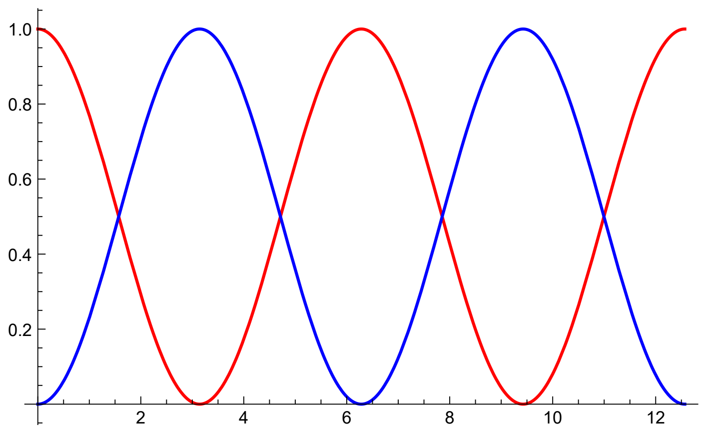
Two Level System and Rabi Flopping

In[11]:=
```
sol = DSolve[{c1'[t] == -i * Ω/2 * Exp[-i * ϕ] * Exp[i * δ * t] * c2[t],
    c2'[t] == -i * Ω/2 * Exp[i * ϕ] * Exp[-i * δ * t] * c1[t],
    c1[0] == 1, c2[0] == 0}, {c1[t], c2[t]}, t][[1]];
```

δ=0

In[18]:=
```
Plot[{Abs[c1[t]]^2, Abs[c2[t]]^2} /. sol /. {Ω → 1, δ → 0, ϕ → 0} // Evaluate,
 {t, 0, 4 * π}, PlotStyle → {Red, Blue}]
```

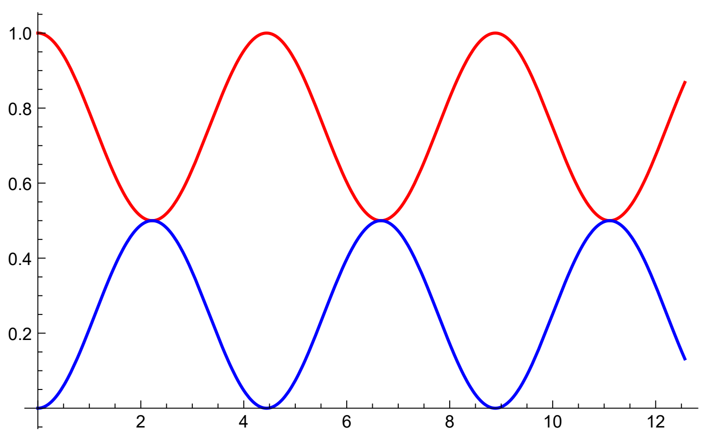
Out[18]=



δ=Ω

In[19]:=
```
Plot[{Abs[c1[t]]^2, Abs[c2[t]]^2} /. sol /. {Ω → 1, δ → 1, ϕ → 0} // Evaluate,
 {t, 0, 4 * π}, PlotStyle → {Red, Blue}]
```

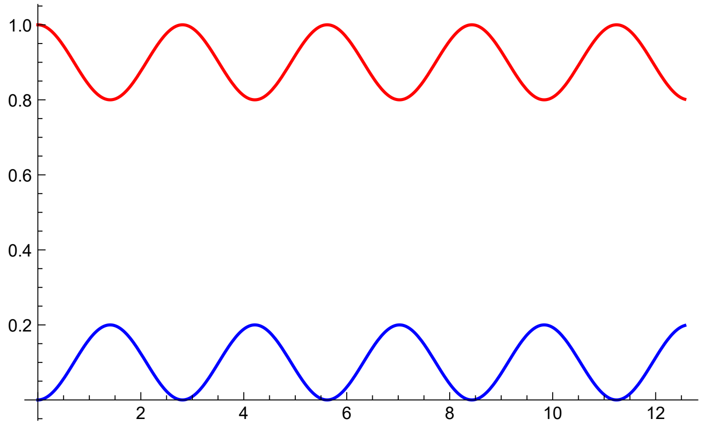
Out[19]=



δ=2*Ω

In[20]:=
```
Plot[{Abs[c1[t]]^2, Abs[c2[t]]^2} /. sol /. {Ω → 1, δ → 2, ϕ → 0} // Evaluate,
 {t, 0, 4 * π}, PlotStyle → {Red, Blue}]
```

Out[20]=



AC Stark Shifts

In[24]:= $H = \hbar * \{\{0, \frac{\Omega}{2} * \mathrm{Exp}[-i * \phi]\}, \{\frac{\Omega}{2} * \mathrm{Exp}[i * \phi], -\delta\}\}$

Out[24]= $\{\{0, \frac{1}{2} e^{-i\phi} \Omega \hbar\}, \{\frac{1}{2} e^{i\phi} \Omega \hbar, -\delta \hbar\}\}$

In[25]:= `H // MatrixForm`

Out[25]//MatrixForm=

$$\begin{pmatrix} 0 & \frac{1}{2} e^{-i\phi} \Omega \hbar \\ \frac{1}{2} e^{i\phi} \Omega \hbar & -\delta \hbar \end{pmatrix}$$

In[35]:= $\mathrm{FullSimplify}[\mathrm{Eigensystem}[H] /. \{\sqrt{e^{2 i \phi} (\delta^2 + \Omega^2)} \to e^{i * \phi} * \sqrt{\delta^2 + \Omega^2}\}]$

Out[35]= $\{\{-\frac{1}{2}\left(\delta + \sqrt{\delta^2 + \Omega^2}\right) \hbar, \frac{1}{2}\left(-\delta + \sqrt{\delta^2 + \Omega^2}\right) \hbar\}, \{\{\frac{e^{-i\phi}\left(\delta - \sqrt{\delta^2 + \Omega^2}\right)}{\Omega}, 1\}, \{\frac{e^{-i\phi}\left(\delta + \sqrt{\delta^2 + \Omega^2}\right)}{\Omega}, 1\}\}\}$

Bragg Diffraction Simulations

time in units of $\frac{1}{4\,\omega_r}$, frequencies in units of $4\,\omega_r$

Two-Photon Bragg

very efficient π pulse for low Rabi frequency

In[126]:=

```
Ω = 0.3/4;
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -ⅈ * Ω/2 * (Exp[ⅈ * 1 * τ]) * g_(n+1)[τ] - ⅈ * Ω/2 * (Exp[-ⅈ * 1 * τ]) * g_(n-1)[τ] -
      ⅈ * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[135]:=

```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, π/Ω}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[17 * n/Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[135]=

In[136]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → π/Ω]^2
```

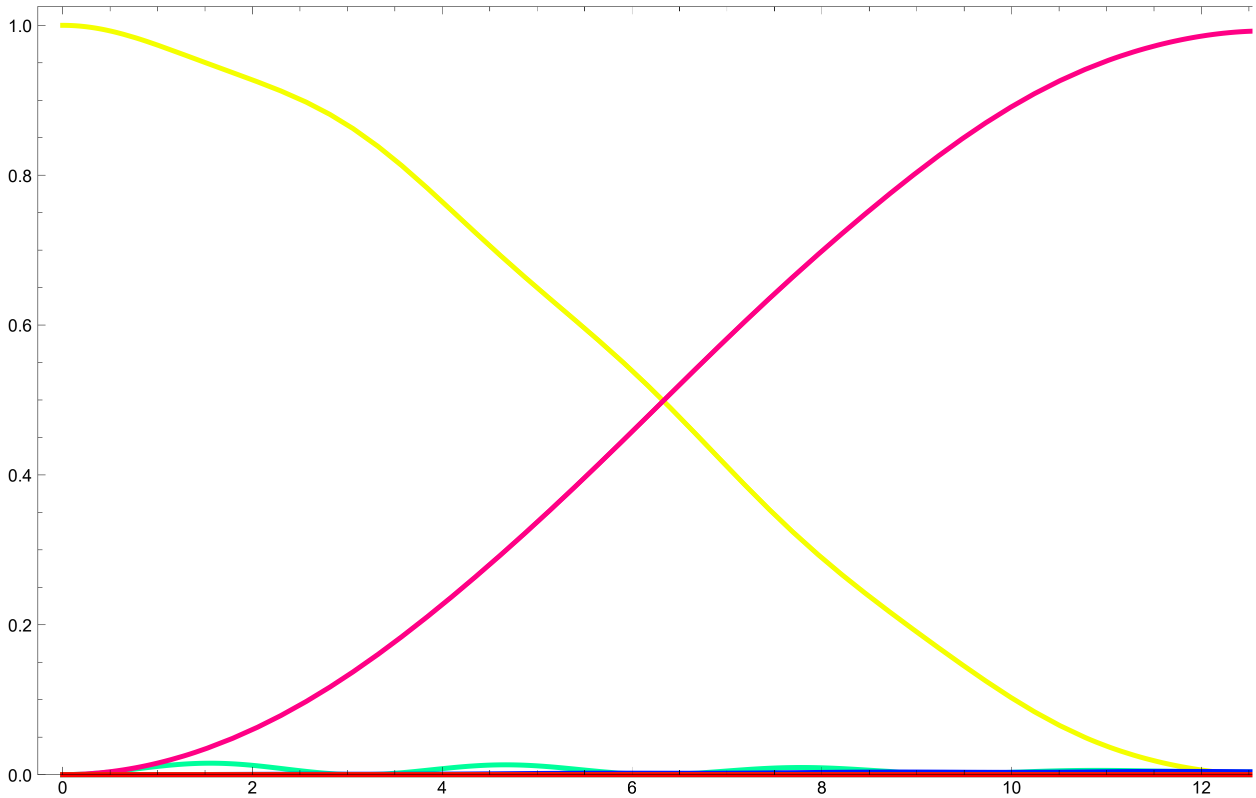
Out[136]= 0.999313

other levels start to get populated as Rabi frequency increases

In[137]:=
```
Ω = 1/4;
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n '[τ] == -ⅈ * Ω/2 * (Exp[ⅈ * 1 * τ]) * g_(n+1)[τ] - ⅈ * Ω/2 * (Exp[-ⅈ * 1 * τ]) * g_(n-1)[τ] -
      ⅈ * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1) '[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1) '[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[146]:=
```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, π/Ω}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[(17 * n)/(Length[coefs])], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[146]=



In[147]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → π/Ω]^2
```

Out[147]= 0.992327

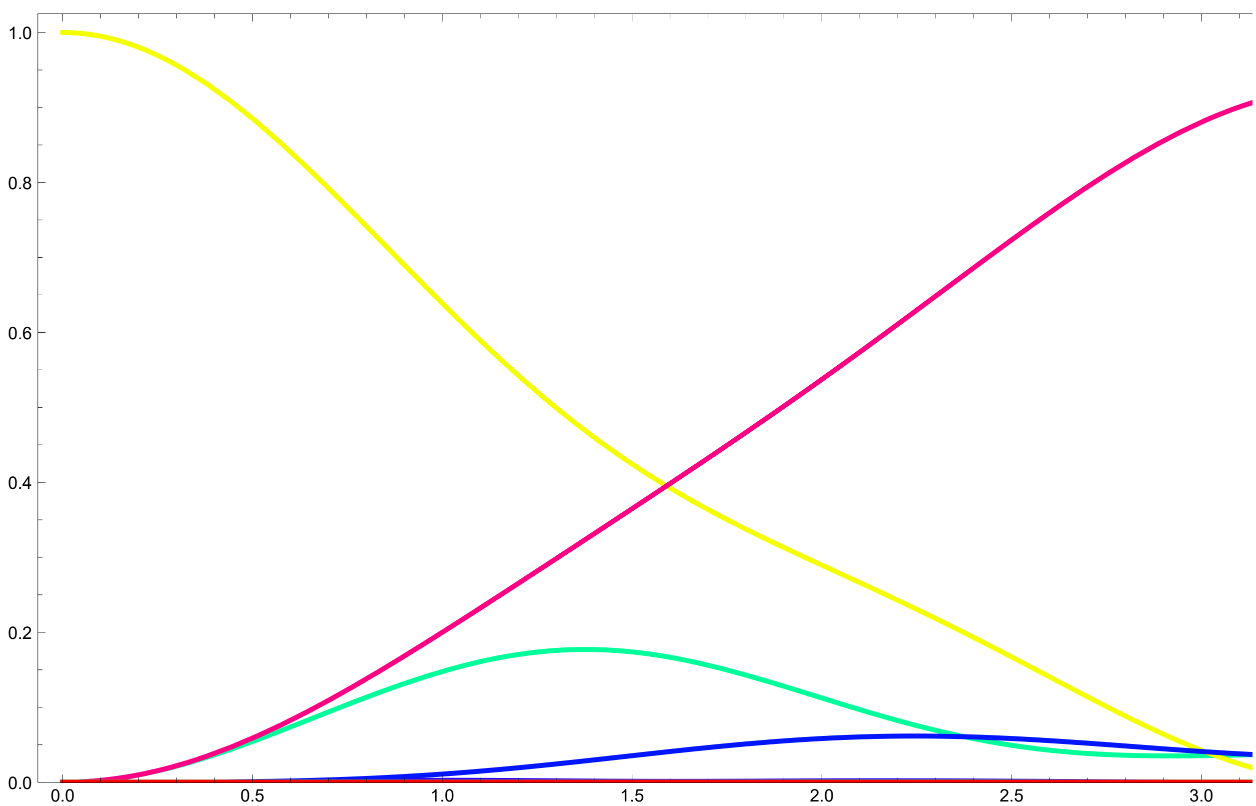
In[148]:=

```
Ω = 4/4;
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -i * Ω/2 * (Exp[i * 1 * τ]) * g_(n+1)[τ] - i * Ω/2 * (Exp[-i * 1 * τ]) * g_(n-1)[τ] -
      i * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[157]:=

```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, π/Ω}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[17 * n / Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[157]=



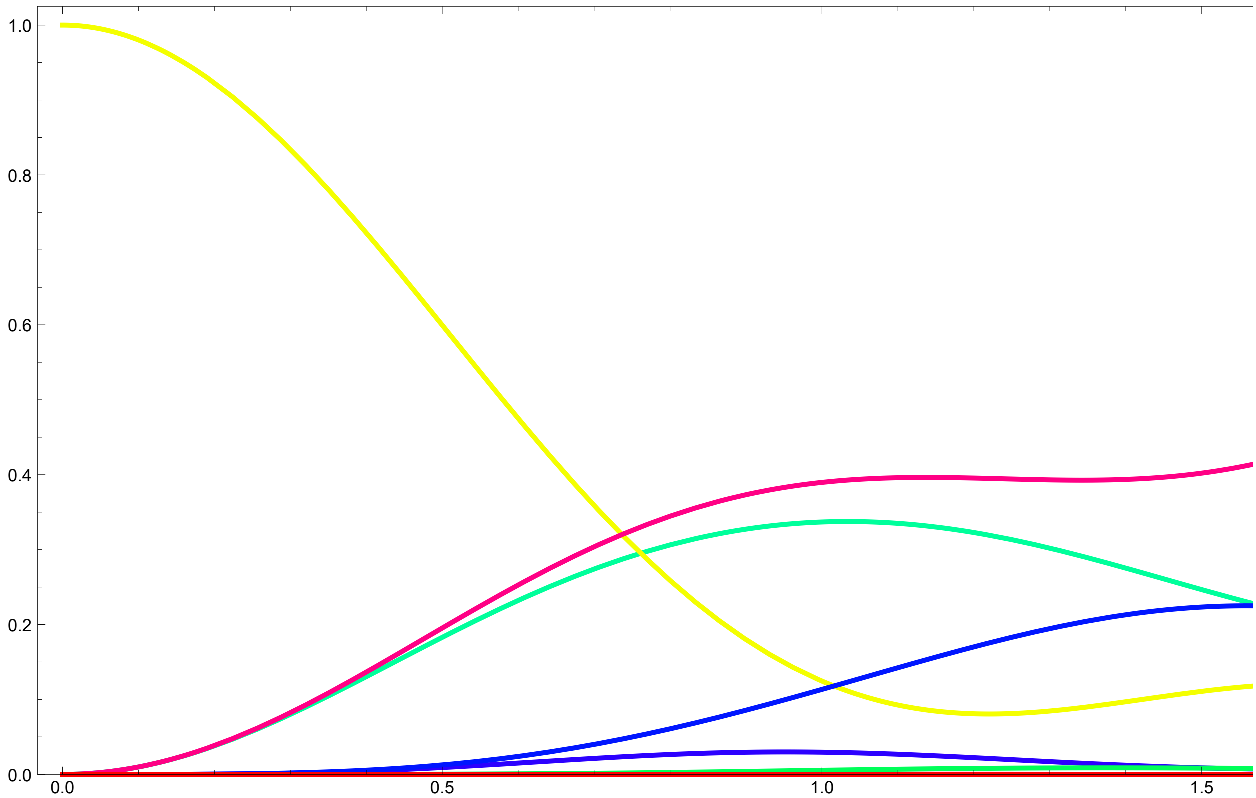
In[158]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → π/Ω]^2
```

Out[158]=
```
0.907597
```

In[159]:=
```
Ω = 8/4;
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -ⅈ * Ω/2 * (Exp[ⅈ * 1 * τ]) * g_(n+1)[τ] - ⅈ * Ω/2 * (Exp[-ⅈ * 1 * τ]) * g_(n-1)[τ] -
      ⅈ * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[168]:=
```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, π/Ω}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[17 * n / Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[168]=



In[169]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → π/Ω]^2
```

Out[169]= 0.41452

It can help to use Gaussian shaped pulses instead of square pulses, though still do not want to make Rabi frequency too high

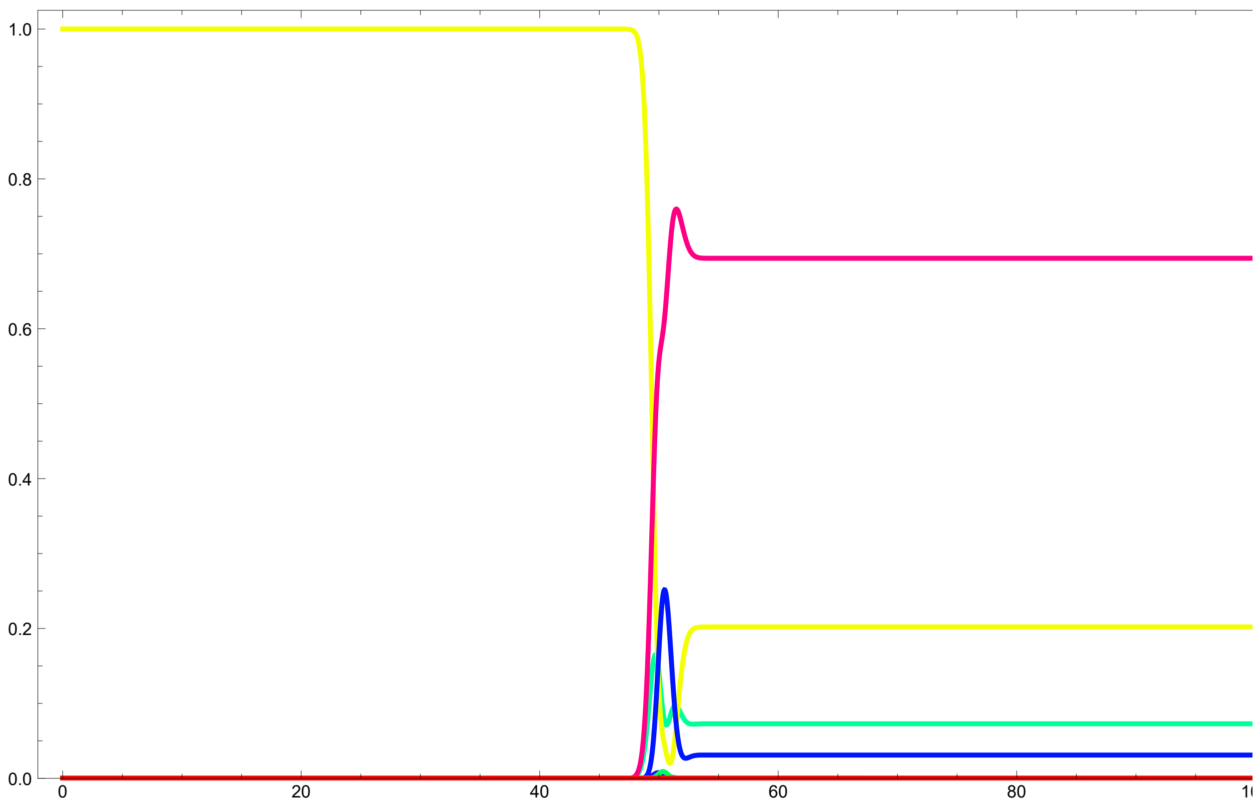
In[320]:=

```
Ω = 8/4 * ⅇ^(-(τ-50)^2/1.2^2);
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -ⅈ * Ω/2 * (Exp[ⅈ * 1 * τ]) * g_(n+1)[τ] - ⅈ * Ω/2 * (Exp[-ⅈ * 1 * τ]) * g_(n-1)[τ] -
      ⅈ * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
   Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[319]:=

```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, 100}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[(17 * n)/Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[319]=



In[330]:=

```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → 100]^2
```

Out[330]= 0.781413

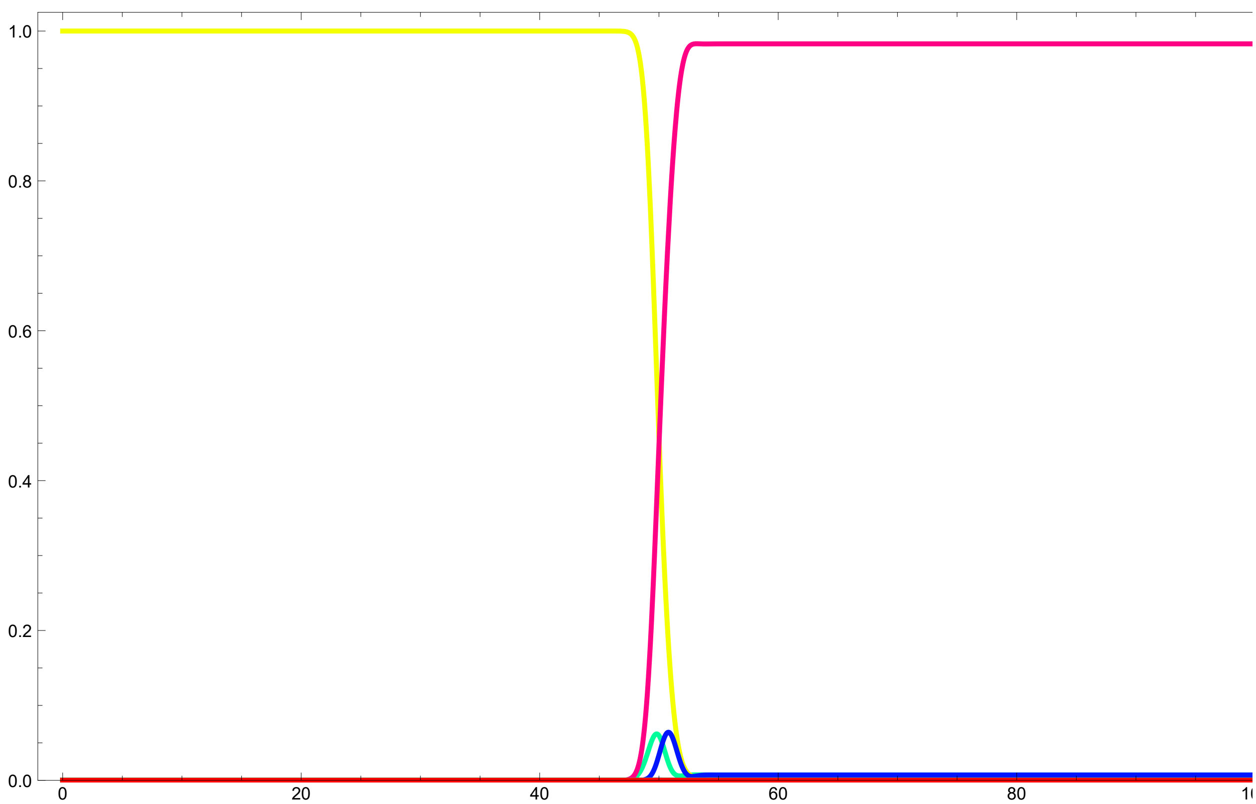
In[400]:=

```
Ω = 4/4 * ⅇ^(-(τ-50)^2/1.8^2);
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -ⅈ * Ω/2 * (Exp[ⅈ * 1 * τ]) * g_(n+1)[τ] - ⅈ * Ω/2 * (Exp[-ⅈ * 1 * τ]) * g_(n-1)[τ] -
      ⅈ * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[409]:=

```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, 100}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[17 * n / Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[409]=



In[410]:=

```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 1 + 2]] /. τ → 100]^2
```

Out[410]= 0.982926

Six-Photon Bragg

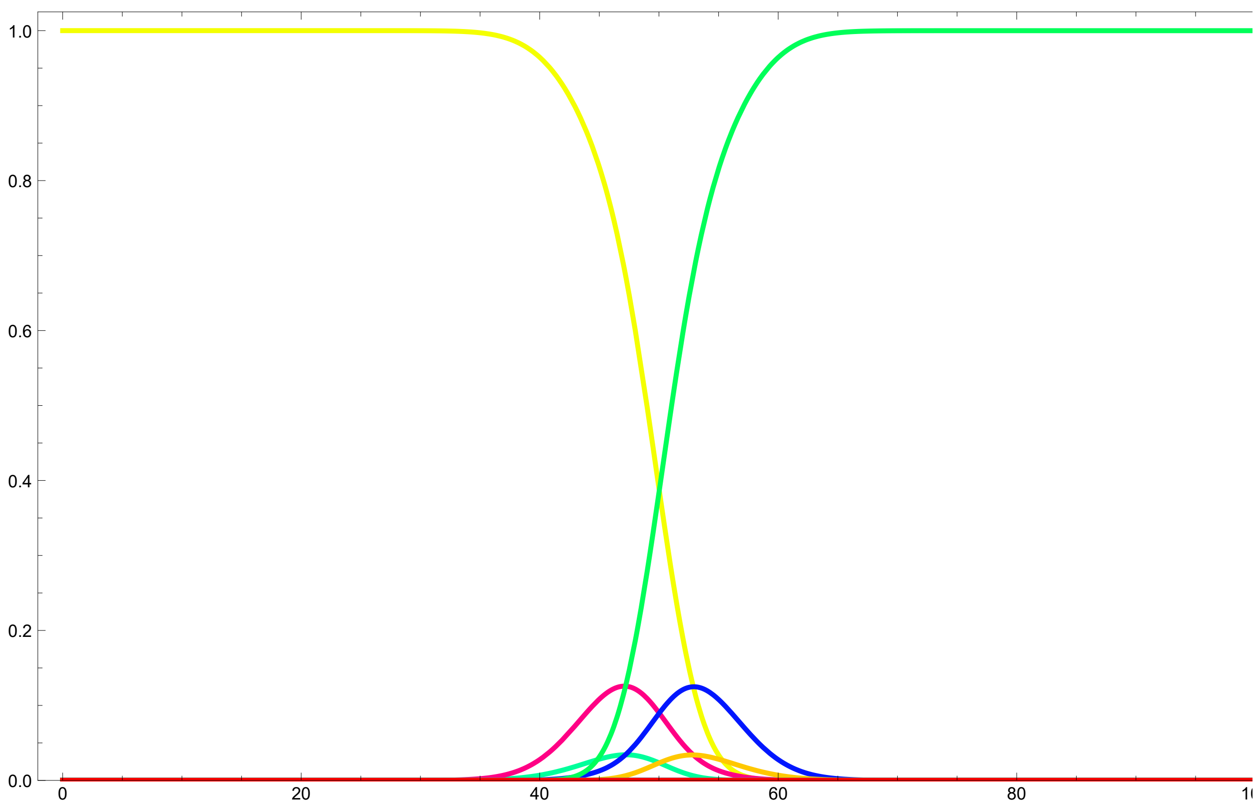
In[411]:=

```
Ω = 7.3/4 * e^(-(τ-50)^2/10^2);
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[gₙ[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[gₙ'[τ] == -i * Ω/2 * (Exp[i * 3 * τ]) * g_(n+1)[τ] - i * Ω/2 * (Exp[-i * 3 * τ]) * g_(n-1)[τ] -
      i * (n^2) * gₙ[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1)'[τ] == 0, g_-(NegMomentumStates+1)[0] == 0,
     g_(PosMomentumStates+1)'[τ] == 0, g_(PosMomentumStates+1)[0] == 0},
    Table[gₙ[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[420]:=

```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, 100}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[17 * n/Length[coefs]], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[420]=



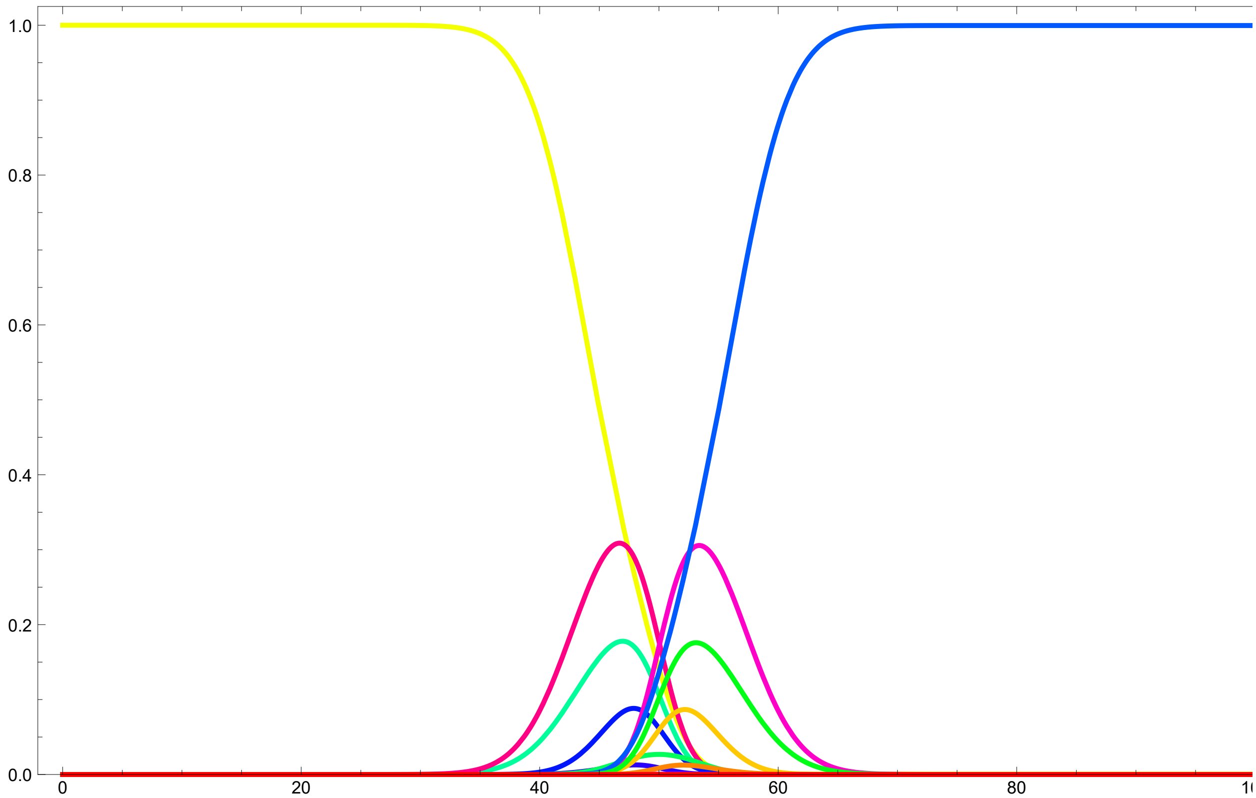
In[421]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 3 + 2]] /. τ → 100]^2
```
Out[421]= 0.999936

Twelve-Photon Bragg

In[572]:=
```
Ω = 33/4 * e^(-(τ-50)^2/10^2);
PosMomentumStates = 15;
NegMomentumStates = 5;
T = 100;
coefs = Table[g_n[τ], {n, -(NegMomentumStates + 1), PosMomentumStates + 1}];
dq = Table[g_n'[τ] == -i * Ω/2 * (Exp[i * 6 * τ]) * g_(n+1)[τ] - i * Ω/2 * (Exp[-i * 6 * τ]) * g_(n-1)[τ] -
      i * (n^2) * g_n[τ], {n, -NegMomentumStates, PosMomentumStates}];
initial = Join[{g_-(NegMomentumStates+1) '[τ] == 0, g_-(NegMomentumStates+1) [0] == 0,
     g_(PosMomentumStates+1) '[τ] == 0, g_(PosMomentumStates+1) [0] == 0},
    Table[g_n[0] == If[n == 0, 1, 0], {n, -NegMomentumStates, PosMomentumStates}]];
eqs = Join[dq, initial];
dqsol = NDSolve[eqs, coefs, {τ, 0, 100}, MaxSteps → 22 000 000];
```

In[581]:=
```
plot1 = Plot[((coefs /. dqsol[[1]]) // Abs)^2 // Evaluate,
  {τ, 0, 100}, PlotRange → {0, 1.025}, Frame → True,
  PlotStyle → Table[{Hue[(17 * n)/(Length[coefs])], Thickness[0.004]}, {n, 1, Length[coefs]}]]
```

Out[581]=



In[582]:=
```
Abs[(coefs /. dqsol[[1]])[[NegMomentumStates + 6 + 2]] /. τ → 100]^2
```

Out[582]= 0.99954