l_B, j_B,
            m_B, EAB, m_A + m_B, (-1)^(l_A+l_B), EAB + ZeemanShift_AB}}],
        Break[];
        ]
      ]]]; (*End search up levels for B*)
    (*Search down levels for B*)
    For[k = 1, k ≤ Δn_max2, k ++,
```

```
      nB = NList[[u, 5]] - k;
      If[nB > lB,
       If[Intersection[NList[[All, 1 ;; 8]], {{nA, lA, jA, mA, nB,
              lB, jB, mB}}] == {} ∧ Intersection[NList2[[All,
             1 ;; 8]], {{nA, lA, jA, mA, nB, lB, jB, mB}}] == {},
         EB = En[ljB, nB];
         EAB = (EA + EB);
         RIntTemp = RInt[R, {nA, lA, jA, nB, lB, jB},
           NList[[u, {1, 2, 3, 5, 6, 7}]]];
         VColumnTemp = RIntTemp * AIntTemp;
         (*Print[nA | nB | mA | mB | (VColumnTemp* 1/(10^-6)^3 *a0^2* (eSI^2/(4. π ϵ0))/h)^1 |
            Choix*Abs[(EAB+ZeemanShiftAB-EZeeman12)]];*)
         If[Abs[(VColumnTemp * 1/(Rtest2)^3 * a0^2 * (eSI^2/(4. π ϵ0))/ h)^2] ≥
           Choix2 * Abs[(EAB + ZeemanShiftAB - NList[[u, 12]])],
          NList2 = Join[NList2, {{nA, lA, jA, mA, nB, lB, jB,
               mB, EAB, mA + mB, (-1)^(lA+lB), EAB + ZeemanShiftAB}}],
          Break[];
          ]
       ]]](*End search down levels for B*)
    ]](* End search up for A*)
   (*Search down levels for A*)
   For[i = 1, i ≤ Δnmax2, i ++,
    nA = NList[[u, 1]] - i;
    If[nA > lA,
     (*Search up levels for B*)
     For[k = 0, k ≤ Δnmax2, k ++,
      nB = NList[[u, 5]] + k;
      If[nB > lB,
       If[Intersection[NList[[All, 1 ;; 8]], {{nA, lA, jA, mA, nB,
              lB, jB, mB}}] == {} ∧ Intersection[NList2[[All,
             1 ;; 8]], {{nA, lA, jA, mA, nB, lB, jB, mB}}] == {},
         (*Print["True",nA,lA,jA,mA,nB,lB,jB,mB]*)
         EB = En[ljB, nB];
         EA = En[ljA, nA];
         EAB = (EA + EB);
         RIntTemp = RInt[R, {nA, lA, jA, nB, lB, jB},
           NList[[u, {1, 2, 3, 5, 6, 7}]]];
         VColumnTemp = RIntTemp * AIntTemp;
         (*Print[nA | nB | mA | mB | (VColumnTemp* 1/(10^-6)^3 *a0^2* (eSI^2/(4. π ϵ0))/h)^1 |
            Choix*Abs[(EAB+ZeemanShiftAB-EZeeman12)]];*)
```

```
                    If[Abs[(VColumnTemp * 1/(Rtest2)^3 * a0^2 * eSI^2/(4. π ϵ0) / h)^2] ≥
                      Choix2 * Abs[(EAB + ZeemanShift_AB - NList[[u, 12]])],
                     NList2 = Join[NList2, {{nA, lA, jA, mA, nB, lB, jB,
                          mB, EAB, mA + mB, (-1)^(lA+lB), EAB + ZeemanShift_AB}}],
                     Break[];
                     ]
                  ]]]; (*End search up levels for B*)
                (*Search down levels for B*)
                For[k = 1, k ≤ Δn_max2, k++,
                 nB = NList[[u, 5]] - k;
                 If[nB > lB,
                  If[Intersection[NList[[All, 1 ;; 8]], {{nA, lA, jA, mA, nB,
                          lB, jB, mB}}] == {} ⋀ Intersection[NList2[[All,
                        1 ;; 8]], {{nA, lA, jA, mA, nB, lB, jB, mB}}] == {},
                    EB = En[ljB, nB];
                    EAB = (EA + EB);
                    RIntTemp = RInt[R, {nA, lA, jA, nB, lB, jB},
                      NList[[u, {1, 2, 3, 5, 6, 7}]]];
                    VColumnTemp = RIntTemp * AIntTemp;
                    (*Print[nA | nB | mA | mB | (VColumnTemp* 1/(10^-6)^3 *a0^2* eSI^2/(4. π ϵ0) /h)^1 |
                      Choix*Abs[(EAB+ZeemanShift_AB-EZeeman_12)]];*)
                    If[Abs[(VColumnTemp * 1/(Rtest2)^3 * a0^2 * eSI^2/(4. π ϵ0) / h)^2] ≥
                      Choix2 * Abs[(EAB + ZeemanShift_AB - NList[[u, 12]])],
                     NList2 = Join[NList2, {{nA, lA, jA, mA, nB, lB, jB,
                          mB, EAB, mA + mB, (-1)^(lA+lB), EAB + ZeemanShift_AB}}],
                     Break[];
                     ]
                  ]]](*End search down levels for B*)
               ]](* End search down for A*)
           ]]]]]]]]]]
]
Print[Length[NList2]]
```

```
Round[3.633939456297579 × 10^9 - Start]
```

```
{nA, lA, jA, mA, nB, -1 + lB, jB, mB}
```

```
0
```

```
{10.044000, Null}
```

```
1
```

```
NList = Union[NList, NList2];
Length[NList]
```

```
11
```

```
For[u = 1, u ≤ Length[NList], u++,
 (*Print[u];*)
 If[Intersection[NList[[All, 1 ;; 8]],
    {NList[[u, #]] & /@ {5, 6, 7, 8, 1, 2, 3, 4}}] == {},
  NList = Join[NList, {NList[[u, #]] & /@
      {5, 6, 7, 8, 1, 2, 3, 4, 9, 10, 11, 12}}]]
]
Length[NList];
Print["# 1st + 2nd  order terms = ", Length[NList]]
```

```
# 1st + 2nd  order terms = 12
```

```
Intersection[NList[[All, 1 ;; 8]],
 {NList[[20, #]] & /@ {5, 6, 7, 8, 1, 2, 3, 4}}]
```

$$\left\{\left\{61, 1, \frac{3}{2}, \frac{1}{2}, 59, 1, \frac{3}{2}, \frac{1}{2}\right\}\right\}$$

```
(*NList=Union[NList,NList2];*)
Length[NList]
If[Intersection[NList[[All, 1 ;; 8]], {{n1, l1, j1, m1, n2, l2, j2, m2}}] == {},
  NList = Join[NList, {{n1, l1, j1, m1, n2, l2, j2, m2,
      E12, m1 + m2, (-1)^(l1+l2), EZeeman12}}], Print["Oui1"]];
If[Intersection[NList[[All, 1 ;; 8]], {{n2, l2, j2, m2, n1, l1, j1, m1}}] == {},
  NList = Join[NList, {{n2, l2, j2, m2, n1, l1, j1, m1,
      E12, m1 + m2, (-1)^(l1+l2), EZeeman12}}], Print["Oui2"]];

Intersection[NList[[All, 1 ;; 8]], {{n2, l2, j2, m2, n1, l1, j1, m1}}]

Length[NList]
NList;
```

```
12
```

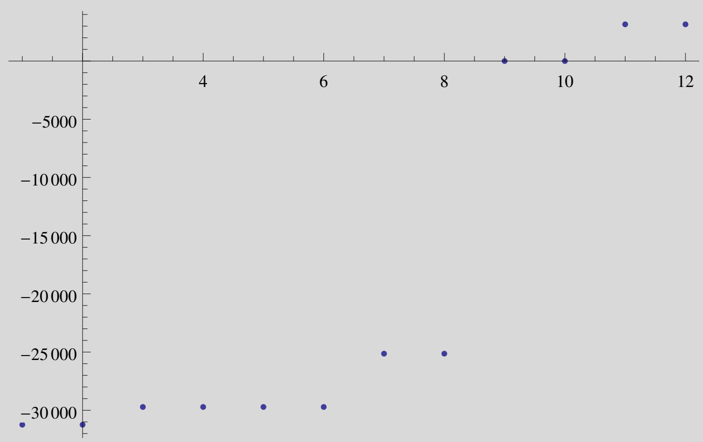
```
Oui1
```

```
Oui2
```

$$\left\{\left\{59, 58, \frac{117}{2}, \frac{117}{2}, 60, 59, \frac{119}{2}, \frac{119}{2}\right\}\right\}$$

```
12
```

```
(*Find minimal Energydistance of found levels to initial levels*)

EDiff = {};
For[i = 1, i ≤ Length[NList], i++,
  EDiff = Join[EDiff, {(E12 - NList[[i, 9]]) / 10^6}]];
ListPlot[EDiff[[Ordering[EDiff]]], PlotRange → {All, All}]
EDiffAbs = {};

For[i = 1, i ≤ Length[NList], i++,
  EDiffAbs = Join[EDiffAbs, {{NList[[i, 1]], NList[[i, 2]],
      NList[[i, 3]], NList[[i, 4]], NList[[i, 5]], NList[[i, 6]],
      NList[[i, 7]], NList[[i, 8]], Abs[(E12 - NList[[i, 9]]) / 10^6]}}]];
EDiffAbs[[Ordering[EDiffAbs[[All, {9}]]]]][[Range[9]]] (*show
 first 20 elements with lowest energy distance to initial level*)

(*Ordering NList
 {{i,lA,jA,mA,k,lB,jB,mB,EAB,mA+mB,(-1)^(lA+lB),EABZeeman}}*)
NList = NList[[Ordering[NList[[All, {12}]]]]] ;
(*Ordering NList beginning with highest Energy,
considering Zeeman shift*)
LabelList = {};
For[i = 1, i ≤ Length[NList], i++,
 LabelList = Join[LabelList, {NList[[i, 1]] |
      NList[[i, 2]] | NList[[i, 3]] | NList[[i, 4]] ‖ NList[[i, 5]] |
      NList[[i, 6]] | NList[[i, 7]] | NList[[i, 8]]}] ;]
(*If [ (n1== NList[[i,1]]) && (l1== NList[[i,2]]) &&
   (j1== NList[[i,3]]) && (m1== NList[[i,4]]) && (n2== NList[[i,5]]) &&
   (l2== NList[[i,6]]) && (j2== NList[[i,7]]) && (m2== NList[[i,8]])
    CentralLevel = i ]*)

(*Make List of Hilbert Space Basis level
 names for Plot of Energies vs. Distance of atoms*)
(*NList=NList[[Ordering[NList[[All,{10,11,9}]]]]] ;*)
(*Elements of NList sorted beginning with M=mA+mB,
then Parity and then highest energy*)
(*NList=NList[[Ordering[NList[[All,{2,3,4,6,7,8}]]]]]*)
(*Elements of NList sorted by order: lA,jA,mA,lB,jB,mB*)
(*NList=NList[[Ordering[NList[[All,{1,5}]]]]]*)
NList ;
LabelList;
```

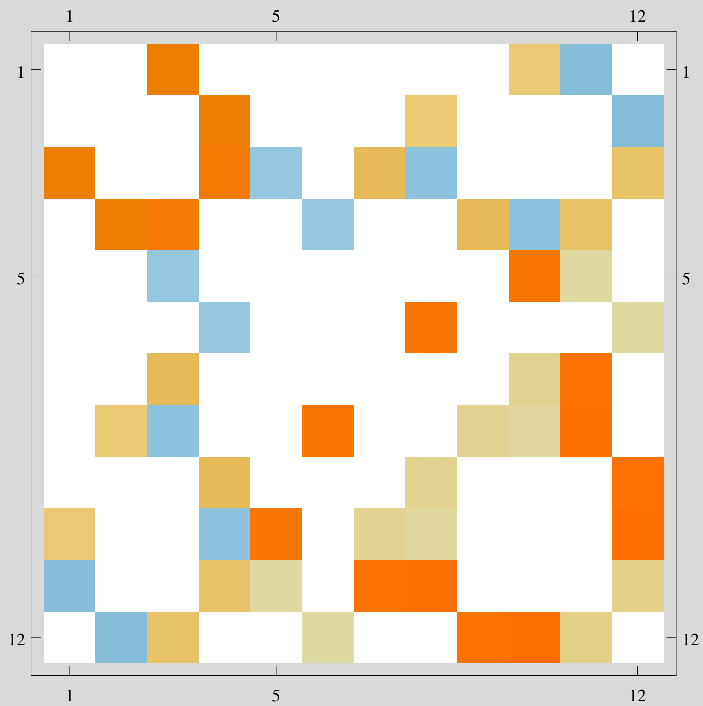


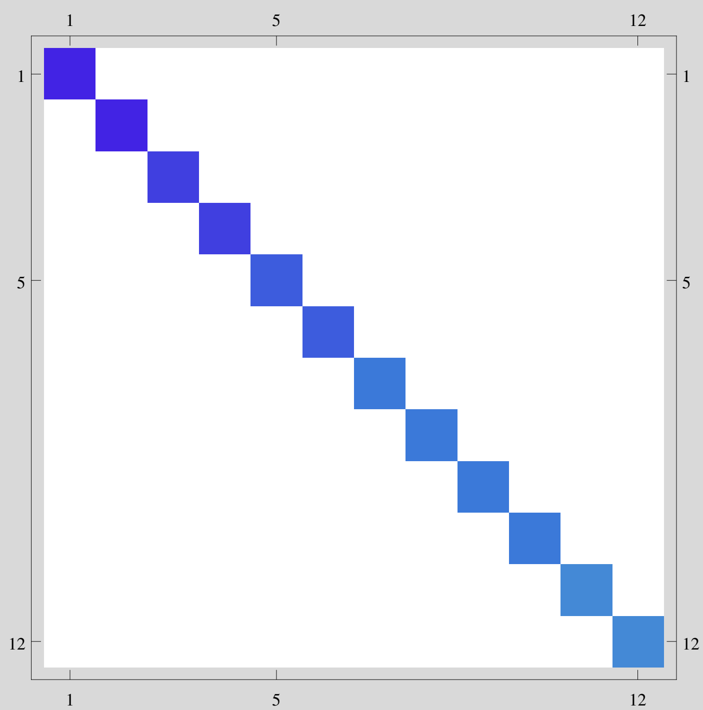
$$\left\{\left\{59, 58, \frac{117}{2}, \frac{117}{2}, 60, 59, \frac{119}{2}, \frac{119}{2}, 0.\right\},\right.$$
$$\left\{60, 59, \frac{119}{2}, \frac{119}{2}, 59, 58, \frac{117}{2}, \frac{117}{2}, 0.\right\},$$
$$\left\{58, 57, \frac{115}{2}, \frac{115}{2}, 61, 60, \frac{121}{2}, \frac{121}{2}, 3153.53\right\},$$
$$\left\{61, 60, \frac{121}{2}, \frac{121}{2}, 58, 57, \frac{115}{2}, \frac{115}{2}, 3153.53\right\},$$
$$\left\{58, 57, \frac{115}{2}, \frac{115}{2}, 62, 60, \frac{121}{2}, \frac{121}{2}, 25\,136.6\right\},$$
$$\left\{62, 60, \frac{121}{2}, \frac{121}{2}, 58, 57, \frac{115}{2}, \frac{115}{2}, 25\,136.6\right\},$$
$$\left\{59, 57, \frac{115}{2}, \frac{115}{2}, 61, 60, \frac{121}{2}, \frac{121}{2}, 29\,716.4\right\},$$
$$\left\{61, 59, \frac{119}{2}, \frac{119}{2}, 59, 58, \frac{117}{2}, \frac{117}{2}, 29\,716.4\right\},$$
$$\left.\left\{61, 60, \frac{121}{2}, \frac{121}{2}, 59, 57, \frac{115}{2}, \frac{115}{2}, 29\,716.4\right\}\right\}$$

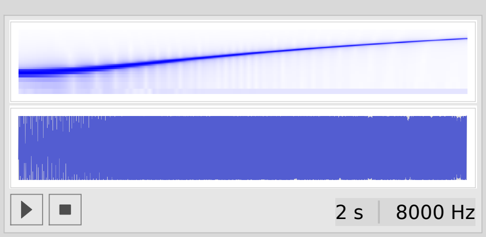
```
(*Fill up Matrix VMatrix with Sandwiched VdW Interaction <A,
B|VvdW(R)|A',B'>=RInt[R,{A,B},{A',B'}]*AInt[{A,B},{A',B'}] *)


Clear[AI]
VMatrix = Array[V, {Length[NList], Length[NList]}];
MatrixForm[VMatrix];
Start = AbsoluteTime[];
Dynamic[Refresh[Round[AbsoluteTime[] - Start], UpdateInterval → 1]]
Timing[For[i = 1, i ≤ Length[NList], i++,
          For[k = 1, k ≤ i, k++,
   If[NList[[i, 11]] ≠ NList[[k, 11]], VMatrix[[i, k]] = 0, (*Parity*)
    (*If[NList[[i,10]]≠NList[[k,10]],
     VMatrix[[i,k]]=0,(*Sum of m1 and m2*)*)

    VMatrix[[i, k]] =
     RInt[R, {NList[[i, 1]], NList[[i, 2]], NList[[i, 3]], NList[[i, 5]],
        NList[[i, 6]], NList[[i, 7]]}, {NList[[k, 1]], NList[[k, 2]],
        NList[[k, 3]], NList[[k, 5]], NList[[k, 6]], NList[[k, 7]]}] *
      AInt[{NList[[i, 2]], NList[[i, 3]], NList[[i, 4]], NList[[i, 6]],
        NList[[i, 7]], NList[[i, 8]]}, {NList[[k, 2]], NList[[k, 3]],
        NList[[k, 4]], NList[[k, 6]], NList[[k, 7]], NList[[k, 8]]}, θ];
    VMatrix[[k, i]] = Conjugate[VMatrix[[i, k]]]
   ] (*If m1 and m2*)
  ] (*If parity*)
 ]
]
Clear[i, k, AI]
MatrixForm[VMatrix] ;
MatrixPlot[VMatrix,
  ColorFunction → (GrayLevel[1 - #] &), ColorFunctionScaling → True];
MatrixForm[VMatrix];
MatrixPlot[VMatrix]

(*Absolute Energies of undisturbed levels, including Zeeman effect *)
Clear[i, f, R]
EI = {};
For[i = 1, i ≤ Length[NList], i++, EI = Join[EI, {NList[[i, 12]]} ] ]
EI = DiagonalMatrix[EI];
MatrixPlot[EI]
Play[Sin[1000 t (1 + t^2)], {t, 0, 2}]
EmitSound[%]
```

```
Round[3.633939456299122 × 10^9 - Start]
```

```
{0.668000, Null}
```

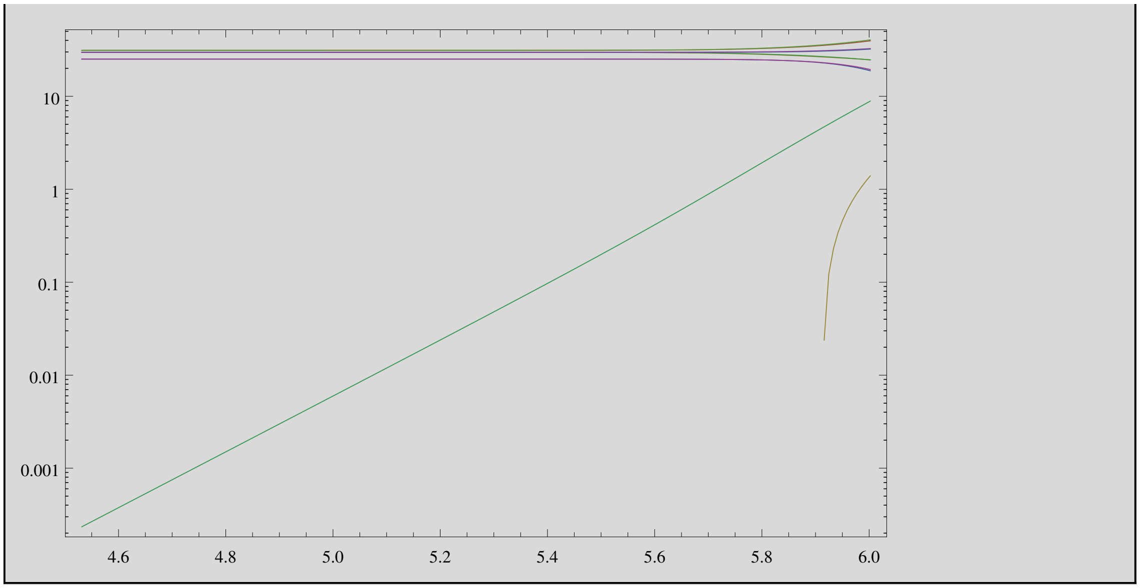

1
5
12
1
5
12


1
5
12
1
5
12

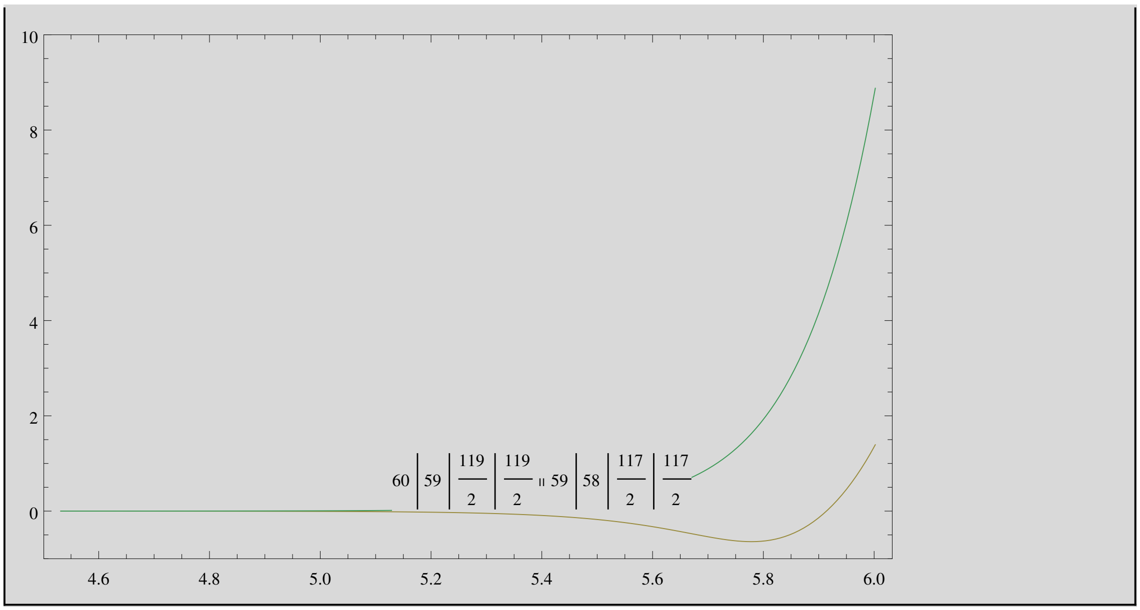

2 s
8000 Hz

```
(*MultiListPlot:
 Generate Plottable List MPList and Label (LabelList)*)

Clear[i, k, L, MPList, R]
MPList = Array[L, {Length[NList], 2}];
MPList;
Temp = {};

(*Include the R^-3 terms and caculate the
 energy for diferent interatomic distances *)
RMAX = 3.0 * 10^-5.0; (* μm*)
RMIN = 1.0 * 10^-6.;
R = RMAX;
RStep = 1.02;
RLabel = 1.0 * 10^-5.4;
For[i = 1, i ≤ Length[NList], i++, MPList[[i]] = {}];
Clear[i]
(*Eigenvalues[EI/10^9+VMatrix*1/R^3*a0^2*eSI^2/(4. π ϵ0)/h/10^9]*)
```

```
While[R > RMIN, R = R / RStep; Temp =
  Eigenvalues[EI / 10^9 + VMatrix * 1/R^3 * a0^2 * eSI^2/(4. π ϵ0) / h / 10^9] - EZeeman12 / 10^9;
 For[i = 1, i ≤ Length[NList], i++,
  MPList[[i]] = Join[MPList[[i]], {{Log[10, 1 / R], Temp[[i]]}}] ]]

(* Energy in [GHz] *)
(* Energy in [GHz] *)
(*Temp=Eigenvalues[EI/10^9+VMatrix*1/RLabel^3*a0^2*eSI^2/(4. π ϵ0)/h/10^9]-E12/10^9;*)
(*Generate Labels for Levels*)
TempList = Array[T, Length[NList]];
For[i = 1, i ≤ Length[NList], i++, TempList[[i]] = {Log[10, 1 / RLabel],
   (Temp[[Length[NList]]] + EZeeman12 / 10^9 - EI[[i, i]] / 10^9) *
     RMIN^3 / RLabel^3 - EZeeman12 / 10^9 + EI[[i, i]] / 10^9, LabelList[[i]]} ]

Show[ListLogPlot[MPList, Joined → True,
  PlotRange → (*{(*{Log[10,1/RMIN],Log[10,1/RMAX]}*)All,{-4,4.}}*)All,
  Frame → True]]

test = Show[ListPlot[MPList, Joined → True,
   PlotRange → (*{Log[10,1/RMIN],Log[10,1/RMAX]}*){All, {-1, 10}},
   Frame → True], Graphics[
   {Inset[TempList[[#, 3]], {TempList[[#, 1]], TempList[[#, 2]]}]} & /@
    Range@Length@TempList]]
```



10
1
0.1
0.01
0.001
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0



10
8
6
4
2
0
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0

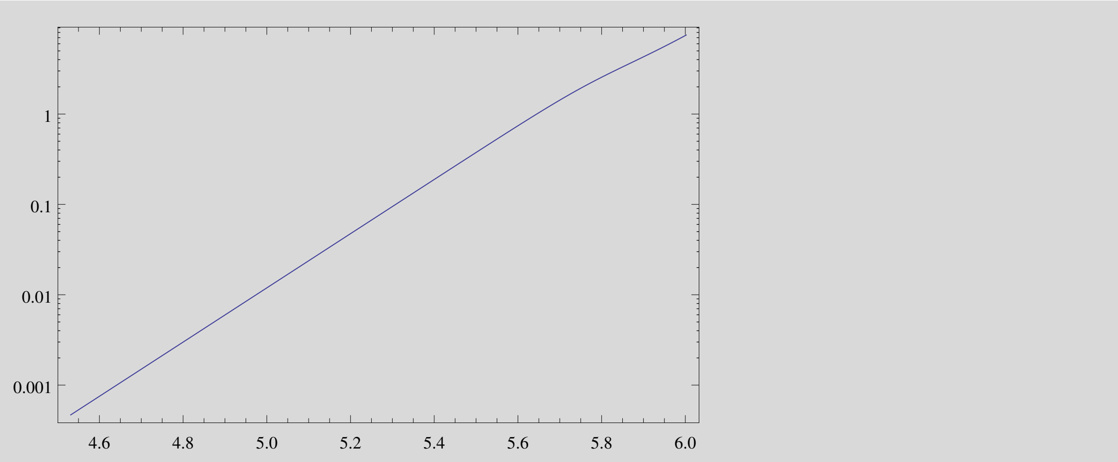
```
Clear[i];
LabelListTemp = {} ;
For [i = 1, i ≤ Length[LabelList], i++,
 If[(NList[[i, Range[4]]] == {n1, l1, j1, m1} ∧
     NList[[i, Range[5, 8]]] == {n2, l2, j2, m2}) ∨
   (NList[[i, Range[4]]] == {n2, l2, j2, m2} ∧
     NList[[i, Range[5, 8]]] == {n1, l1, j1, m1}),
  LabelListTemp = Join [LabelListTemp, {{i, NList[[i]]}}]
 ]
]
LabelListTemp
```

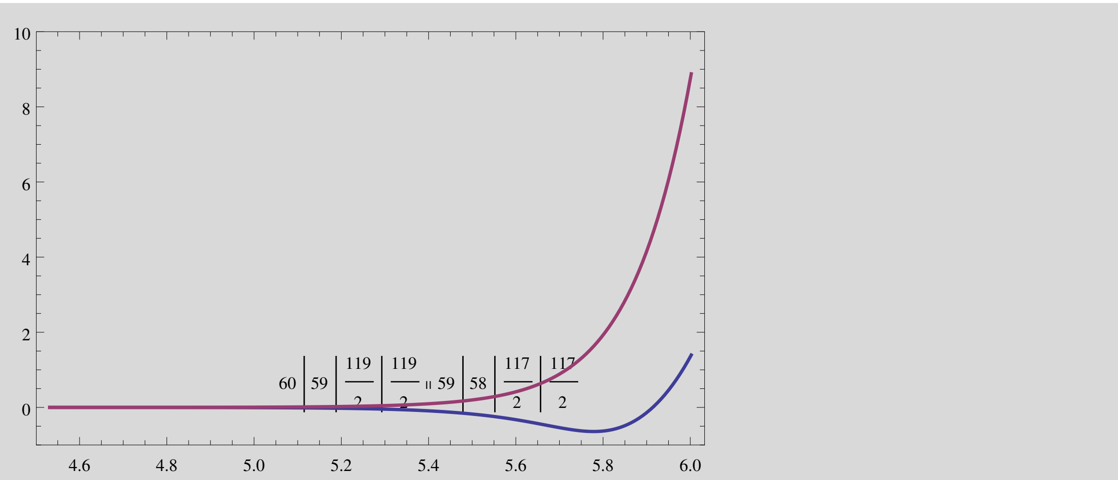
$\left\{\left\{3, \left\{59, 58, \frac{117}{2}, \frac{117}{2}, 60, 59, \frac{119}{2}, \frac{119}{2}, -1.85892\times10^{12}, 118, -1, -1.85747\times10^{12}\right\}\right\}, \left\{4, \left\{60, 59, \frac{119}{2}, \frac{119}{2}, 59, 58, \frac{117}{2}, \frac{117}{2}, -1.85892\times10^{12}, 118, -1, -1.85747\times10^{12}\right\}\right\}\right\}$

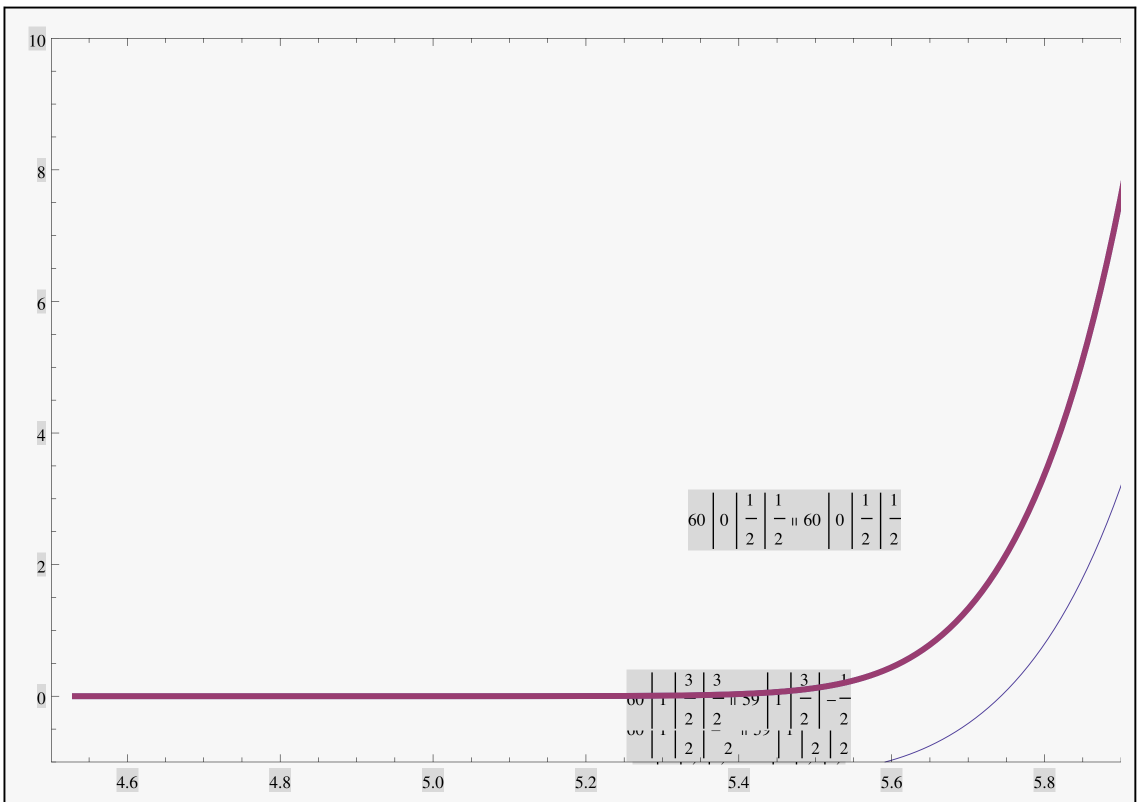
```
CentralLevel1 = 3;
CentralLevel2 = 4;
LabelList[[CentralLevel1]]
LabelList[[CentralLevel2]]
Show[ListLogPlot[Transpose[{MPList[[CentralLevel1, All, 1]],
    MPList[[CentralLevel2, All, 2]] - MPList[[CentralLevel1, All, 2]]}],
  Joined → True, PlotRange → (*{All,{-.0,.01}}*)All, Frame → True]]
Show[test, ListPlot[{MPList[[CentralLevel1]], MPList[[CentralLevel2]]},
  Joined → True, PlotRange → All(*All*),
  PlotStyle → Thickness[0.005], Frame → True]]
```

$59 \,|\, 58 \,|\, \frac{117}{2} \,|\, \frac{117}{2} \;\shortparallel\; 60 \,|\, 59 \,|\, \frac{119}{2} \,|\, \frac{119}{2}$

$60 \,|\, 59 \,|\, \frac{119}{2} \,|\, \frac{119}{2} \;\shortparallel\; 59 \,|\, 58 \,|\, \frac{117}{2} \,|\, \frac{117}{2}$


1
0.1
0.01
0.001
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0


10
8
6
4
2
0
60 | 59 | 119/2 | 119/2 ‖ 59 | 58 | 117/2 | 117/2
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0


10
8
6
4
2
0
60 | 0 | 1/2 | 1/2 ‖ 60 | 0 | 1/2 | 1/2
4.6
4.8
5.0
5.2
5.4
5.6
5.8

```
Length[MPList[[CentralLevel1]] ]
EnergyOutput = Array[L, {Length[MPList[[CentralLevel1]]], 4}] ;
(*EnergyOutput= MPList[[CentralLevel1 ]]  ;
Energy60s60pTest2 = {};
Energy60s60pTest2 = MPList[[CentralLevel2]];
Length[EnergyOutput]
 Length[Energy60s60pTest2]
 For [i = 1, i ≤ Length [EnergyOutput], i++ ,
  EnergyOutput [[i,1]] = 10 ^(-EnergyOutput [[i,1]] ) ;
   EnergyOutput [[i,3]] = -Energy60s60pTest2 [[i,2]]]
 Clear[i];
For [i = 1, i ≤ Length [Energy60s60pTest2], i++ ,
 Energy60s60pTest2 [[i,1]] = 10 ^(-Energy60s60pTest2 [[i,1]] ) ;   ]
*)
 For [i = 1, i ≤ Length[MPList[[CentralLevel1]]], i++,
 EnergyOutput[[i, 1]] = 10^(-MPList[[CentralLevel1]][[i, 1]]);
 EnergyOutput[[i, 2]] = MPList[[CentralLevel1]][[i, 2]];
 EnergyOutput[[i, 3]] = MPList[[CentralLevel2]][[i, 2]];
 EnergyOutput[[i, 4]] =
  MPList[[CentralLevel2]][[i, 2]] - MPList[[CentralLevel1]][[i, 2]];
]
EnergyOutput;
Export [
 "/media/2kome/DATA/Works/Lab/Simulation/Circulation/60C59C_00deg_2nd
   order.dat", EnergyOutput, "Table"]
(*Export ["S:\\Simulations et calculs\\Interation\\2nd
   order\\02-05-2014\\60S59S_00deg_2nd order.dat",
 EnergyOutput,"Table"]  *)
```

```
172
```

```
/media/2kome/DATA/Works/Lab/Simulation/Circulation/60C59C_00deg_2nd
  order.dat
```

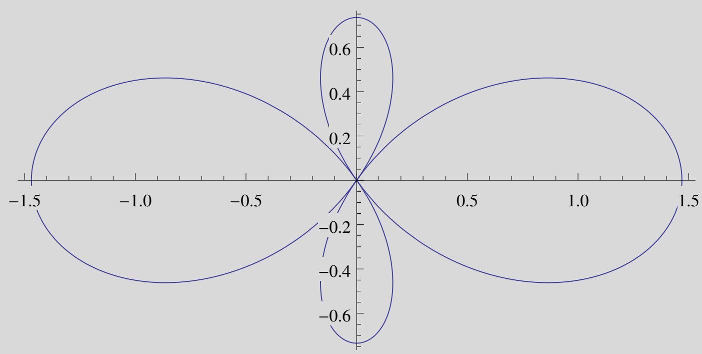
```
"D:\\Documents\\raulcteixeira\\Documents\\Simulations et
  calculs\\Interation\\2nd order\\60S60P3-2 1-2_00deg_2nd order.dat"
```

```
nx = 60; lx = 0; sx = 1 / 2; jx = 1 / 2; mx = 1 / 2;
(*Atom2, |n2,l2,s2,j2,m2>*)
ny = 60; ly = 1; sy = 1 / 2; jy = 3 / 2; my = -1 / 2;
```

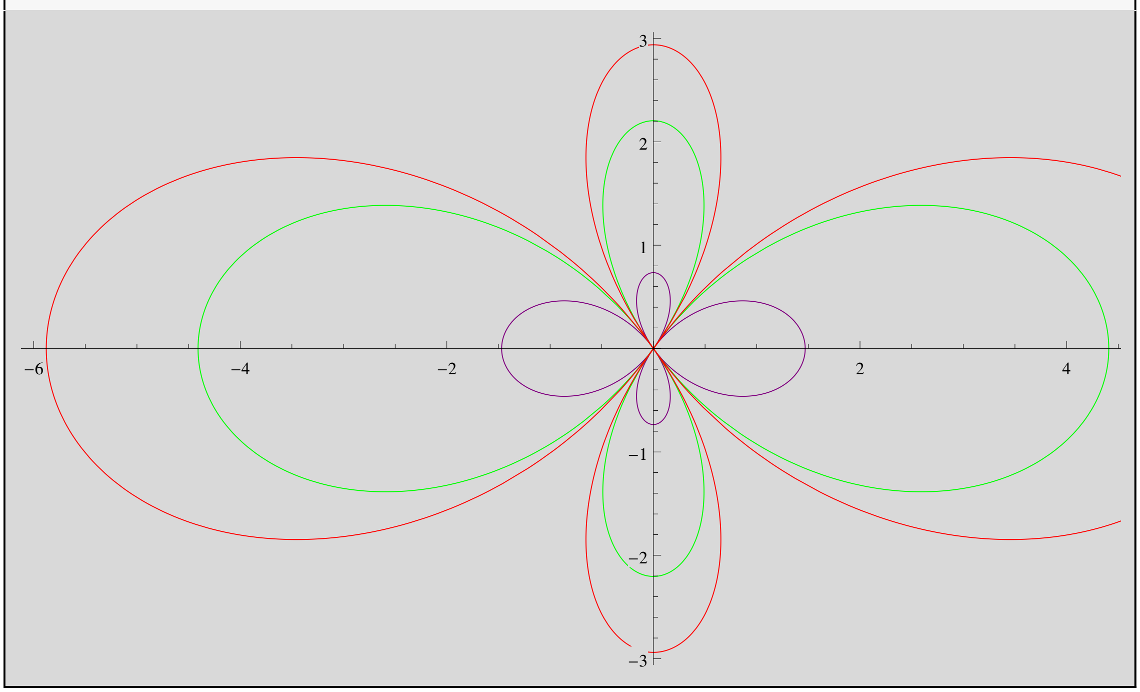
$$\text{PolarPlot}\left[\frac{1}{\text{Rtest}^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9 \right.$$

$$\text{RInt}[R, \{n_y, l_y, j_y, n_x, l_x, j_x\}, \{n_x, l_x, j_x, n_y, l_y, j_y\}] *$$

$$\left.\text{AInt}[\{l_y, j_y, m_y, l_x, j_x, m_x\}, \{l_x, j_x, m_x, l_y, j_y, m_y\}, \gamma], \{\gamma, 0, 2\,\text{Pi}\}\right]$$



```
nx = 60; lx = 0; sx = 1 / 2; jx = 1 / 2; mx = 1 / 2;
(*Atom2, |n2,l2,s2,j2,m2>*)
ny = 60; ly = 1; sy = 1 / 2; jy = 3 / 2; my = 3 / 2;

PolarPlot[{1/Rtest^3 * a0^2 * esI^2/(4. π ϵ0) / h / 10^9
    RInt[R, {ny, ly, jy, nx, lx, jx}, {nx, lx, jx, ny, ly, jy}] *
    AInt[{ly, jy, -1 / 2, lx, jx, mx}, {lx, jx, mx, ly, jy, -1 / 2}, γ],
  1/Rtest^3 * a0^2 * esI^2/(4. π ϵ0) / h / 10^9
    RInt[R, {ny, ly, jy, nx, lx, jx}, {nx, lx, jx, ny, ly, jy}] *
    AInt[{ly, jy, 3 / 2, lx, jx, mx}, {lx, jx, mx, ly, jy, 3 / 2}, γ], 1/Rtest^3 * a0^2 *
    esI^2/(4. π ϵ0) / h / 10^9 RInt[R, {ny, ly, jy, nx, lx, jx}, {nx, lx, jx, ny, ly, jy}] *
    AInt[{ly, jy, 1 / 2, lx, jx, mx}, {lx, jx, mx, ly, jy, 1 / 2}, γ]},
 {γ, 0, 2 Pi}, PlotStyle → {Purple, Green, Red}]
```

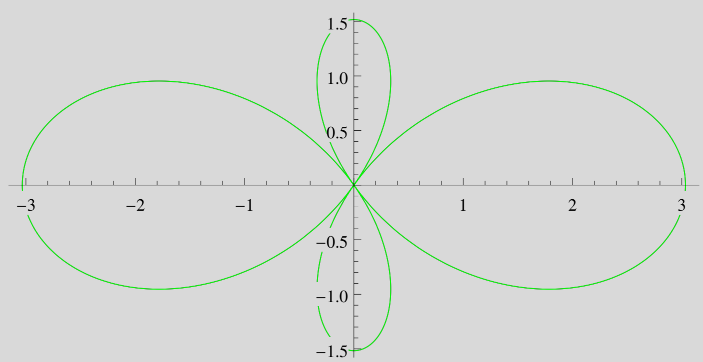


$n_x = 60;\ l_x = 0;\ s_x = 1/2;\ j_x = 1/2;\ m_x = 1/2;$

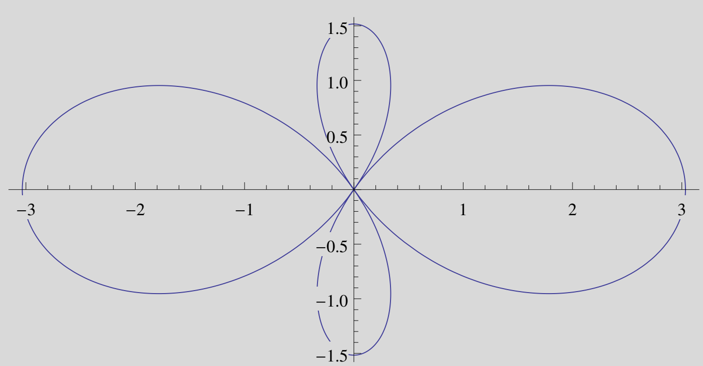
(*Atom2, |n_2, l_2, s_2, j_2, m_2>*)

$n_y = 60;\ l_y = 1;\ s_y = 1/2;\ j_y = 1/2;\ m_y = 1/2;$

$$\text{PolarPlot}\Big[\Big\{\frac{1}{\text{Rtest}^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\ \pi\, \epsilon_0} \Big/ h \Big/ 10^9$$
$$\text{RInt}[R, \{n_y, l_y, j_y, n_x, l_x, j_x\}, \{n_x, l_x, j_x, n_y, l_y, j_y\}] *$$
$$\text{AInt}[\{l_y, j_y, -1/2, l_x, j_x, m_x\}, \{l_x, j_x, m_x, l_y, j_y, -1/2\}, \gamma],$$
$$\frac{1}{\text{Rtest}^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\ \pi\, \epsilon_0} \Big/ h \Big/ 10^9$$
$$\text{RInt}[R, \{n_y, l_y, j_y, n_x, l_x, j_x\}, \{n_x, l_x, j_x, n_y, l_y, j_y\}] *$$
$$\text{AInt}[\{l_y, j_y, 1/2, l_x, j_x, m_x\}, \{l_x, j_x, m_x, l_y, j_y, 1/2\}, \gamma]\Big\},$$
$$\{\gamma, 0, 2\,\text{Pi}\}, \text{PlotStyle} \rightarrow \{\text{Purple}, \text{Green}\}\Big]$$



$$\text{PolarPlot}\Big[\frac{1}{\text{Rtest}^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\ \pi\, \epsilon_0} \Big/ h \Big/ 10^9$$
$$\text{RInt}[R, \{n_y, l_y, j_y, n_x, l_x, j_x\}, \{n_x, l_x, j_x, n_y, l_y, j_y\}] *$$
$$\text{AInt}[\{l_y, j_y, -1/2, l_x, j_x, m_x\}, \{l_x, j_x, m_x, l_y, j_y, -1/2\}, \gamma], \{\gamma, 0, 2\,\text{Pi}\}\Big]$$



```
Eigenvalues[EI][[168]]
Eigenvalues[EI][[169]]
Eigenvectors[EI][[168]]
Eigenvectors[EI][[169]]
```

-2.0712×10^{12}

-2.0712×10^{12}

```
{0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

```
{0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

$R = 4.54 \times 10^{-6}$

$\text{Eigenvalues}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right][[168]] - \text{EZeeman}_{12} / 10^9$

$\text{Eigenvectors}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right][[168, 168]]$

$\text{Eigenvectors}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right][[168, 169]]$

$\text{Eigenvectors}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right][[169, 168]]$

$\text{Eigenvectors}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right][[169, 169]] \Big/ \text{Norm}\left[\text{Eigenvectors}\left[EI / 10^9 + \text{VMatrix} * \frac{1}{R^3} * a_0{}^2 * \frac{e_{SI}{}^2}{4.\,\pi\,\epsilon_0} \Big/ h \Big/ 10^9\right]\right]$

MPList[[168, 2]]

4.54×10^{-6}

0.00405204

0.70651

-0.70651

0.674848

0.674848

$\{4.54008,\ 6.17429 \times 10^{-8}\}$