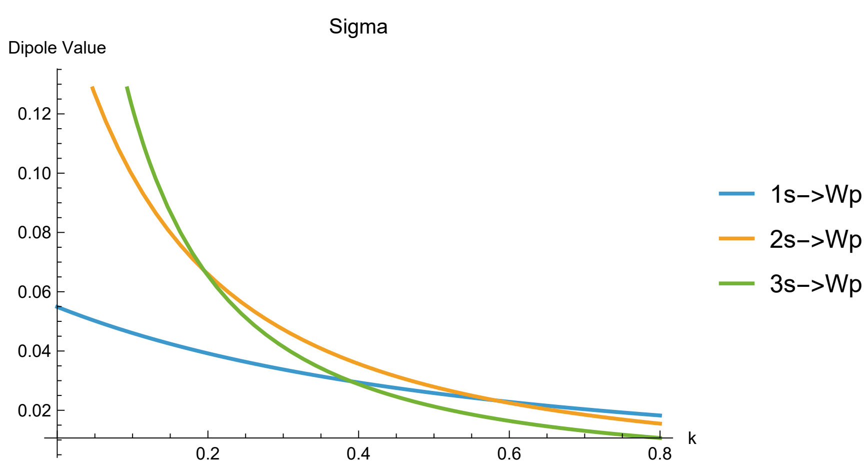
```
In[•]:= Z = 1;
        a0 = 5.29177210544 × 10^-11;  (*Bohr radius *)
        c = 299 792 458; (* Speed of light *)
        Eh = 4.3597447222060 * 10^-18;
        Ry = Eh / 2; (* Rydberg energy *)
        hbar = 1.054571817 * 10^-34;
        h = 6.62607015 * 10^-34;

In[•]:= Get["D:\\thesis\\mathematica\\dipole_moments\\dipolemoments.m"];

In[•]:= Radialdpup[n_, l_, k_, lprime_] := Module[{gValue, lmax},
          lmax = Max[l, lprime];
          gValue = gup[n - 1][n, k] / Sqrt[Rho[k]];
          (* Full formula *)
         gValue
        ]

In[•]:= Radialgup[n_, l_, W_, lprime_] := Module[{gValue, lmax, k},
          lmax = Max[l, lprime];
           k = Sqrt[W / Ry];
          gValue = gup[n - 1][n, k];
          (* Full formula *)
         gValue
        ]

In[•]:= Plot[Evaluate[{Radialgup[1, 0, W Ry, 1], Radialgup[2, 0, W Ry, 1], Radialgup[3, 0, W Ry, 1]}],
         {W, 0, 0.8},
         PlotLabel → "Sigma",
         AxesLabel → {"k", "Dipole Value"},
         PlotLegends → {"1s->Wp", "2s->Wp", "3s->Wp"},
          PlotRange → {Automatic, Automatic}]

Out[•]=
```



In[•]:=
```
Radialgup2[n_, l_, W_, lprime_] := Module[{gValue, lmax, k},
  lmax = Max[l, lprime];
  k = Sqrt[W / Ry];
  gValue = gup[n - l][n, k];
  (* Full formula *)
  Abs[gValue] ^2
]
```

In[•]:=
```
Sigma[n_, l_, W_, lprime_] := Module[{gValue, lmax, w},
  lmax = Max[l, lprime];
  w = (W + Ry / n^2) / hbar;
  gValue = gup[n - l][n, Sqrt[W / Ry]]; (*MISTAKE if lprime<l use gdown*)
  (* Full formula *)
  (4 * π^2 * w * a0 * 2 * Ry) / (3 * c) * 10^4 (lmax / (2 * l + 1)) * Abs[gValue] ^2
]
```

```
eE0 = Sqrt[h];
Ei[i_] := -Ry / i^2;
ωi[i_] := Ei[i] / hbar;
(*T[w_]:=100;*)
SumPWOmega2up[n_, l_, W_] := eE0^2 * n^4 / Z^4 *
   Sum[((l + 1)^2 - m^2) / (4 (l + 1)^2 - 1), {m, -l, l}] * Abs[gup[n - l][n, Sqrt[W / Ry]]]^2;
SumPWOmega2down[n_, l_, W_] := eE0^2 * n^4 / Z^4 *
   Sum[(l^2 - m^2) / (4 l^2 - 1), {m, -l, l}] * Abs[gdown[n - l][n, Sqrt[W / Ry]]]^2;
```

In[•]:=
```
sincTerm[W_, ω_, i_, T_] := Sinc[((W - Ei[i]) / hbar - ω) / 2 * T]^2
```

In[•]:=
```
ωmin = 0;
ωmax = - 1.5 * Ei[1] / hbar;
```

In[•]:=
```
lambda[w_] := 2 * Pi * c / w;
```

In[•]:=
```
wfromlam[L_] := 2 * Pi * c / L;
```

In[•]:=
```
lambdamin = lambda[ωmax]
lambdamax = lambda[-Ei[2] / (1.2 * hbar)]
```

Out[•]= 6.07511×10^{-8}

Out[•]= 4.37408×10^{-7}

In[•]:=
```
lambda[-Ei[1] / (hbar)]
```

Out[•]= 9.11267×10^{-8}

In[•]:=
```
lambda[-Ei[2] / (hbar)]
```

Out[•]= 3.64507×10^{-7}

Case 1

In[∘]:=
```
integrand[W_, ω_, T_] := T^2 / (2 * hbar^2) (SumPWOmega2up[1, 0, W] × sincTerm[W, ω, 1, T] +
    SumPWOmega2up[2, 0, W] × sincTerm[W, ω, 2, T])

Pni[ω_?NumericQ, T_?NumericQ] :=
 NIntegrate[integrand[W, ω, T], {W, 0, (ω + 10 * 2 Pi / T) hbar}]
```

In[∘]:=
```
(* Plot range *)

data1 = Table[{ω, Pni[ω, -8 Pi hbar / Ei[1]]}, {ω, ωmin, ωmax, (ωmax - ωmin) / 100}];
(* 100 points *)
(*Export["Pni_data_M=2T=100_v0.wl", data, "WL"];  Wolfram Language format *)
```

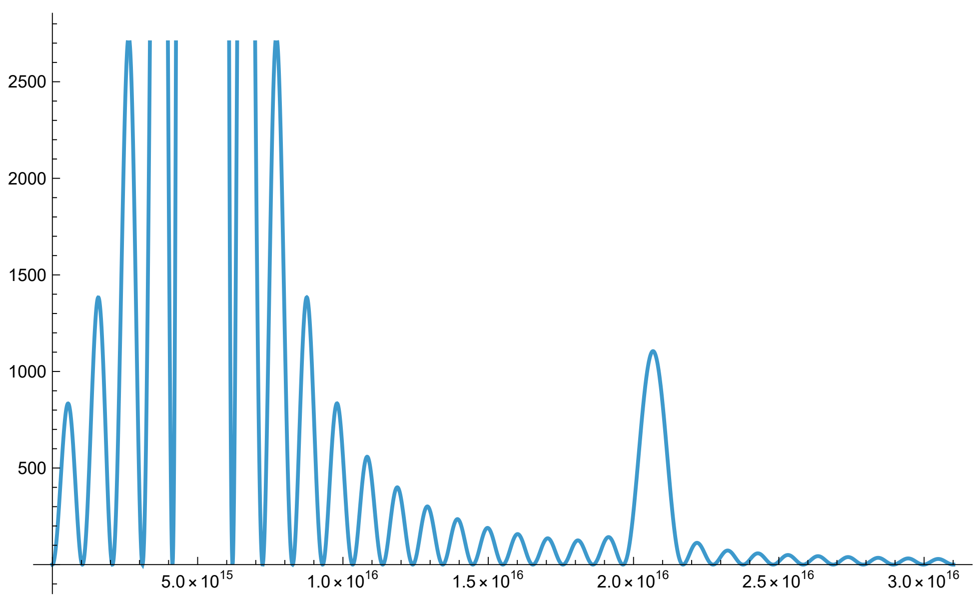
In[∘]:=
```
Ei[1] / hbar
```

Out[∘]=

-2.06707×10^{16}

In[∘]:=
```
Plot[integrand[0, ω, -40 Pi hbar / Ei[1]], {ω, ωmin, ωmax}]
```

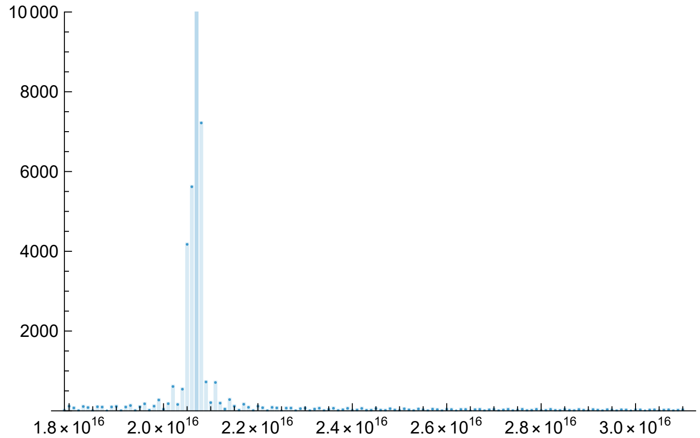
Out[∘]=



In[∘]:=
```
Plot[integrand[0, ω, -4000 Pi hbar / Ei[1]], {ω, ωmin, ωmax}]
```

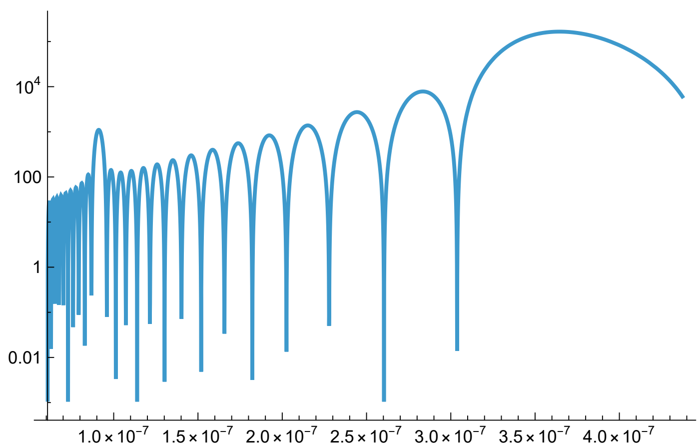
In[•]:=
```
DiscretePlot[integrand[0, ω, -4000 Pi hbar / Ei[1]],
  {ω, 1.79 * 10^16, ωmax, 0.01 * 10^16}, PlotRange → {Automatic, {0, 10000}}]
```

Out[•]=



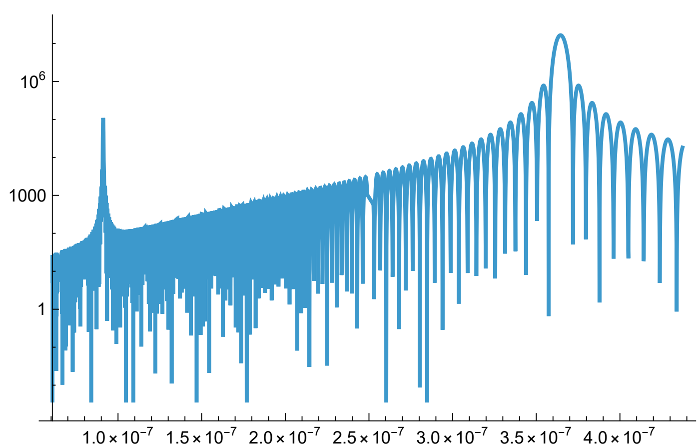
In[•]:=
```
LogPlot[integrand[0, wfromlam[L], -40 Pi hbar / Ei[1]], {L, lambdamin, lambdamax}]
```

Out[•]=



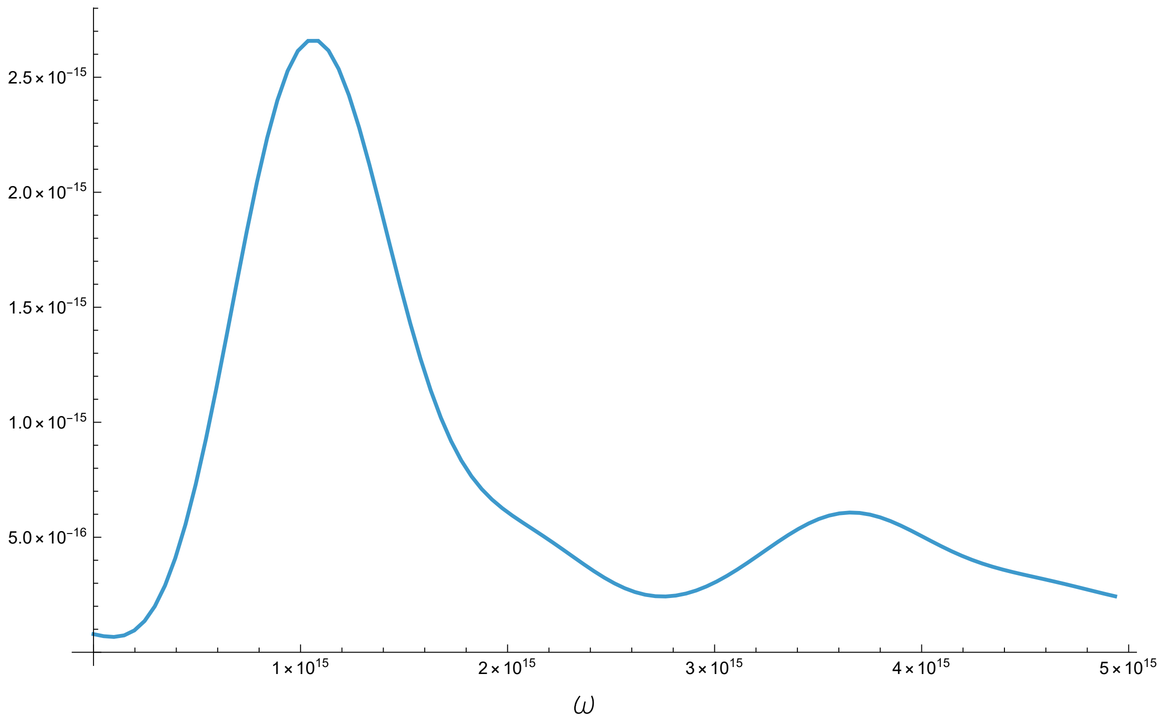
In[•]:=
```
LogPlot[integrand[0, wfromlam[L], -400 Pi hbar / Ei[1]], {L, lambdamin, lambdamax}]
```

Out[•]=



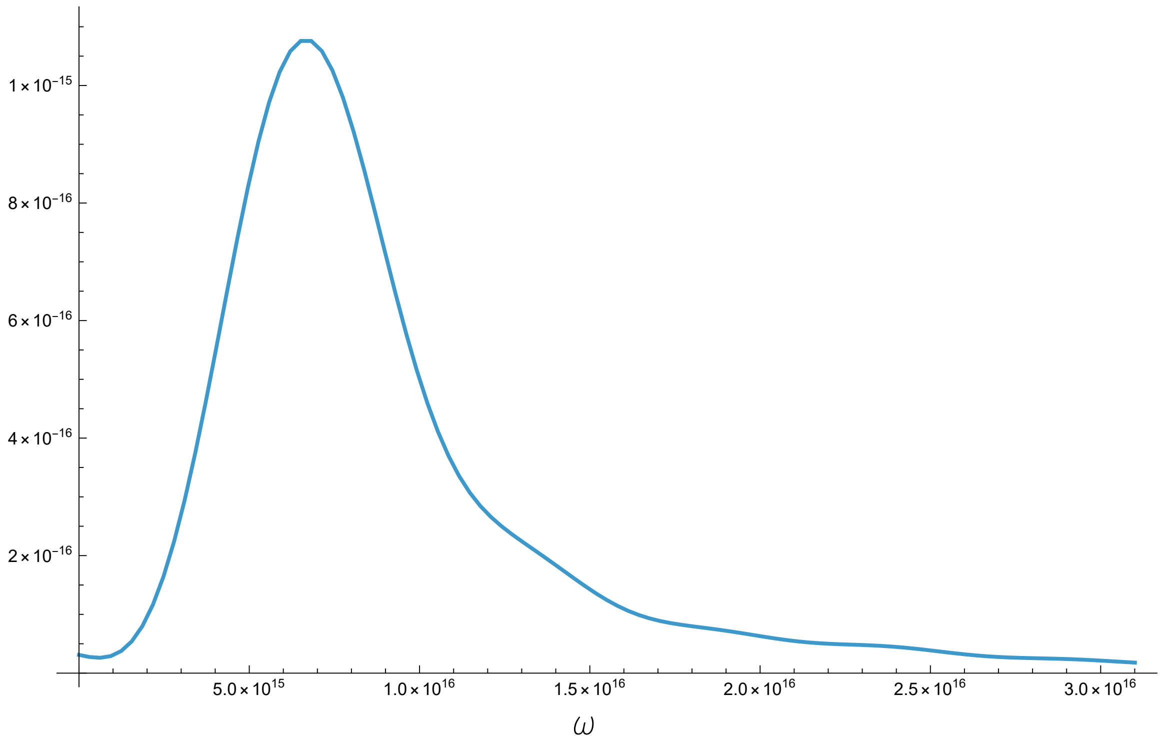
```
(* Plot Pni(ω) *)
P = Labeled[
 ListPlot[data1,
 ImageSize → 600, Joined → True],
 Style["ω", FontSize → 20],
 Bottom
]
```

Out[ ]=



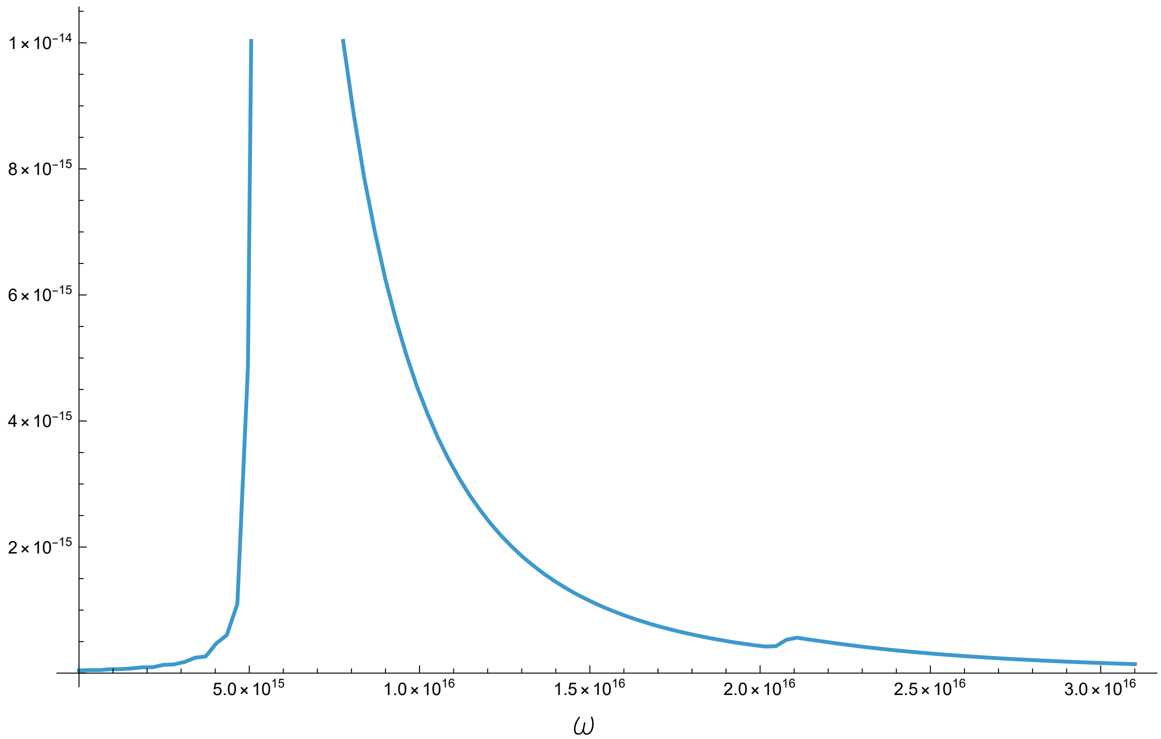
In[ ]:=
```
P = Labeled[
 ListPlot[data1,
ImageSize → 600, Joined → True],
 Style["ω",  FontSize → 20],
 Bottom
]
```

Out[ ]=



```
In[∘]:= P = Labeled[
  ListPlot[data2,
 ImageSize → 600, Joined → True],
  Style["ω",  FontSize → 20],
  Bottom
]
```

Out[∘]=




```
cm = 72 / 2.54 ; (* centimetre *)
Export["Exports/IP_M1s2sT=E1by4E0e=sqrth_v3.png", P, "png", ImageResolution → 300]
```

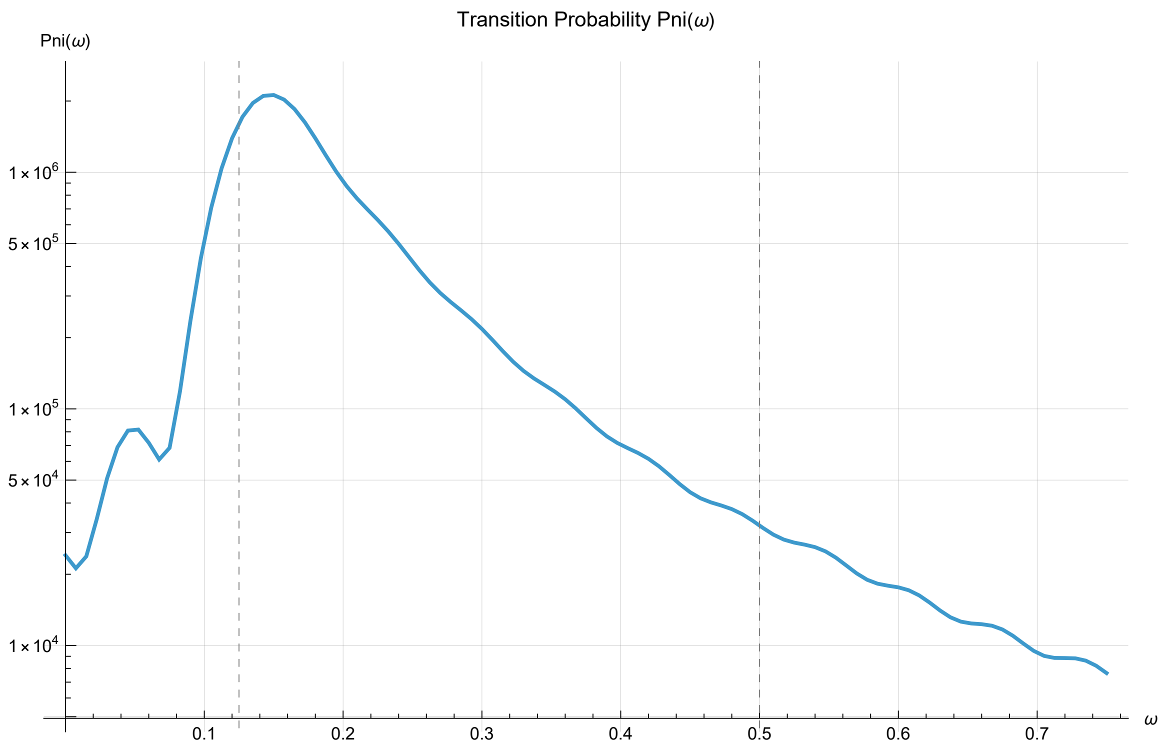
Out[∘]=

```
Exports/IP_M1s2sT=E1by4E0e=sqrth_v3.png
```

```
ωmin = 0;
ωmax = - 1.5 * Ei[0];
data2 = Table[{ω, Pni[ω, 1000]}, {ω, ωmin, ωmax, (ωmax - ωmin) / 100}];
(* 100 points *)
Export["Pni_data_M=2_v0.wl", data2, "WL"]; (* Wolfram Language format *)
```

```
ListLogPlot[data2,
 PlotLabel → "Transition Probability Pni(ω)",
 AxesLabel → {"ω", "Pni(ω)"},
 GridLines → Automatic,
 ImageSize → 600, Joined → True,
Epilog → Table[{Gray, Dashed, Line[{{1 / (2 * i^2), 0}, {1 / (2 * i^2), 40000}}]}, {i, M}]]
(* Control recursion depth *)
```

Out[∘]=



```
(* Constants and Parameters *)
i = 2;          (* Defined i *)
Ei = -1/(2*i^2);      (* Defined E_i *)
Γ23 = Gamma[2/3];     (* Gamma function *)
T[ω_] := 100;      (* T depends on omega *)
E0e = 1;

(* Prefactor term - using En instead of E to avoid conflict *)
prefactor[En_] := 1

(* Sinc-squared term *)
sincTerm[En_, ω_] := Sinc[(En - Ei - ω)/2 * T[ω]]^2

(* Full integrand *)
integrand[En_, ω_] := 1 * prefactor[En] * sincTerm[En, ω]

(* Numerical integration for Pni(ω) *)
Pni[ω_?NumericQ] := NIntegrate[integrand[En, ω], {En, 0, ∞}]

(* Plot range *)
ωmin = 0;
ωmax = -2*Ei;

(* Plot Pni(ω) *)
Plot[Pni[ω], {ω, ωmin, ωmax},
 PlotLabel → "Transition Probability Pni(ω)",
 AxesLabel → {"ω", "Pni(ω)"},
 PlotRange → All,
 GridLines → Automatic,
 PlotPoints → 30, (* Increase sampling for smoother plot *)
 MaxRecursion → 3] (* Control recursion depth *)
```

Out[∘]=