

# Signal for the Speckle Experiment

```
In[1]:= (* Force the notebook to autosave after running each cell *)
SetOptions[$FrontEndSession, NotebookAutoSave -> True]
NotebookSave[]
```

## Retrieving Balmer- $\alpha$ Lines

Mathematica has the really cool feature of allowing one to look up the atomic transition data for any element. No copy/pasting from the NIST atomic spectrum database required!

```
In[3]:= (* Retrieve atomic transitions present in the Balmer  $\alpha$  line *)
balmer $\alpha$  = SpectraLineData[EntityClass["AtomicLine", {"Hydrogen", 1}],
  {Quantity[600, "Nanometers"], Quantity[700, "Nanometers"]}] [[1 ;; 4]]
```

```
Out[3]= { H: (3d1)2D3/2 -> (2p1)2P1/2 , H: (3p1)2P1/2 -> (2s1)2S1/2 ,
  H: (3p1)2P3/2 -> (2s1)2S1/2 , H: (3d1)2D5/2 -> (2p1)2P3/2 }
```

## Spectrum with No Inhomogenous Broadening

The line profiles will have a Lorentzian shape when considering only natural broadening. From Loudon eq. 2.5.20, this is given by

$$F_{\text{nat}}(\omega) = \frac{\gamma_{\text{sp}}/\pi}{(\omega_0 - \omega)^2 + \gamma_{\text{sp}}^2} \quad (1)$$

where  $2\gamma_{\text{sp}} = A_{ji} = \frac{1}{\tau_R}$  with  $\gamma_{\text{sp}} \equiv$  the radiative decay width,  $A_{ji} \equiv$  the Einstein A coefficient, and  $\tau_R \equiv$  the radiative decay lifetime. To determine the radiant power of emission of each line, we simply multiply the transition probability ( $A_{ji}$ ) by the number of atoms in state  $j = n_j V$  ( $n_j$  being the number density of atoms in state  $j$ , and  $V$  being the volume of the gas) and the energy of the emitted photon  $\hbar\omega_{ji}$ :

$$\Phi_E = A_{ji} \hbar\omega_{ji} n_j V \quad (2)$$

Note that this is only valid for a diffuse gas where reabsorption is negligible. Mathematica can give us the Einstein A coefficients and the frequencies for each transition:

```

In[4]:= (* Get the frequency of photons emitted from each transition *)
lineFreq = SpectralLineData[balmerα, "Frequency"]
lineFreqTHz = QuantityMagnitude[lineFreq];
(* Convert to angular frequency for consistency with Loudon *)
lineω = 2 * π * lineFreq
lineωTHz = QuantityMagnitude[lineω];
(* Get transition probabilities *)
lineProb = SpectralLineData[balmerα, "TransitionProbability"]
lineProbTHz = QuantityMagnitude[UnitConvert[lineProb, "Terahertz"]];

```

```

Out[4]= { 456.812 THz , 456.808 THz , 456.811 THz , 456.802 THz }

```

```

Out[6]= { 2870.23 THz , 2870.21 THz , 2870.23 THz , 2870.17 THz }

```

```

Out[8]= { 5.3877 × 107 per second , 2.2449 × 107 per second ,
         2.2448 × 107 per second , 6.4651 × 107 per second }

```

What we don't know yet is the number of atoms in each state. If the system is in thermal equilibrium, this can be acquired through the Boltzmann distribution such that

$$n_j(T) = \frac{n_t g_j e^{-E_j / (k_B T)}}{Z(T)} \quad (3)$$

where  $n_j \equiv$  the number density of atoms in state  $j$ ,  $n_t \equiv$  the total number density of atoms,  $g_j \equiv$  the statistical weight of state  $j = 2J+1$ ,  $k_B \equiv$  Boltzmann's constant,  $T \equiv$  temperature, and  $Z(T)$  is the partition function for the system. Hence, the scaled lineshape can be written as

$$f_{\text{nat},ji}(\omega, T) = \Phi_{E,ji}(T) F_{\text{nat},ji}(\omega) = \frac{A_{ji} \hbar \omega_{ji} n_t g_j e^{-E_j / (k_B T)}}{Z(T)} \frac{\gamma_{sp} / \pi}{(\omega_{ji} - \omega)^2 + \gamma_{sp}^2} \quad (4)$$

and the unbroadened shape of the Balmer- $\alpha$  line with  $K$  transitions can be given by

$$f_{\text{nat},H\alpha}(\omega, T) = \sum_{m=1}^M \Phi_{E,ji,m}(T) F_{\text{nat},ji,m}(\omega) = \frac{\hbar n_t}{Z(T)} \sum_{m=1}^M A_{ji,m} \omega_{ji,m} g_{j,m} e^{-E_{j,m} / (k_B T)} \frac{\gamma_{sp,m} / \pi}{(\omega_{ji,m} - \omega)^2 + \gamma_{sp,m}^2} \quad (5)$$

Since  $\int_{-\infty}^{\infty} F_L(\omega) d\omega = 1$ , (where  $L$  signifies a Lorentzian lineshape) we can normalize equation (5) to get

$$F_{\text{nat},H\alpha}(\omega, T) = \frac{1}{R(T)} \sum_{m=1}^M g_{j,m} \omega_{ji,m} A_{ji,m} e^{-E_{j,m} / (k_B T)} \frac{\gamma_{sp,m} / \pi}{(\omega_{ji,m} - \omega)^2 + \gamma_{sp,m}^2} \quad (6)$$

where

$$R(T) = \sum_{m=1}^M g_{j,m} \omega_{ji,m} A_{ji,m} e^{-E_{j,m} / (k_B T)} \quad (7)$$

We can now plot equation (6) to see the lineshape for the unbroadened  $H\alpha$  line.

```
In[10]:= (* get initial and final states *)
lineInitialState = SpectralLineData[balmer $\alpha$ , "UpperLevel"]
lineFinalState = SpectralLineData[balmer $\alpha$ , "LowerLevel"]
```

```
Out[10]:= { H:  $(3d^1)^2D_{3/2}$  , H:  $(3p^1)^2P_{1/2}$  , H:  $(3p^1)^2P_{3/2}$  , H:  $(3d^1)^2D_{5/2}$  }
```

```
Out[11]:= { H:  $(2p^1)^2P_{1/2}$  , H:  $(2s^1)^2S_{1/2}$  , H:  $(2s^1)^2S_{1/2}$  , H:  $(2p^1)^2P_{3/2}$  }
```

```
In[12]:= (* get the weight of each initial state *)
initialStateJ = SpectralLineData[lineInitialState, "JValue"]
initialStateWeight = 2 * initialStateJ + 1
```

```
Out[12]:= {  $\frac{3}{2}$ ,  $\frac{1}{2}$ ,  $\frac{3}{2}$ ,  $\frac{5}{2}$  }
```

```
Out[13]:= { 4, 2, 4, 6 }
```

```
In[14]:= (* get the excitation energies of each initial state in eV *)
initialStateE = SpectralLineData[lineInitialState, "Energy"]
initialStateEeV = QuantityMagnitude[initialStateE];
```

```
Out[14]:= { 12.0875 eV , 12.0875 eV , 12.0875 eV , 12.0875 eV }
```

In[16]:=

```
(* function for the normalized unbroadened Balmer- $\alpha$  lineshape *)
kb = QuantityMagnitude[UnitConvert[k, "eV/Kelvins"]];
coeffH $\alpha$ [T_] :=
  lineProbTHz * line $\omega$ THz * initialStateWeight * Exp[-initialStateEeV / (kb * T)];
normR[T_] := Total[coeffH $\alpha$ [T]];
 $\gamma$ sp = lineProb / 2;
lorentzH $\alpha$ [ $\omega$ _] := ( $\gamma$ spTHz /  $\pi$ ) / ((line $\omega$ THz -  $\omega$ )^2 +  $\gamma$ spTHz^2);
nFH $\alpha$ [ $\omega$ _, T_] := Total[coeffH $\alpha$ [T] * lorentzH $\alpha$ [ $\omega$ ]] / normR[T]

(* "Units" version of calculation lets Mathematica handle unit
conversion. This is very slow, so I'm only using it as a double check. *)
coeffH $\alpha$ Units[T_] := lineProb * line $\omega$  *
  initialStateWeight * Exp[-initialStateE / (k * T)];
normRUnits[T_] := Total[coeffH $\alpha$ Units[T]];
 $\gamma$ spTHz = lineProbTHz / 2;
lorentzH $\alpha$ Units[ $\omega$ _] := ( $\gamma$ sp /  $\pi$ ) / ((line $\omega$  -  $\omega$ )^2 +  $\gamma$ sp^2);
nFH $\alpha$ Units[ $\omega$ _, T_] := UnitConvert[
  Total[coeffH $\alpha$ Units[T] * lorentzH $\alpha$ Units[ $\omega$ ]] / normRUnits[T], "Picoseconds"];
```

In[28]:=

```
(* Plot boundaries for frequency *)
(* number of half-widths beyond the centers of the minimum/maximum peaks
at which to set the plot boundaries (not used at the moment) *)
nHalfWidths = 10;

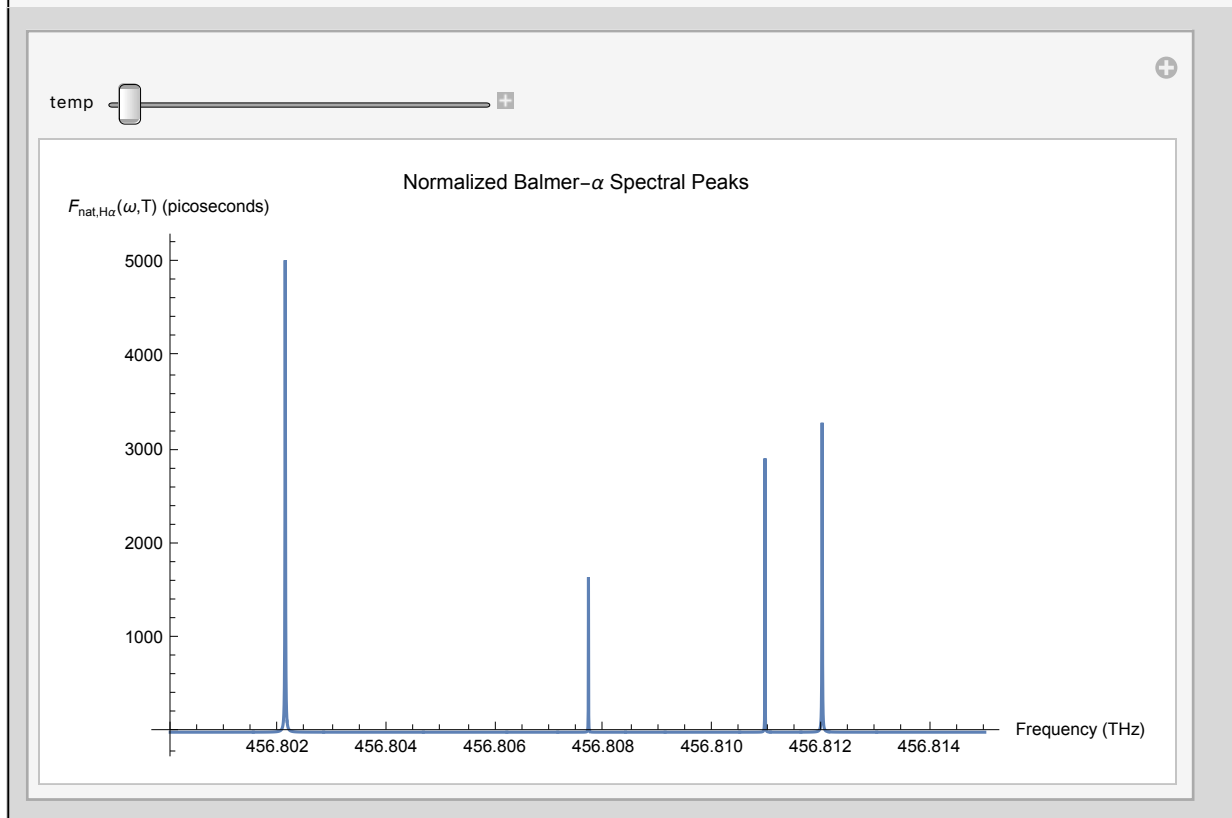
vUnit = "Terahertz";
vMin = 456.8;
vMax = 456.815;

(* Slider boundaries for temperature *)
tempUnit = "Kelvins";
tempMin = 273;
tempMax = 5000;
```

In[71]:=

```
Manipulate[
  Plot[
    nFH $\alpha$ [ $\nu * 2 \pi$ , temp],
    { $\nu$ ,  $\nu$ Min,  $\nu$ Max},
    PlotRange  $\rightarrow$  All,
    PlotLabel  $\rightarrow$  "Normalized Balmer- $\alpha$  Spectral Peaks",
    AxesLabel  $\rightarrow$  {"Frequency (THz)", " $F_{\text{nat}, H\alpha}(\omega, T)$  (picoseconds)"},
    ImageSize  $\rightarrow$  Full],
  {temp, tempMin, tempMax},
  ContinuousAction  $\rightarrow$  True
]
```

Out[71]=



In[72]:=

```
(* Uncomment to have Mathematica take
care of units for comparison. BEWARE: very slow *)

(*Manipulate[
Plot[
  nFHαUnits[Quantity[v*2π,vUnit],Quantity[temp,tempUnit]],
  {v,vMin,vMax},
  PlotRange→ All,
  PlotLabel→ "Normalized Balmer-α Spectral Peaks",
  AxesLabel→{"Frequency (THz)","Fnat,Hα(ω,T)"},
  ImageSize→ Full],
{temp,tempMin,tempMax},
ContinuousAction→True
]*)
```

## The Pure Speckle Signal

Recall that the normalized second order correlation function in the limit of  $N e^{-\sigma^2 \tau^2} \ll 1$  from our paper is

$$g^{(2)}(\tau) \approx 1 + \frac{1}{N} \left| \frac{\sum_{m=1}^M |\varepsilon_m|^2 e^{-i\Delta_m \tau}}{\sum_{m=1}^M |\varepsilon_m|^2} \right|^2 = 1 + \frac{Q(\tau)}{N} \quad (8)$$

where  $Q(\tau)$  is our quantity of interest. For consistency with equation (8), let's define our  $\Delta_m$ 's as the angular frequency difference of each Balmer- $\alpha$  sub-line from their average:

In[37]:=

```
lineΔTHz = lineωTHz - Mean[lineωTHz];
lineΔ = lineω - Mean[lineω]
```

Out[38]=

```
{ 0.0239463 THz , -0.00309728 THz , 0.01733 THz , -0.0381791 THz }
```

Note that the quantity  $|\varepsilon_m|^2 / \sum_{m=1}^M |\varepsilon_m|^2$  is just the relative intensity of each of the normalized peaks in the Balmer- $\alpha$  line. We can use the facts that the peaks in the unbroadened spectrum and that  $\int_{-\infty}^{\infty} F_L(\omega) d\omega = 1$  to find the relative intensity  $c_m(T)$  of each line to be

$$c_m(T) = \frac{g_{j,m} \omega_{ji,m} A_{ji,m} e^{-E_{j,m}/(k_B T)}}{R(T)}. \quad (9)$$

In[39]:=

```
(* Calculate the equation above. I could have done this earlier,
but I didn't... *)
relativeIntensity[T_] := coeffHα[T] / normR[T]
relativeIntensityUnits[T_] := coeffHαUnits[T] / normRUnits[T]
```

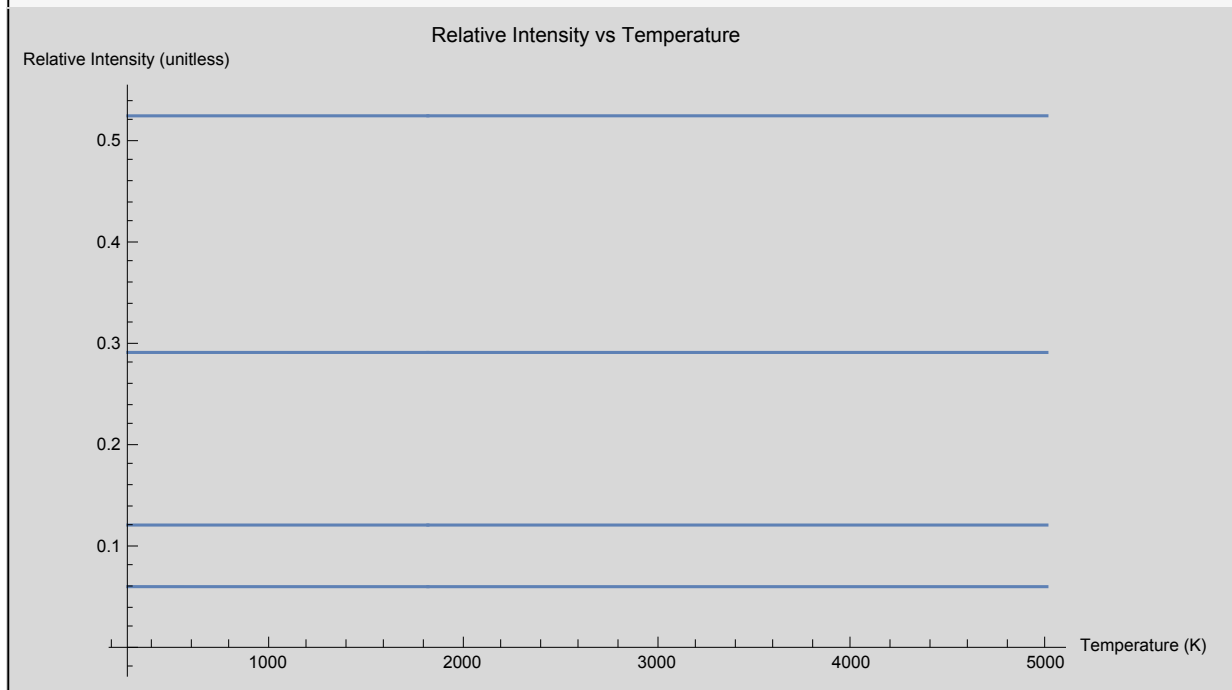
Note that, since the peaks inside the Balmer- $\alpha$  line are so close, their relative intensities are basically

constant:

In[80]:=

```
Plot[
  relativeIntensity[temp],
  {temp, 273, 5000},
  PlotLabel -> "Relative Intensity vs Temperature",
  AxesLabel -> {"Temperature (K)", "Relative Intensity (unitless)"},
  ImageSize -> Full
]
```

Out[80]=



Plugging equation (9) into  $Q(\tau)$ , we get

$$Q(\tau, T) = \left| \sum_{m=1}^M c_m(T) e^{-i\Delta_m \tau} \right|^2. \quad (10)$$

Upon expanding the square, the  $\tau$  dependent part of  $Q(\tau, T)$  resolves to a sum of cosines with period  $2\pi/|\Delta_m - \Delta_{m'}|$ . Calculating these periods, we get

In[41]:=

```
(* Sort the  $\Delta_m$ 's from smallest to largest first so it's
easier to associate the line differences with the spectrum *)
sortedLine $\Delta$  = Sort[line $\Delta$ ];
line $\Delta\Delta$  = UnitConvert[
  Reap[
    For[i = 1, i < Length[line $\Delta$ ], i++,
      For[j = i + 1, j  $\leq$  Length[line $\Delta$ ], j++,
        Sow[2  $\pi$  / Abs[sortedLine $\Delta$ [[i]] - sortedLine $\Delta$ [[j]]]]];
    ]
  ][[2]][[1]],
  "Picoseconds"
]
```

Out[42]=

```
{ 179.101 ps , 113.192 ps , 101.137 ps , 307.588 ps , 232.335 ps , 949.653 ps }
```

We can finally plot our signal to see what it looks like!

In[43]:=

```
sigQ[ $\tau$ _, T_] := Re[
  Total[relativeIntensity[T] * Exp[- $i$  * line $\Delta$ THz *  $\tau$ ]]
  * Conjugate[Total[relativeIntensity[T] * Exp[- $i$  * line $\Delta$ THz *  $\tau$ ]]]
]

sigQUnits[ $\tau$ _, T_] := Re[
  Total[relativeIntensityUnits[T] * Exp[- $i$  * line $\Delta$  *  $\tau$ ]]
  * Conjugate[Total[relativeIntensityUnits[T] * Exp[- $i$  * line $\Delta$  *  $\tau$ ]]]
]
```

In[45]:=

```
(* Slider boundaries for temperature *)
tempUnit = "Kelvins";
tempMin = 273;
tempMax = 10 000;

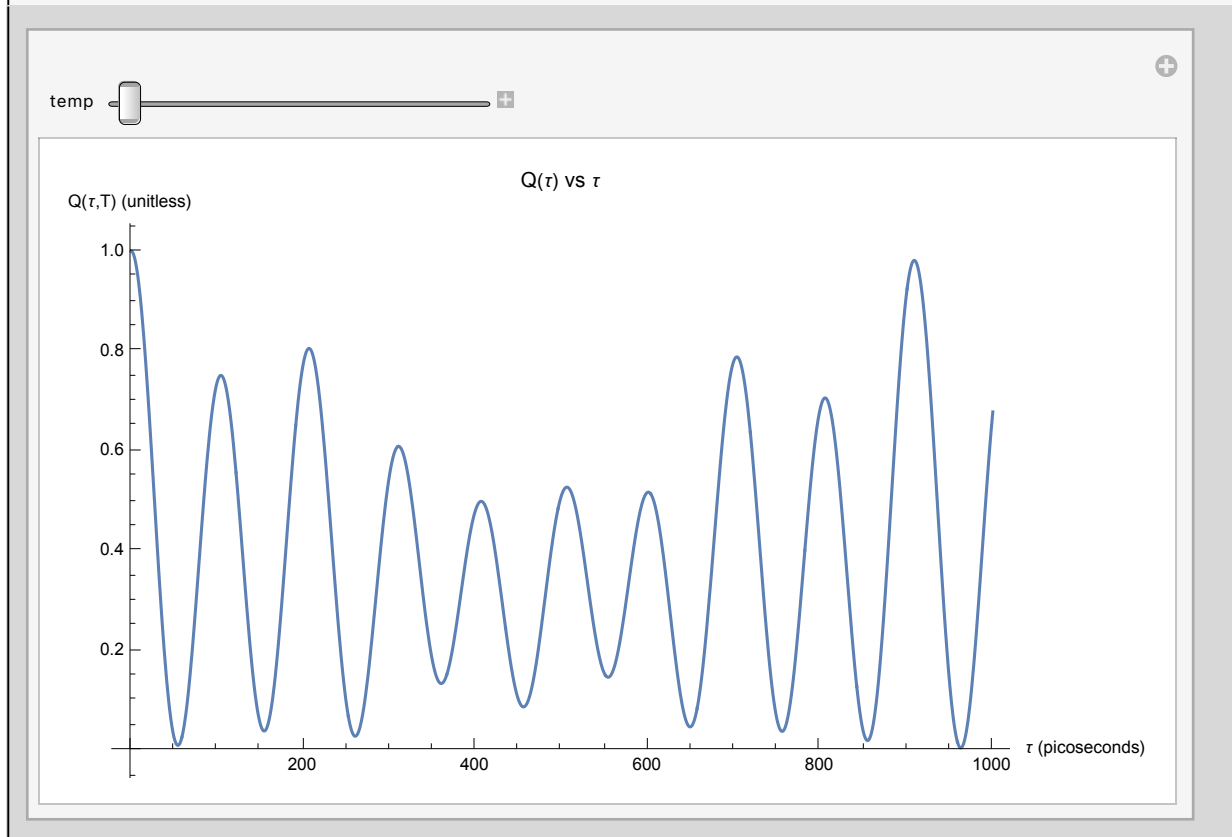
(* Boundaries for  $\tau$  *)
 $\tau$ Unit = "Picoseconds";
 $\tau$ Min = 0;
 $\tau$ Max = 1000;
```



In[82]:=

```
Manipulate[
  Plot[
    sigQ[ $\tau$ , temp],
    { $\tau$ ,  $\tau$ Min,  $\tau$ Max},
    PlotLabel  $\rightarrow$  "Q( $\tau$ ) vs  $\tau$ ",
    AxesLabel  $\rightarrow$  {" $\tau$  (picoseconds)", "Q( $\tau$ ,T) (unitless)"},
    ImageSize  $\rightarrow$  Full
  ],
  {temp, tempMin, tempMax}
]
```

Out[82]=



In[52]:=

```
(* Uncomment to have Mathematica take
care of units for comparison. BEWARE: very slow *)

(*Manipulate[
  Plot[
    sigQUnits[Quantity[ $\tau$ ,  $\tau$ Unit], Quantity[temp, tempUnit]],
    { $\tau$ ,  $\tau$ Min,  $\tau$ Max}
  ],
  {temp, tempMin, tempMax}
]*)
```