Prepare

In[190]:=
```
imgsize = 700;
imgpadding = {{70, 20}, {50, 10}};
```

Always use the first element in the vector to be the component of small eigenvalue.

In[192]:=
```
$MachinePrecision
```

Out[192]=
```
15.9546
```

Modules

Making Plots

In[193]:=
```
pltNorm[sol_, c1_, c2_, start_, end_] :=
  Plot[Evaluate[Abs[c1[x]]^2 + Abs[c2[x]]^2 /. sol[[1]]],
   {x, start, end}, PlotRange → All, ImageSize → imgsize, Frame → True];
```

Plot transitioin Probabilities: Probability of **c2**

In[194]:=
```
pltProb[sol_, c2_, c1_, start_, end_, pltLabel_, frameLabel_] :=
  Plot[Evaluate[Abs[c2[x]]^2 / (Abs[c1[x]]^2 + Abs[c2[x]]^2) /. sol[[1]]],
   {x, start, end}, PlotRange → All, PlotLabel → pltLabel,
   ImageSize → imgsize, Frame → True, FrameLabel → frameLabel,
   ImagePadding → imgpadding, PerformanceGoal → "Quality"];
```

Rotation matrices

Rotation from vacuum to instantaneous matter basis

In[195]:=
```
vac2instRotation[a0_, a1_, tv_] :=
 Module[{thetaxConst, alpha0 = a0, alpha1 = a1, thetav = tv}, thetaxConst = 1/2 ArcSin[
     Sin[2 thetav] / (√(1 + (alpha0 + alpha1)^2 - 2 (alpha0 + alpha1) Cos[2 thetav]))];
  {{Cos[thetav - thetaxConst], Sin[thetav - thetaxConst]},
   {-Sin[thetav - thetaxConst], Cos[thetav - thetaxConst]}}]
```

Pauli matrices

In[196]:=
```
sigma3 = PauliMatrix[3];
sigma1 = PauliMatrix[1];
```

Conversion from length (km) to energy (eV): 197.33 MeV*fm=1 =>

In[198]:=
```
hbarc = 197.33;
```

In[199]:=
```
km2eV[km_] := Module[{length = km}, (hbarc * 10^6) / (km * 10^(18))]
eV2km[eV_] := Module[{energy = eV}, (hbarc * 10^(-18)) / (eV * 10^(-6))]
```

Paramters

Paramters are grabbed from Kneller’s paper

Initial Condition

In (averaged) matter basis

In[201]:=
```
θv = 0.573;
energy = 20 * 10^6; (* eV *)
deltam2 = 3 * 10^(-3); (* eV^2 *)
(* The following are calculated *)
omegav = deltam2 / (2 energy); (* 7.5*10^(-11) *)
(*eV, delta^2m= 3*10^(-3)eV^2, E =20MeV*)
α0 = Cos[2 θv]
(* α0=0;*)
α1 = 0;
(* α1=0.1α0; *)
(*β=0.937*) β = (2 * km2eV[5.2533]) / omegav
phasem = 0;
```

Out[205]= 0.412135

Out[207]= 1.00168

In[209]:=
```
θm = ArcTan[Sin[θv] / (Cos[θv] - α0)]
```

Out[209]= 0.902364

The initial condition in matter basis

In[210]:=
```
initAM = {1, 1} / √2
```

Out[210]= $\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

Rotate to vacuum eigenbasis

In[211]:=
```
vac2instRotation[α0, 0, θv] // MatrixForm
initVac = Transpose[vac2instRotation[α0, 0, θv]].initAM
```

Out[211]//MatrixForm=

$$\begin{pmatrix} 0.977528 & -0.210805 \\ 0.210805 & 0.977528 \end{pmatrix}$$

Out[212]= {0.840278, 0.542155}

In[213]:=
```
endpoint = omegav / km2eV[10^4];
% // N
```

Out[214]= 3800.74

Test of Defined Functions and Parametser + Check Numbers

In[215]:=
```
"σ3 is " MatrixForm@sigma3
"σ1 is " MatrixForm@sigma1
```

Out[215]= σ3 is $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Out[216]= σ1 is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Test Rotation

In[217]:=
```
vac2instRotationTest = Module[{θv = ArcSin@√0.307},
  Grid[{{"Zero Matter Potential: " <> ToString@vac2instRotation[0, 0, θv]},
    {"MSW Resonance: " <> ToString@vac2instRotation[Cos[2 * θv], 0, θv]}}]]
```

Out[217]=
```
          Zero Matter Potential: {{1., 0.}, {0., 1.}}
MSW Resonance: {{0.980433, -0.196852}, {0.196852, 0.980433}}
```

Test Unit convsersion

In[218]:=
```
km2eV[10^(-18)]
eV2km[10^6]
```

Out[218]= 1.9733×10^{8}

Out[219]= 1.9733×10^{-16}

Check Numbers

In[220]:=

$$\frac{\mathrm{ArcSin}\left[\frac{\mathrm{Sin}[2\,\theta v]}{\sqrt{1+(\alpha 0+\alpha 1)\mathrm{\hat{}}2-2\,(\alpha 0+\alpha 1)\,\mathrm{Cos}[2\,\theta v]}}\right]}{2}$$

$$\mathrm{Sin}\left[2\,\mathrm{ArcTan}\left[\frac{\mathrm{Sin}[2\,\theta v]}{\mathrm{Cos}[2\,\theta v]-\alpha 0}\right]\Big/\,2\right]\mathrm{\hat{}}2$$

Out[220]= 0.785398

Power::infy : Infinite expression $\frac{1}{0.}$ encountered. >>

Out[221]= Indeterminate

In[222]:= `omegav // N`

Out[222]= 7.5×10^{-11}

In[223]:= `(6 / omegav) * 10^(-7) * 197 / Pi // N`

Out[223]= 501 656.

In[224]:= `1 / km2eV[1] // N`

Out[224]= 5.06765×10^{9}

In[225]:= `θv * Pi / 180`

Out[225]= 0.0100007

In[226]:=

$$\left(2\,\mathrm{Pi}\,/\,\mathrm{omegav}*\sqrt{1+\alpha 0\mathrm{\hat{}}2-2\,\alpha 0\,\mathrm{Cos}[2\,\theta v]}\right)*10\mathrm{\hat{}}(-7)*197$$

Out[226]= 1.5037×10^{6}

In[227]:= `(100 / omegav) * 10^(-7) * 197 // N`

Out[227]= 2.62667×10^{7}

Some Conversions and Comparisons

A Table of All Parameters

In[228]:=
```
Grid[
 {{"-", "θv", "Δm^2/eV^2", "E/MeV", "ωv/eV", "λm/km", "C*", "Phase of Matter Profiles"},
  {"Kneller", 0.573, N@3 * 10^(-3), 20, N@3 * 10^(-3) / (2 * 20 * 10^6), 5.27, 0.1, "η"},
  {"Now", θv, N@deltam2, N@energy * 10^(-6), N@omegav,
   eV2km[β omegav / 2], α1 / α0, phasem}}, Frame → All]
```

Out[228]=

-	θ_V	$\Delta m^2/eV^2$	E/MeV	ω_V/eV	λ_m/km	C_*	Phase of Matter Profiles
Kneller	0.573	0.003	20	7.5×10^{-11}	5.27	0.1	η
Now	0.573	0.003	20.	7.5×10^{-11}	5.2533	0.	0

Numerical Check

Frequencies/wavelength in Kneller’s paper

Vacuum

In[229]:=
```
N@omegav
```

Out[229]= 7.5×10^{-11}

Vacuum Wavelegth to km (Using Kneller’s convention that wavelength=2/)

In[230]:=
```
eV2km[omegav] * 2
```

Out[230]= 5.26213

Matter frequency

In[231]:=
```
omegam = omegav * Sqrt[α0^2 + 1 - 2 α0 Cos[2 θv]]
```

Out[231]= 6.83342×10^{-11}

Corresponding wavelength

In[232]:=
```
eV2km[omegam] * 2
```

Out[232]= 5.77544

Wavelength of transition probability in Kneller’s resonance

In[233]:=
```
(6 * 10^9 / 5) * 10^(-5)
```

Out[233]= 12 000

This corresponds to many vacuum lengths

In[234]:=
```
% / (eV2km[omegav] * 2)
```

Out[234]= 2280.44

Equation Solving

Sovling Schrodinger equation Hamiltonian

The normalized Hamiltonian in vacuum basis is

In[235]:=
```
hamilVac[alpha0_, alpha1_, beta_, x_] :=
 Module[{lambda = alpha0 + alpha1 * Sin[beta * x + phasem]},
  - 1/2 sigma3 + lambda/2 Cos[2 * θv] sigma3 + lambda/2 Sin[2 * θv] sigma1]
```

In[236]:=
```
hamilVac[α0, α1, β, x]
```

Out[236]=
```
{{-0.415072, 0.187753}, {0.187753, 0.415072}}
```

In[237]:=
```
waveVac0[x_] = {waveVac01[x], waveVac02[x]};
solVac0 = NDSolve[{I D[waveVac0[x], x] == hamilVac[α0, α1, β, x].waveVac0[x],
    waveVac0[0] == initVac}, waveVac0[x], {x, 0, endpoint}]
```

Out[238]=
```
{{waveVac01[x] → InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x],
  waveVac02[x] → InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x]}}
```

In[239]:=
```
solVac0Convergence = Assuming[waveVac02[x] ∈ Complexes, NDSolve[
    {I D[waveVac0[x], x] == hamilVac[α0, α1, β, x].waveVac0[x], waveVac0[0] == initVac},
    waveVac0[x], {x, 0, endpoint}, Method → "StiffnessSwitching"]]
```

Out[239]=
```
{{waveVac01[x] → InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x],
  waveVac02[x] → InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x]}}
```

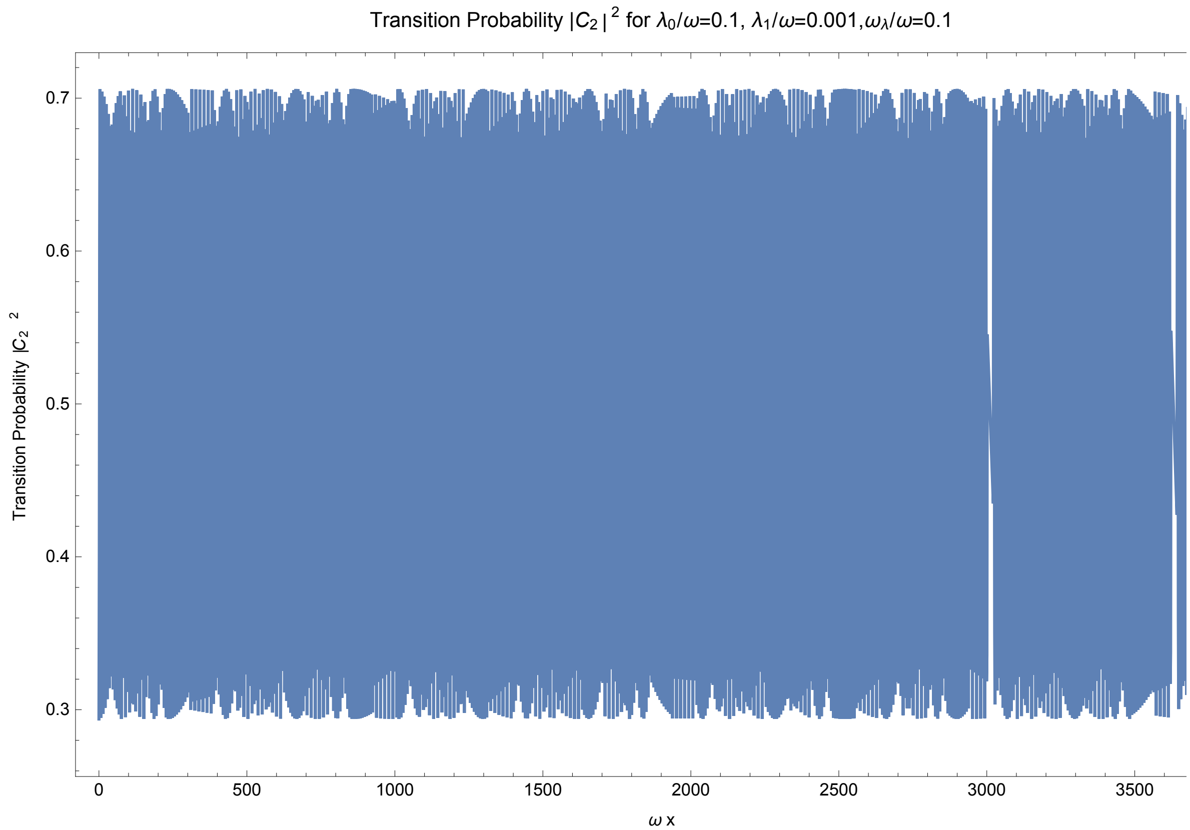
In[240]:=
```
Plot[Evaluate[Abs[waveVac01[x]]^2 + Abs[waveVac02[x]]^2 /.
    {solVac0[[1]], solVac0Convergence[[1]]}], {x, 0, endpoint},
 PlotStyle → {Blue, Red}, PlotRange → All, ImageSize → imgsize, Frame → True]
```

Out[240]=

In[241]:=
```
pltNorm[solVac0, waveVac01, waveVac02, 0, endpoint];
```

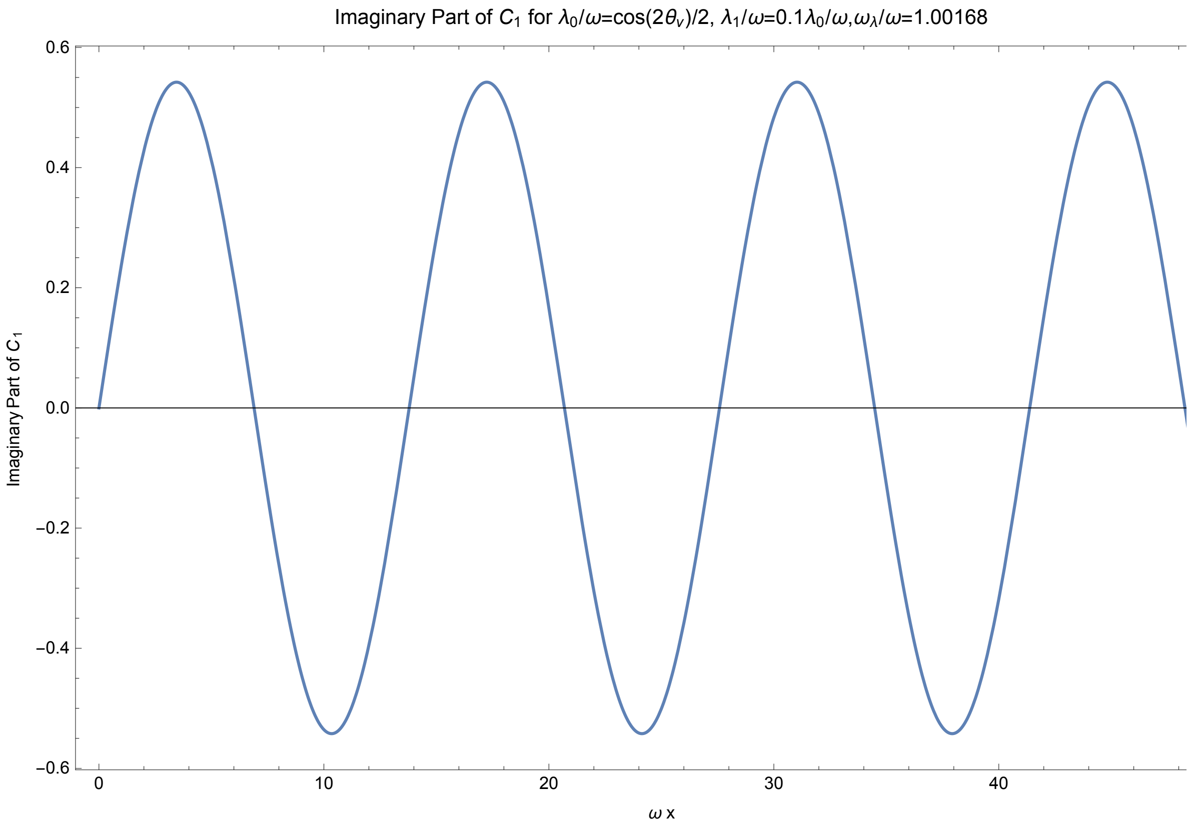
In[242]:=
```
pltProb[solVac0, waveVac02, waveVac01, 0, endpoint,
  "Transition Probability |C₂|² for λ₀/ω=0.1, λ₁/ω=0.001,ωλ/ω=0.1",
  {"ω x", "Transition Probability |C₂|²"}]
```

Out[242]=



In[243]:=
```
Plot[Evaluate[Im[waveVac01[x]]] /. solVac0[[1]],
 {x, 0, 50}, PlotRange → All, PlotLabel →
  "Imaginary Part of C₁ for λ₀/ω=cos(2θᵥ)/2, λ₁/ω=0.1λ₀/ω,ωλ/ω=" <> ToString[β],
 ImageSize → imgsize, Frame → True, FrameLabel → {"ω x", "Imaginary Part of C₁"},
 ImagePadding → imgpadding]
```

Out[243]=



The Imaginary Part

In[244]:=
```
Grid[{{Plot[Evaluate[Im[waveVac01[x]]] /. solVac0[[1]],
    {x, 0, endpoint}, PlotRange → All,
    PlotLabel → "Imaginary Part of C₁ for λ₀/ω=cos(2θᵥ)/2, λ₁/ω=0.1λ₀/ω,ωλ/ω=" <>
      ToString[β], ImageSize → imgsize, Frame → True,
    FrameLabel → {"ω x", "Imaginary Part of C₁"}, ImagePadding → imgpadding],
   Plot[Evaluate[Re[waveVac01[x]]] /. solVac0[[1]],
    {x, 0, endpoint}, PlotRange → All, PlotLabel →
     "Re Part of C₁ for λ₀/ω=cos(2θᵥ)/2, λ₁/ω=0.1λ₀/ω,ωλ/ω=" <> ToString[β],
    ImageSize → imgsize, Frame → True, FrameLabel → {"ω x", "Real Part of C₁²"},
    ImagePadding → imgpadding]},
  {Plot[Evaluate[Im[waveVac02[x]]] /. solVac0[[1]],
    {x, 0, endpoint}, PlotRange → All,
    PlotLabel → "Imaginary Part of C₂ for λ₀/ω=cos(2θᵥ)/2, λ₁/ω=0.1λ₀/ω,ωλ/ω=" <>
      ToString[β], ImageSize → imgsize, Frame → True,
    FrameLabel → {"ω x", "Imaginary Part of C₂"}, ImagePadding → imgpadding],
   Plot[Evaluate[Re[waveVac02[x]]] /. solVac0[[1]],
    {x, 0, endpoint}, PlotRange → All, PlotLabel →
     "Real Part of C₂ for λ₀/ω=cos(2θᵥ)/2, λ₁/ω=0.1λ₀/ω,ωλ/ω=" <> ToString[β],
    ImageSize → imgsize, Frame → True, FrameLabel → {"ω x", "Real Part of C₂"},
    ImagePadding → imgpadding]}}];
```

Solving in Vacuum Basis then Rotate

This problem can be solved in matter basis,hw

In[245]:=
```
eqn1 = I c1'[x] ==
  Cos[2 θv]/2 (α0 + α1 Cos[β x]) c1[x] + Sin[2 θv]/2 (α0 + α1 Cos[β x]) c2[x] Exp[-I x]
eqn2 = I c2'[x] == -Cos[2 θv]/2 (α0 + α1 Cos[β x]) c2[x] +
  Sin[2 θv]/2 (α0 + α1 Cos[β x]) c1[x] Exp[I x]
```

Out[245]= $i\, c1'[x] == 0.0849277\, c1[x] + 0.187753\, e^{-i x}\, c2[x]$

Out[246]= $i\, c2'[x] == 0.187753\, e^{i x}\, c1[x] - 0.0849277\, c2[x]$

In[247]:=
```
solVac = NDSolve[{eqn1, eqn2, c1[0] == initVac[[1]], c2[0] == initVac[[2]]},
  {c1, c2}, {x, 0, endpoint}]
```

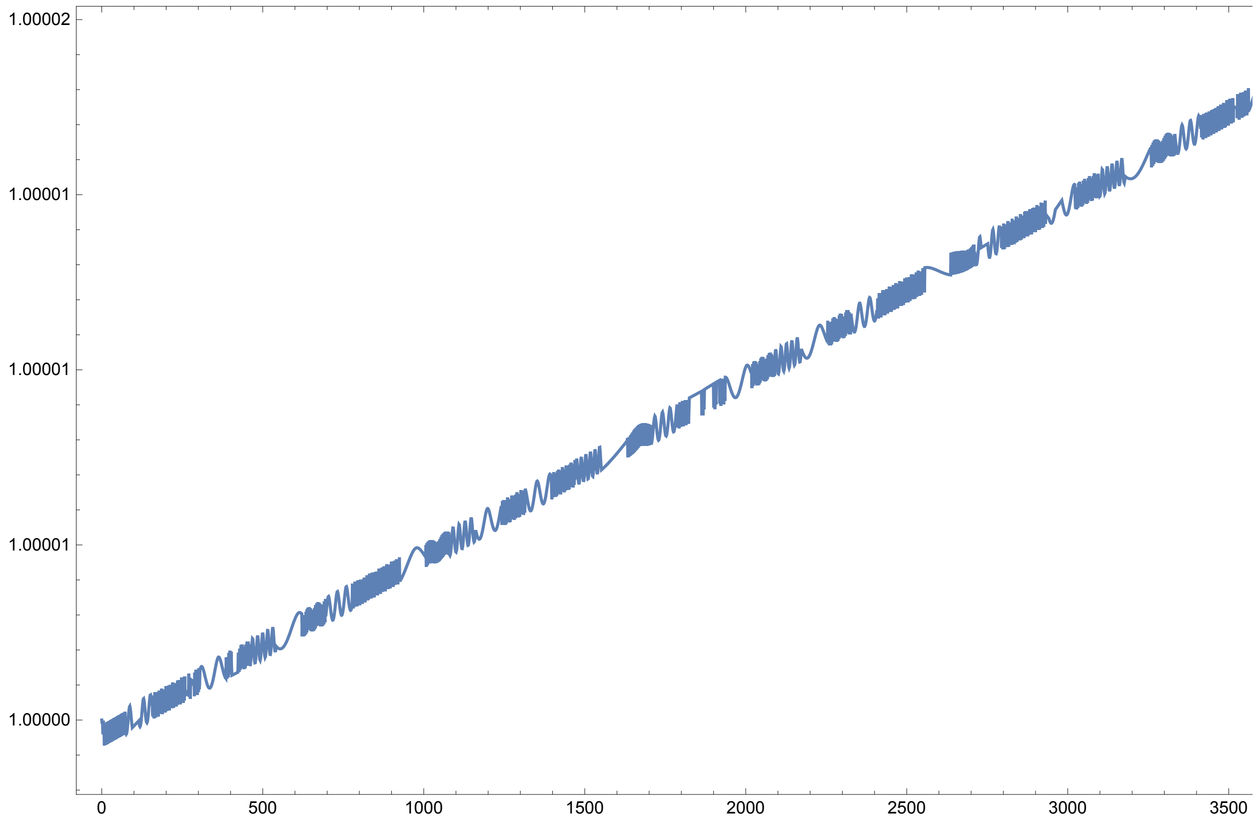
Out[247]= {{c1 → InterpolatingFunction[Domain: {{0., 3.80 × 10³}} Output: scalar],

c2 → InterpolatingFunction[Domain: {{0., 3.80 × 10³}} Output: scalar]}}

Check normalization

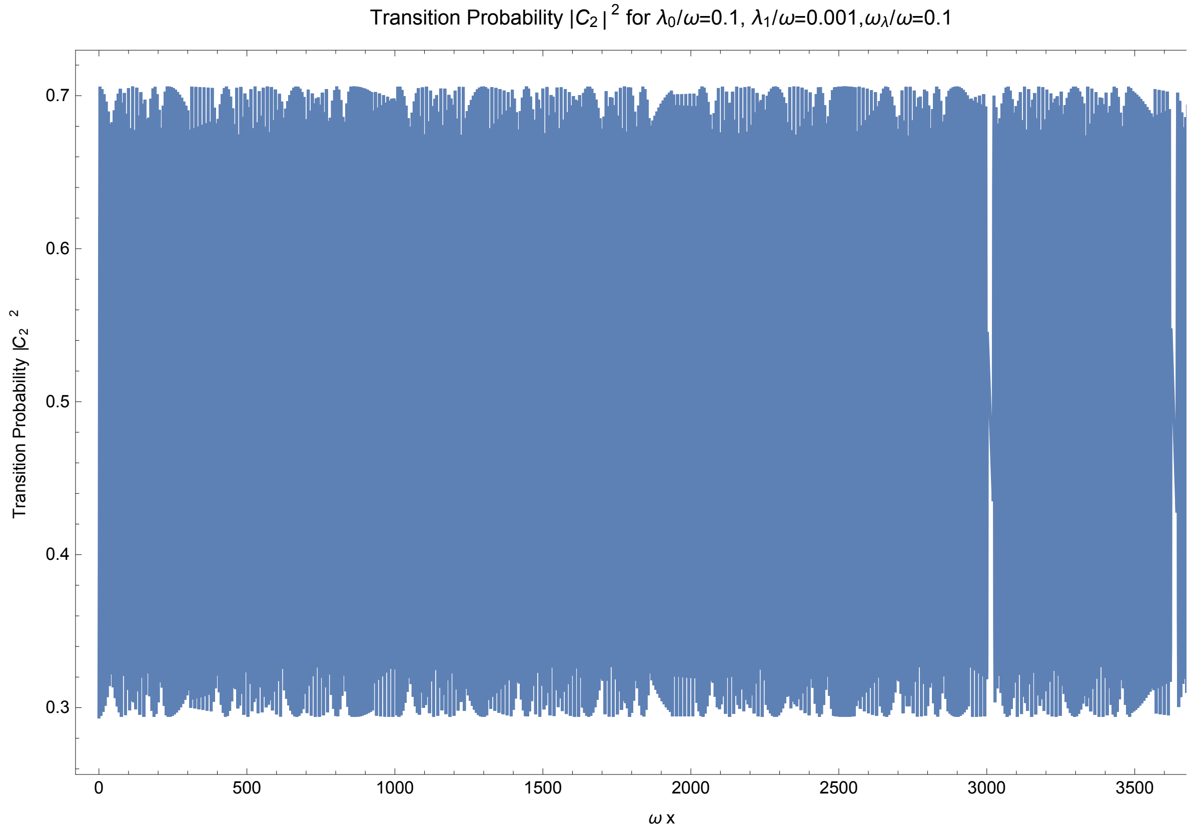
In[248]:=
```
normVac = pltNorm[solVac, c1, c2, 0, endpoint]
```

Out[248]=



In[249]:=
```
pltProb[solVac, c2, c1, 0, endpoint,
 "Transition Probability |C₂|² for λ₀/ω=0.1, λ₁/ω=0.001,ωλ/ω=0.1",
 {"ω x", "Transition Probability |C₂|²"}]
```

Out[249]=



Matter Basis

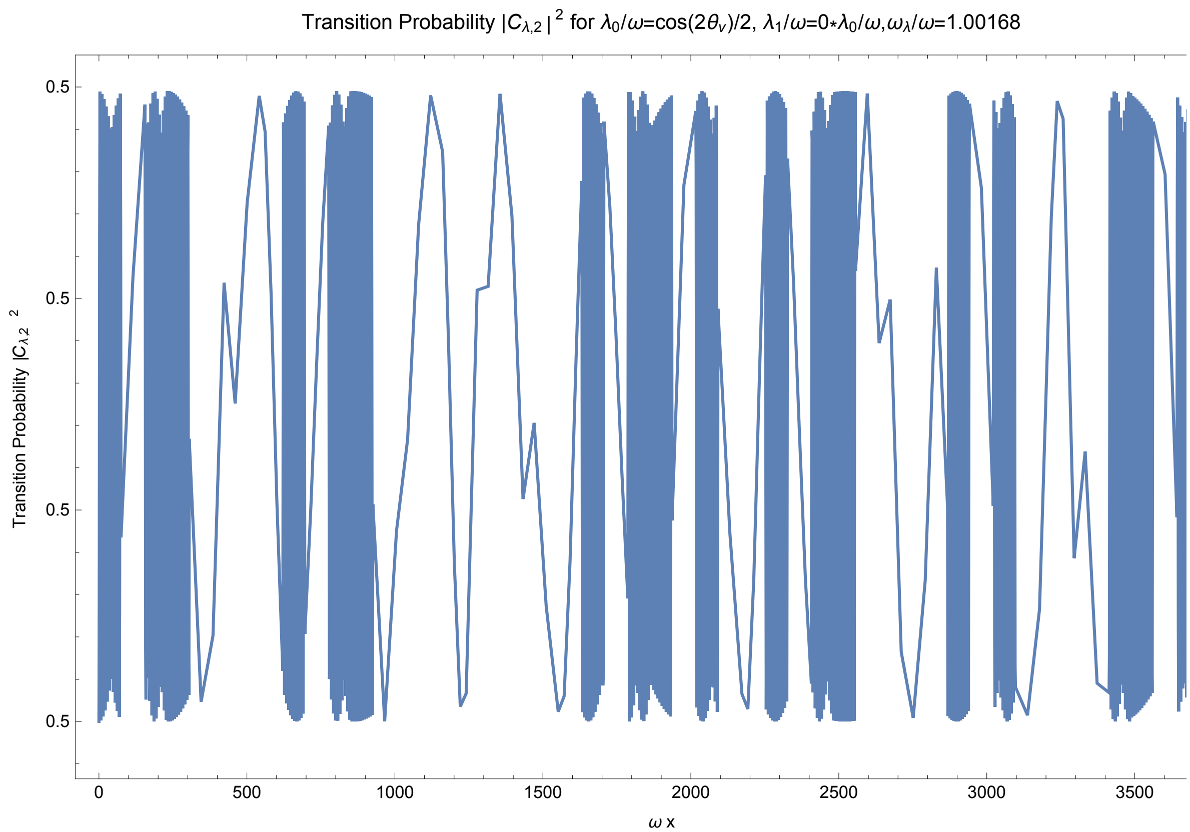
Wavefunction in matter basis is

In[250]:=
```
instWaveFunction[x_] =
 vac2instRotation[α0, 0, θv].{waveVac01[x], waveVac02[x]} /. solVac0[[1]]
```

Out[250]=
```
{-0.210805 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x] +
  0.977528 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x],
 0.977528 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x] +
  0.210805 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x]}
```

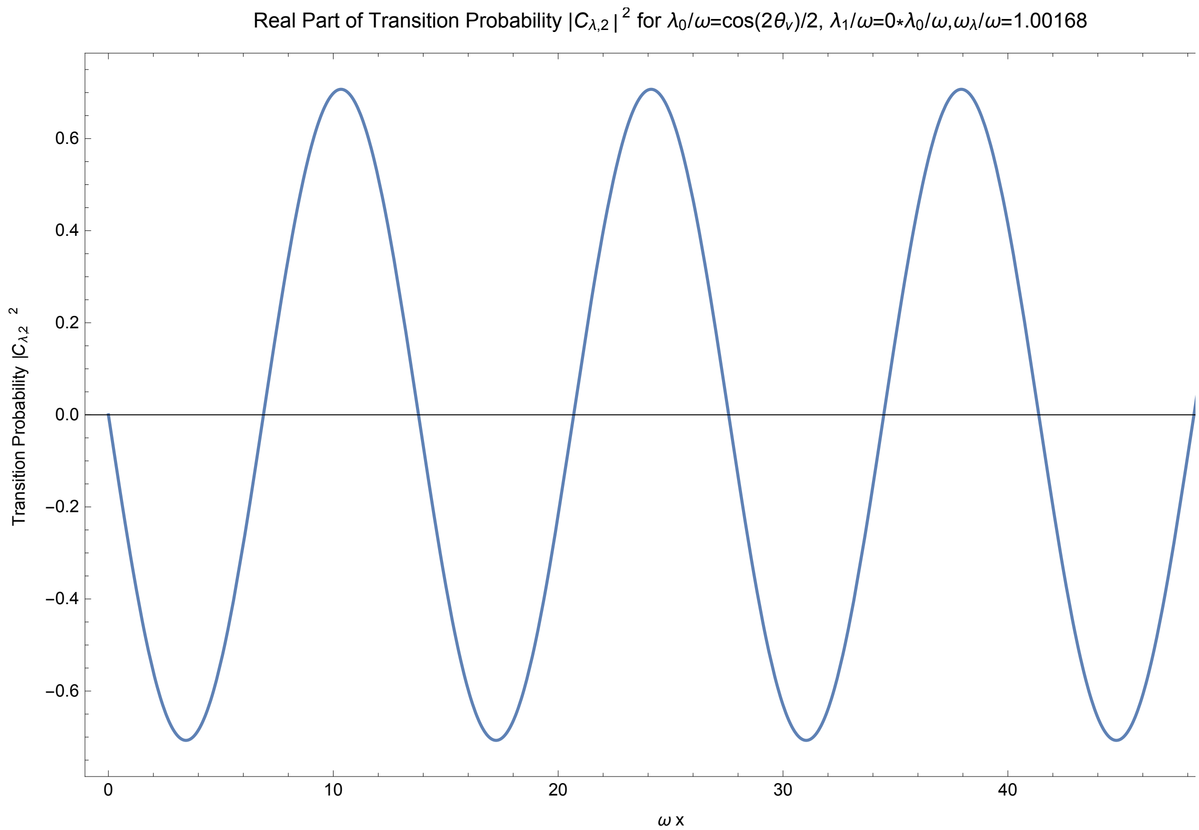
In[251]:=
```
probInstHeavy = Plot[Abs[instWaveFunction[x][[2]]]^2 /
    (Abs[instWaveFunction[x][[1]]]^2 + Abs[instWaveFunction[x][[2]]]^2),
   {x, 0, endpoint}, PlotRange → All,
   PlotLabel → "Transition Probability |Cλ,2|² for λ0/ω=cos(2θv)/2, λ1/ω=" <>
     ToString[α1] <> "*λ0/ω,ωλ/ω=" <> ToString[β], ImageSize → imgsize, Frame → True,
   FrameLabel → {"ω x", "Transition Probability |Cλ,2|²"}, ImagePadding → imgpadding]
```

Out[251]=



In[252]:=
```
instHeavyRe =
  Plot[Im[instWaveFunction[x][[2]]], {x, 0, 50}, PlotRange → All, PlotLabel →
    "Real Part of Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=" <>
     ToString[α1] <> "*λ0/ω,ωλ/ω=" <> ToString[β], ImageSize → imgsize, Frame → True,
   FrameLabel → {"ω x", "Transition Probability |Cλ,2|^2"}, ImageePadding → imgpadding]
```

Out[252]=



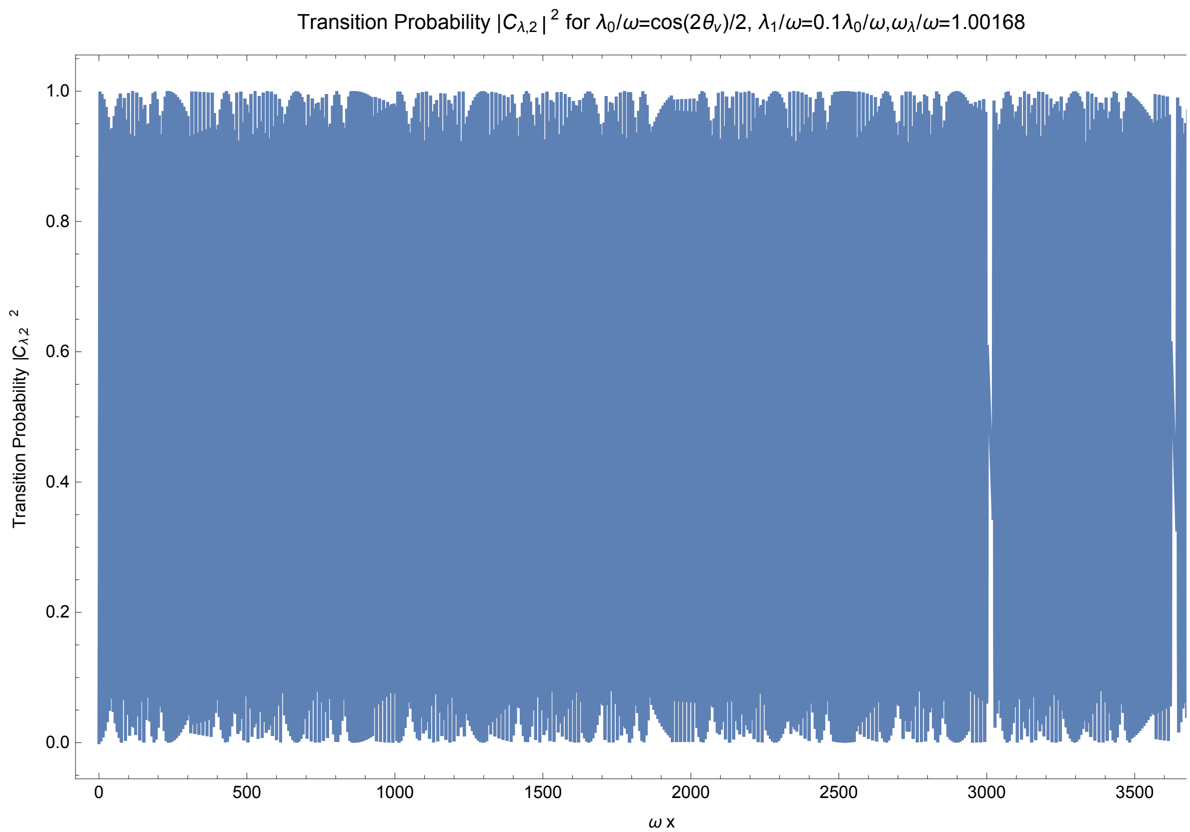
Flavor Basis

In[253]:=
```
instWaveFunctionf[x_] = {{Cos[θv], Sin[θv]}, {-Sin[θv], Cos[θv]}}.
   {waveVac01[x], waveVac02[x]} /. solVac0[[1]]
```

Out[253]=
```
{0.542155 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x] +
   0.840278 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x],
  0.840278 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x] -
   0.542155 InterpolatingFunction[ Domain: {{0., 3.80×10^3}} Output: scalar ][x]}
```

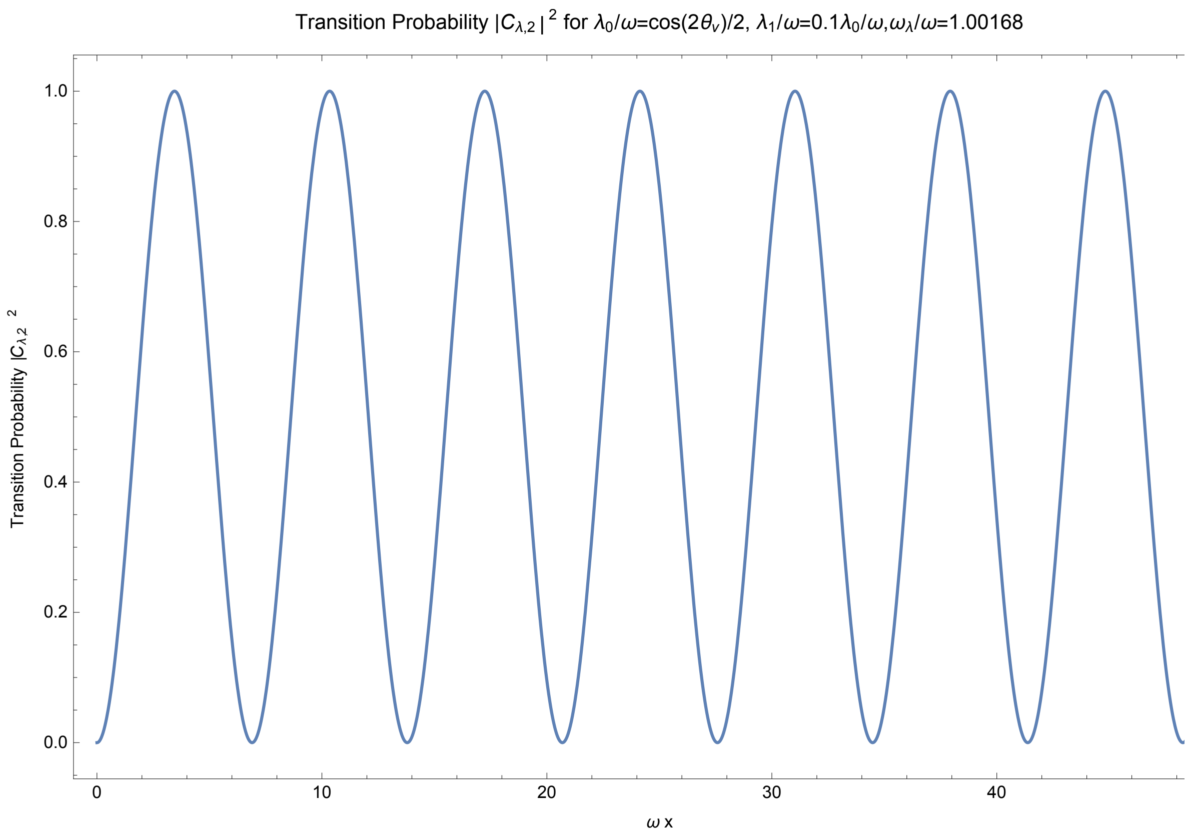
In[254]:=
```
probf = Plot[Abs[instWaveFunctionf[x][[2]]]^2 /
    (Abs[instWaveFunctionf[x][[2]]]^2 + Abs[instWaveFunctionf[x][[1]]]^2),
   {x, 0, endpoint}, PlotRange → All, PlotLabel →
    "Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=0.1λ0/ω,ωλ/ω=" <>
     ToString[β], ImageSize → imgsize, Frame → True,
   FrameLabel → {"ω x", "Transition Probability |Cλ,2|^2"}, ImagePadding → imgpadding]
```

Out[254]=



In[255]:=
```
Plot[Abs[instWaveFunctionf[x][[2]]]^2/
   (Abs[instWaveFunctionf[x][[2]]]^2 + Abs[instWaveFunctionf[x][[1]]]^2),
  {x, 0, 50}, PlotRange → All, PlotLabel →
   "Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=0.1λ0/ω,ωλ/ω=" <>
    ToString[β], ImageSize → imgsize, Frame → True,
  FrameLabel → {"ω x", "Transition Probability |Cλ,2|^2"}, ImagePadding → imgpadding]
```

Out[255]=



Calculate in Matter Basis Directly

The calculation can be done in averaged matter basis directly.

Hamiltonian in (averaged) Matter basis is

In[256]:=
```
hamilAM[alpha0_, alpha1_, beta_, x_] :=
 vac2instRotation[α0, 0, θv].hamilVac[alpha0, alpha1, beta, x].
   Transpose[vac2instRotation[α0, 0, θv]] // FullSimplify
```

Test this function

In[257]:=
```
hamilAM[α0, α1, β, x]
```

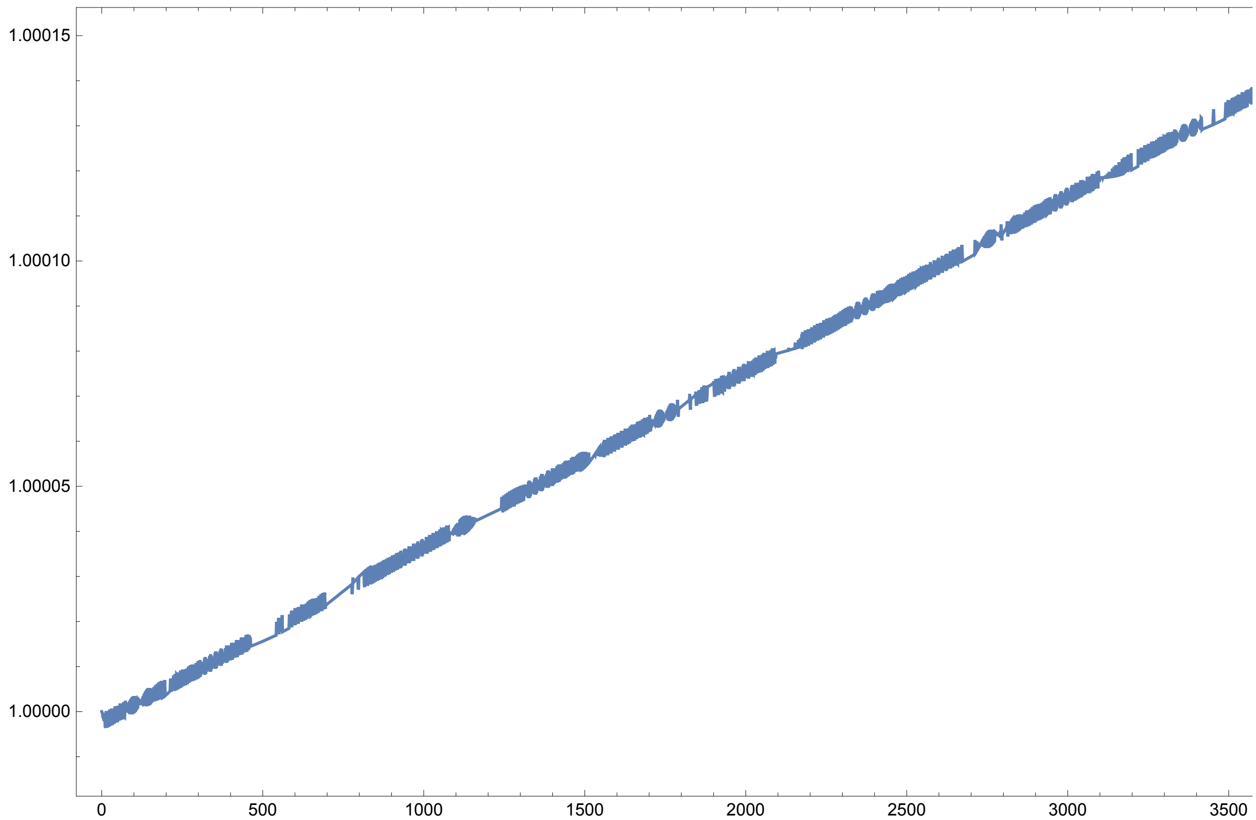
Out[257]= $\{\{-0.455561, 6.78839\times10^{-9}\}, \{6.78839\times10^{-9}, 0.455561\}\}$

System to be solved is the Schrodinger equation

In[258]:=
```
waveAM[x_] = {waveAM1[x], waveAM2[x]};
```

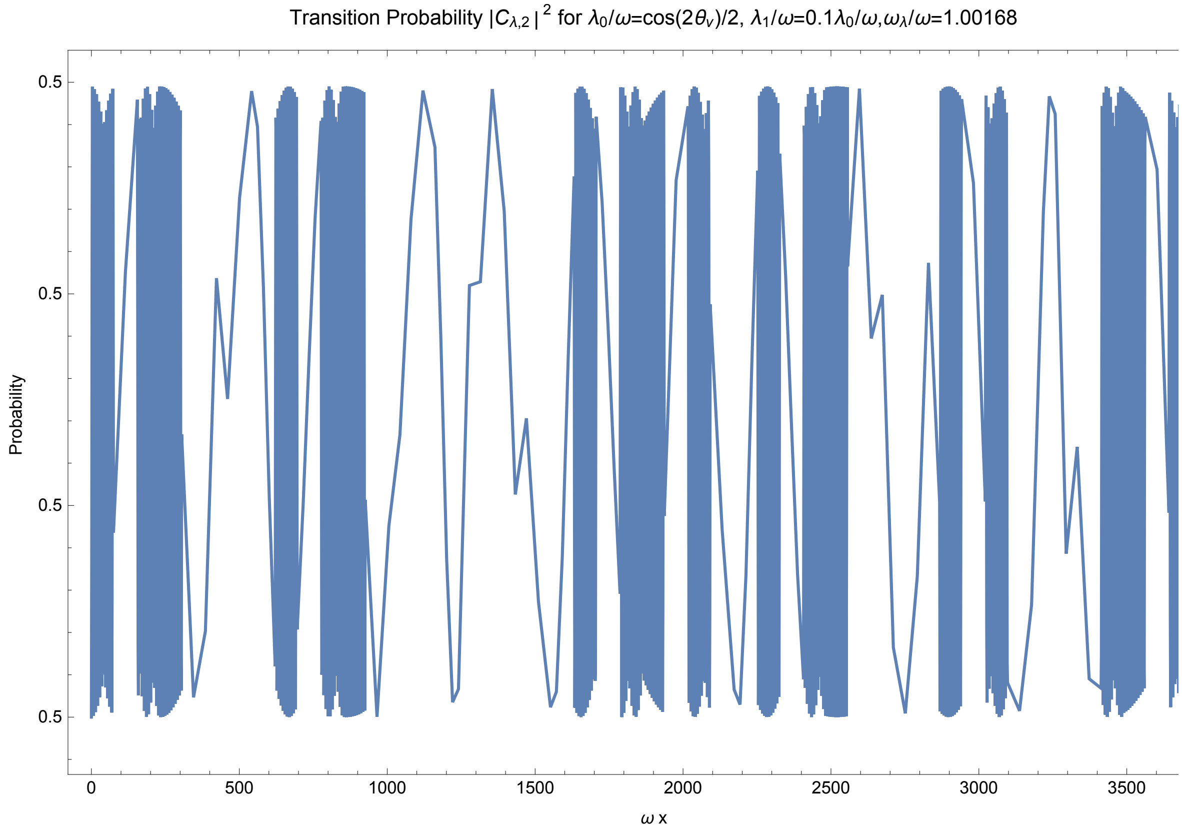
```
In[259]:= solAM = NDSolve[{I D[waveAM[x], x] == hamilAM[α0, α1, β, x].waveAM[x],
     waveAM[0] == initAM}, waveAM[x], {x, 0, endpoint}];

In[260]:= pltNorm[solAM, waveAM1, waveAM2, 0, endpoint]
```

Out[260]=



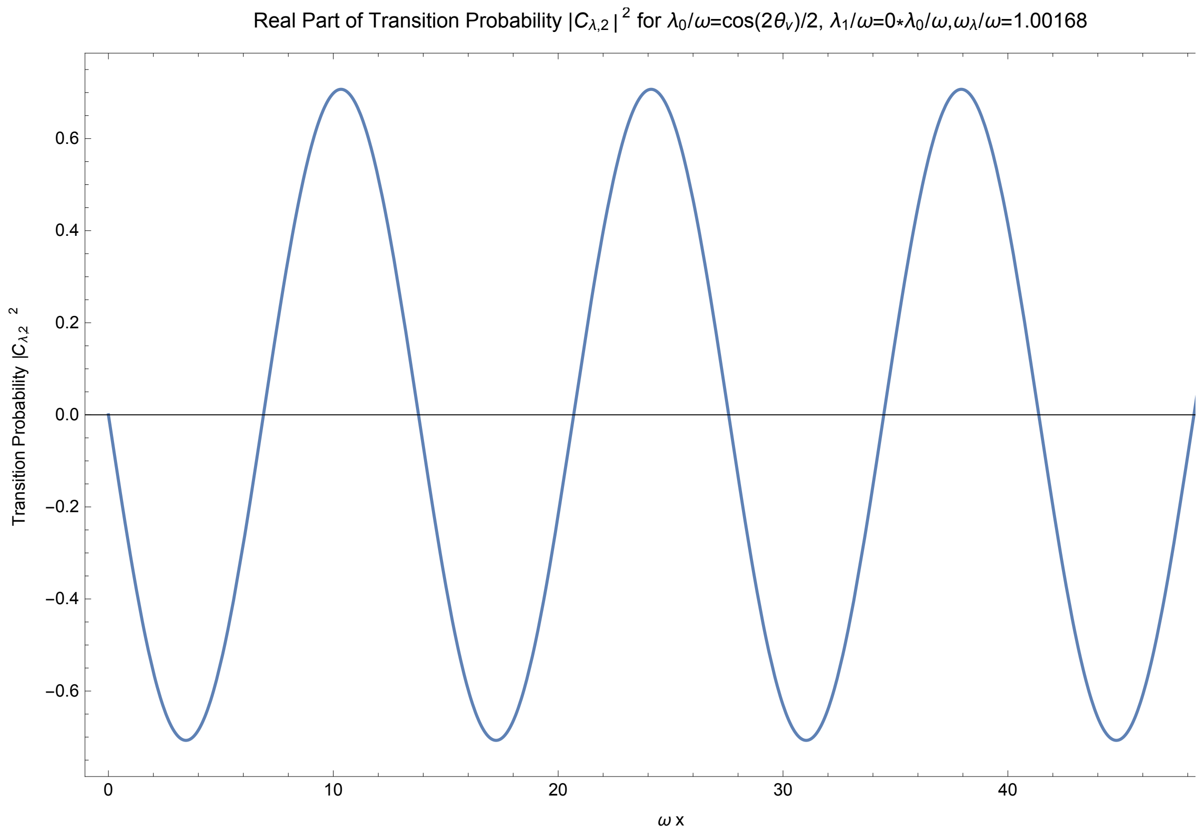
In[261]:=
```
pltProb112 = pltProb[solAM, waveAM2, waveAM1, 0, endpoint,
  "Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=0.1λ0/ω,ωλ/ω=" <>
   ToString[β], {"ω x", "Probability"}]
```

Out[261]=



In[262]:=
```
instHeavy2Re =
 Plot[Im[waveAM[x][[2]]] /. solAM, {x, 0, 50}, PlotRange → All, PlotLabel →
   "Real Part of Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=" <>
    ToString[α1] <> "*λ0/ω,ωλ/ω=" <> ToString[β], ImageSize → imgsize, Frame → True,
  FrameLabel → {"ω x", "Transition Probability |Cλ,2|^2"}, ImagePadding → imgpadding]
```

Out[262]=



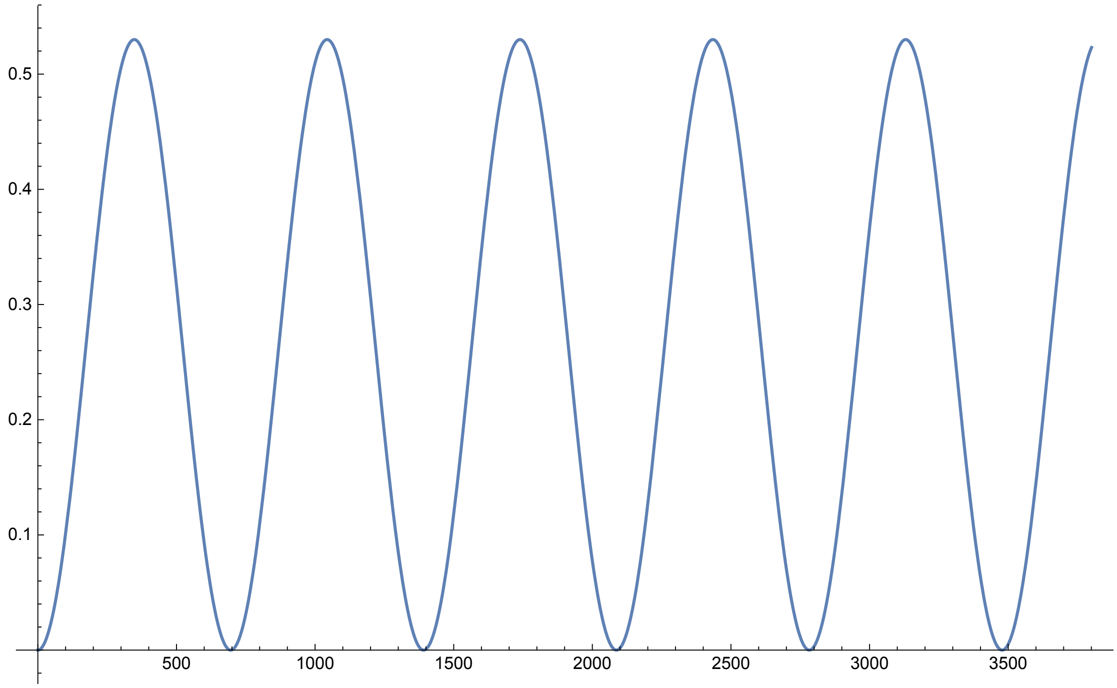
In[263]:=
```
Sin[2 Pi x]
```

Out[263]= Sin[2 π x]

In[280]:=

```
Plot[0.53 Sin[2 Pi (2 Pi x)/omegav km2eV[2.3 * 10^4]]^2, {x, 0, endpoint}, ImageSize → Large]
```

Out[280]=



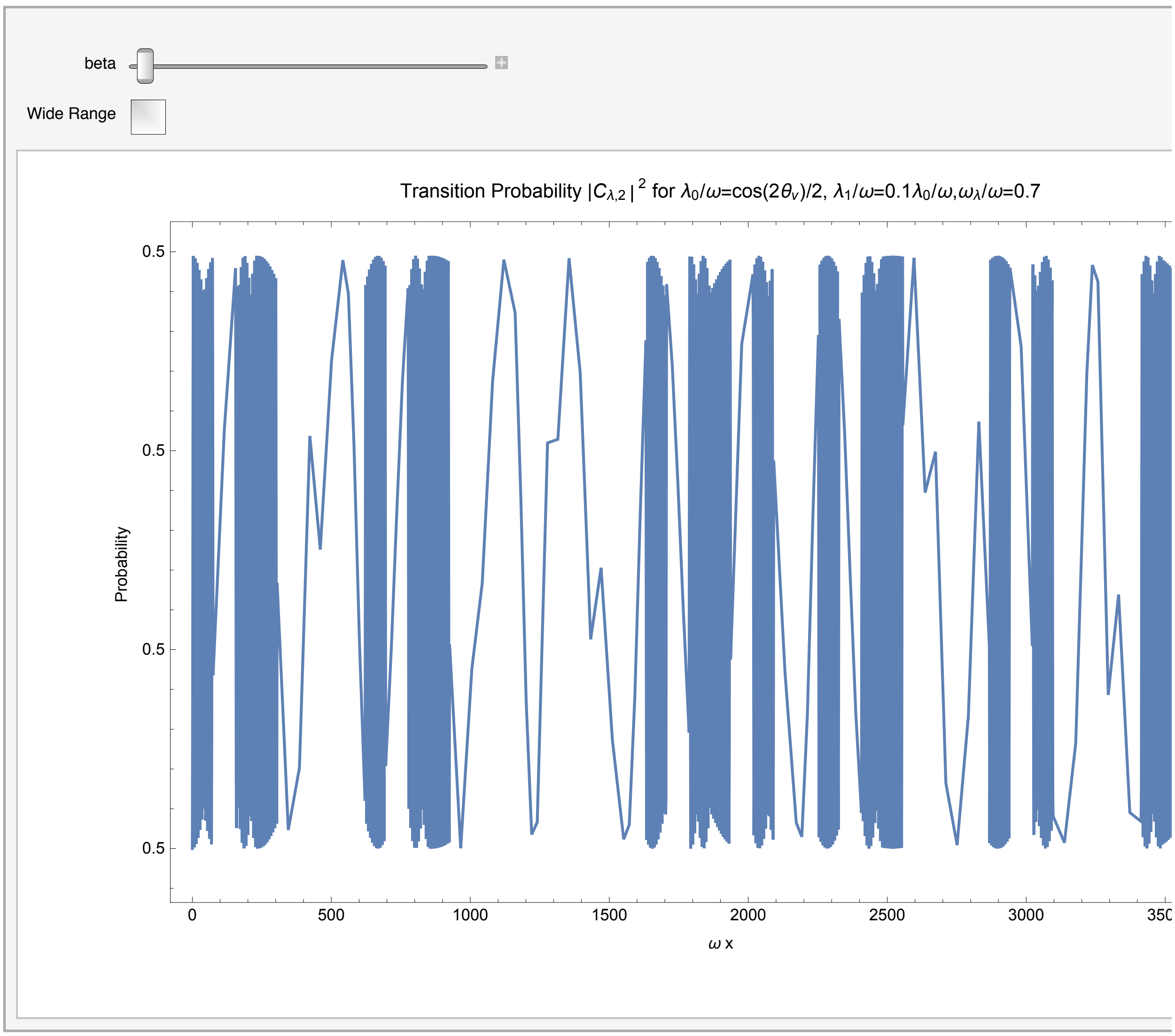
Scanning Parameters

In[265]:=
```
testrangestart = 0
testrangeend = endpoint
Manipulate[pltProb[NDSolve[{I D[waveAM[x], x] == hamilAM[α0, α1, beta, x].waveAM[x],
     waveAM[0] == initAM}, waveAM[x], {x, testrangestart, testrangeend}],
   waveAM2, waveAM1, testrangestart, testrangeend,
   "Transition Probability |Cλ,2|^2 for λ0/ω=cos(2θv)/2, λ1/ω=0.1λ0/ω,ωλ/ω=" <>
    ToString[beta], {"ω x", "Probability"}],
  {beta, 0.7, If[wide, 5, 1.0], 0.01}, {{wide, False, "Wide Range"}, {False, True}}]
```

Out[265]= 0

Out[266]= 3800.74

Out[267]=



Stimulated?

Temp