In[ ]:=

```
(* Radiative Quenching, Chapter 3 *)
(* This one uses my paper *)

(*electron mass *)
me = 1;
(* Proton Mass *)
mp = 1836;
(* Reduced Mass *)
μ = mp/2;
hbar = 1;
hbarSI = 1.054571817*^-34;
cLight = 137.036 ;(*SpeedOfLight, atomic units*)
cLightSI = 299 792 458 ;(*SpeedOfLight, m/s*)
oneSec = 2.418884326509*^-17; (* 1 sec SI ↔ 1 sec AU *)
aBorh = 5.29177*^-11;  (* Si ↔ AU *)

(* RawData format R,E, E + 1/R,A,p *)

vSg1RawData =
  Import["/Users/zelimir/Work/Physics-Thesis/thesis-2/ZelimirOverleaf/
     MathematicaCodeFinal/gerade1sV3.mx"];
vSu2RawData =
  Import["/Users/zelimir/Work/Physics-Thesis/thesis-2/ZelimirOverleaf/
     MathematicaCodeFinal/ungerade1sV3.mx"];


vSg1Data = vSg1RawData⟦1 ;; 44⟧;
vSu2Data = vSu2RawData⟦1 ;; 44⟧;

(* Calculate the Transition Dipole Moment *)
(* Upper Limit for integration *)
maxR = vSg1Data⟦Length[vSg1Data]⟧⟦1⟧;
Print["maxR=", maxR]
maxR = 20;
maxIndex = 40;

Print["R=", vSg1Data⟦maxIndex, 1⟧, ", A=",
  vSg1Data⟦maxIndex, 4⟧, ", p=", vSg1Data⟦maxIndex, 5⟧];
Print["R=", vSu2Data⟦maxIndex, 1⟧, ", A=",
  vSu2Data⟦maxIndex, 4⟧, ", p=", vSu2Data⟦maxIndex, 5⟧];
```

In[ ]:=

```
(* Load Potential curves for 1sg, 2sg states
"R,E, E + 1/R,A,p *)
(* 1sg state, potential curve *)
vG = Interpolation[
   Transpose[{vSg1Data[[All, 1]], vSg1Data[[All, 3]]}], InterpolationOrder → 3];
(* 2sg state, potential curve *)
vU = Interpolation[
   Transpose[{vSu2Data[[All, 1]], vSu2Data[[All, 3]]}], InterpolationOrder → 3];

deltaV[r_] := Abs[vG[r] - vU[r]];
deltaV[20]
Plot[{vG[r], vU[r]}, {r, 0.2, maxR}, PlotRange → Full, PlotLegends → {"VG", "VU"},
 PlotLabel → "Potential curver for the Gerade and Ungerade state"]
```

In[ ]:=

```
(* Compute the Dipole momement D(r) for the single value of R *)
singleDR[R_, a1_, p1_, a2_, p2_, maxDistance_] :=
  Module[{norm1, norm2, dInt, l1, l2, m1, m2, lmNorm1, lmNorm2},
   (* 1s wavefunction *)
   l1 = NDSolveValue[{(ξ^2 - 1) L''[ξ] + ξ * L'[ξ] + (a1 + 2 R ξ - p1^2 ξ^2) L[ξ] == 0 ,
      L[1.001] == 0, L'[1.001] == 1 }, L, {ξ, 1.001, maxDistance}];
   m1 = NDSolveValue[{(1 - η^2) M''[η] - η * M'[η] - (-a1 + p1^2 η^2) M[η] == 0,
      M[0.999] == 0, M'[0.999] == 1}, M, {η, -.999, .999}];

   (* 2s wavefunction *)
   l2 = NDSolveValue[{(ξ^2 - 1) L''[ξ] + ξ * L'[ξ] + (a2 + 2 R ξ - p2^2 ξ^2) L[ξ] == 0 ,
      L[1.001] == 0, L'[1.001] == 1 }, L, {ξ, 1.001, maxDistance}];
   m2 = NDSolveValue[{(1 - η^2) M''[η] - η * M'[η] - (-a2 + p2^2 η^2) M[η] == 0,
      M[0.999] == 0, M'[0.999] == 1}, M, {η, -.999, .999}];

   Print["R=", R];
   (* Compute <ψ|ψ> = <ψ1s|ψ1s> + <ψ2s|ψ2s> *)
   (* <ψ1s|ψ1s> *)
   norm1 = Sqrt[NIntegrate[R^2/4 Abs[l1[ξ]]^2 Abs[ m1[η]]^2 (ξ^2 - η^2)/(Sqrt[ξ^2 - 1] Sqrt[1 - η^2]),
      {η, -.999, .999}, {ξ, 1.001, maxDistance}]];
   (* <ψ2s|ψ2s> *)
```

```
    norm2 = Sqrt[NIntegrate[R^2/4 Abs[l2[ξ]]^2 Abs[m2[η]]^2 (ξ^2 - η^2)/(Sqrt[ξ^2 - 1] Sqrt[1 - η^2]),
        {η, -.999, .999}, {ξ, 1.001, maxDistance}]];

    dInt = (R/2)^3 1/(norm1 norm2) NIntegrate[l2[ξ] × l1[ξ] × m1[η] × m2[η] (ξ η (ξ^2 - η^2))/(Sqrt[ξ^2 - 1] Sqrt[1 - η^2]),
        {η, -0.999, 0.999}, {ξ, 1.001, maxDistance}];
    {R, dInt }
   ];

singleDR[vSg1Data[[24]][[1]], vSg1Data[[19]][[4]],
 vSg1Data[[19]][[5]], vSu2Data[[19]][[4]], vSu2Data[[19]][[5]], 5]
(*singleDR[vSg1Data[[14]][[1]],vSg1Data[[19]][[4]],
  vSg1Data[[19]][[5]],vSu2Data[[19]][[4]],vSu2Data[[19]][[5]],5]
 singleDR[vSg1Data[[15]][[1]],vSg1Data[[20]][[4]],
  vSg1Data[[20]][[5]],vSu2Data[[20]][[4]],vSu2Data[[20]][[5]],5]*)
```

In[∘]:=

```
(* Get all Rs, and coefficients A and p and calculate D(R) for all Rs *)
(* RawData format R,E, E + 1/R,A,p *)
inputData = Transpose[
   {vSg1Data[[1 ;; maxIndex, 1]], vSg1Data[[1 ;; maxIndex, 4]], vSg1Data[[1 ;; maxIndex, 5]],
    vSu2Data[[1 ;; maxIndex, 4]], vSu2Data[[1 ;; maxIndex, 5]]}];

allDR = Map[singleDR[#[[1]], #[[2]], #[[3]], #[[4]], #[[5]], maxR] &, inputData];
```

In[◦]:=
```
(* Dipole moment in SI units *)
drAUtoSI = 8.478*^-30; (* Cm *)
allDRSI = allDR;
allDRSI[[All, 2]] = allDR[[All, 2]] * drAUtoSI;
allDRSI

dR =
  Interpolation[Transpose[{allDR[[All, 1]], allDR[[All, 2]]}], InterpolationOrder → 3];
Plot[dR[r], {r, 0.1, 7}, PlotLabel → "D(R)",
 AxesLabel → {"R(a0)", "D(R)(in a.u.)"}, PlotRange → Full]

dRSI = Interpolation[
   Transpose[{allDRSI[[All, 1]], allDRSI[[All, 2]]}], InterpolationOrder → 3];
Plot[dRSI[r], {r, 0.1, maxR}, PlotLabel → "D(R)",
 AxesLabel → {"R(a0)", "D(R)  (units of C m)"}, PlotRange → Full, Axes → True]
```

In[◦]:=

$$\text{aR[R\_] := } \frac{4}{3\,\text{hbar cLight}^3}\,(\text{dR[R]})^2\left(\frac{\text{vU[R] - vG[R]}}{\text{hbar}}\right)^3;$$

```
aR[1]

aRAUtoSI = 4.1341*^16;

(* Einstein A coefficient in SI  and a.u. units *)
```

$$\text{aRSI2[R\_] := } \frac{4}{3\,\text{hbarSI cLightSI}^3}\,(\text{dRSI[R]})^2\left(\frac{(\text{vU[R] - vG[R]})\ 4.35974*\text{\^{}-18}}{\text{hbarSI}}\right)^3;$$

```
Plot[aRSI2[R], {R, 0.1, 1}, AxesLabel → {"R (a.u)", "A(R)s^-1"},
 PlotLabel → "Einstein's A coefficient for Σg -> Σu transition"]

Plot[aRSI2[R] * 10^6, {R, 0.1, 1}, AxesLabel → {"R (a.u.)", "A(R)10^6s^-1"},
 PlotLabel → {"Einstein's A coefficient for g -> u transition"}]
```