```
In[•]:= x=.; Remove["Global`*"];
        Off[General::spell1];
        ClearAll
Out[•]=
        ClearAll
```

```
In[•]:= conjugateRule={Complex[re_,im_]:>Complex[re,-im]};
        conjugate[z_]:=z/.conjugateRule;
        real[z_]:=(1/2)*(z+conjugate[z])//Simplify;
        imag[z_]:=(-I/2)*(z-conjugate[z])//Simplify;
        abs[z_]:=Sqrt[(z*(z/.conjugateRule))]//Simplify;
```

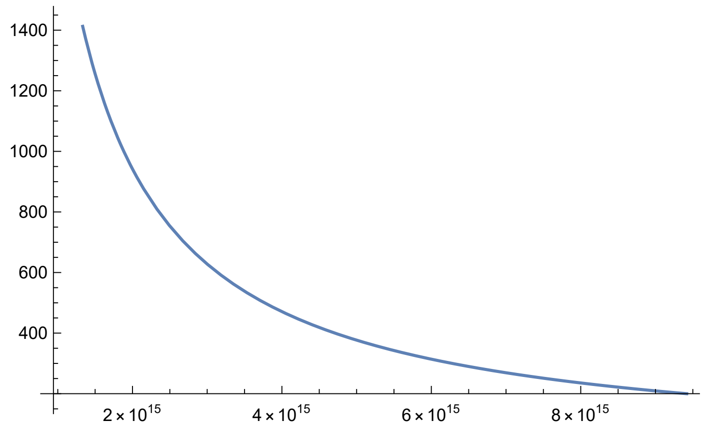
```
In[•]:= (*SetDirectory["D:\\Cuba\\Cuba-4.2"]
        Install["Vegas"];
        Install["Divonne"];
        Install["Suave"];
        Install["Cuhre"];
```

Constants

```
In[•]:= c = 299792458 ;
        hb = 1.0545715964207855 * 10^-34;
        q = 1.602 * 10^-19;
        ϵ0 = 8.85 * 10^-12;
        m = 9.1 * 10^-31;
        a0 = 3.5668 * 10^-10;
        den = 8 / a0^3;
        den1 = 2 / a0^3;
        G = Sqrt[2^2 + 2^2 + 0^2] * 2 * Pi / a0;
        normstructurefactor = 8 / 8;
        atomscatfactor = 1.603;
        re = 2.82 * 10^-13 * 10^-2;
        ϵ0 = 8.85 * 10^-12;
        m = 9.1 * 10^-31;
```

```
In[•]:= λv[w_] := 2 * Pi * c / w * 10^9
```

```
In[•]:= Plot[λv[w], {w, 2 * Pi * c / 2000 * 10^9, 2 * Pi * c / 200 * 10^9}]
Out[•]=
```



Scattering Factors (Carbon)

Note : Carbondata are hidden in the following cell.

In[ ]:=
```
listf1 = carbondata[[All, {2, 3}]];
listf2 = carbondata[[All, {2, 4}]];
f1[w_] := Interpolation[listf1, InterpolationOrder → 4][w]
f2[w_] := Interpolation[listf2, InterpolationOrder → 4][w]
```

In[ ]:=

Refractive index and loss

In[ ]:=
```
n[w_] := 1 - den * re * 2 * Pi * c^2 / w^2 * f1[w];
α[w_] := Which[w > 10^17, w / c * den * re * 2 * Pi * c^2 / w^2 * f2[w], w < 10^16, 10^-2];
```

In[ ]:=
```
α[w_] := 0.00001;
```

In[ ]:=
```
α0[w_] := w / c * den * re * 2 * Pi * c^2 / w^2 * f2[w];
```

In[ ]:=

In[ ]:=

In[ ]:=
```
nvisible[w_] :=
 Sqrt[0.3306 * λv[w]^2 / (λv[w]^2 - 175^2) + 4.3356 * λv[w]^2 / (λv[w]^2 - 106^2) + 1]
```

In[ ]:=
```
nvisible[10^15]
```

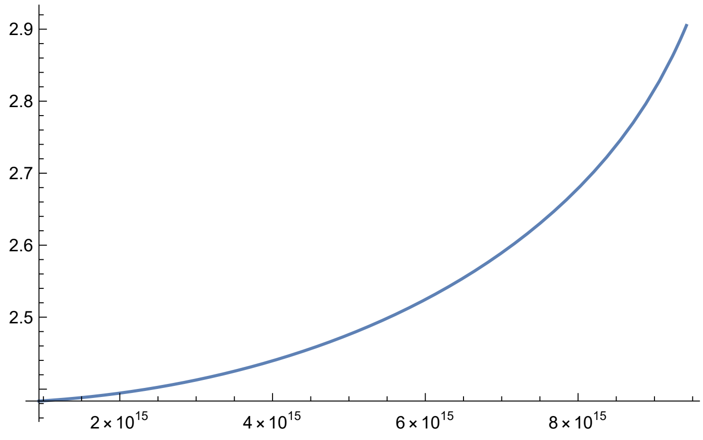
Out[ ]=
```
2.38387
```

In[ ]:=

In[ ]:=

In[ ]:=
```
Plot[nvisible[w], {w, 2 * Pi * c / 2000 * 10^9, 2 * Pi * c / 200 * 10^9}]
```

Out[ ]=



In[ ]:=

```
In[•]:= α[wi0]
Out[•]=
        0.00001

In[•]:=
```

Parameters

```
In[•]:= wp0 = 9 * 10^3 * q / hb;

In[•]:= N[20 * (1 + 0.9 * 10^-4)]
Out[•]=
        20.0018

In[•]:= λ = 619.92097 * 10^-9; (*2 eV*)
        (*λ=563.56451*10^-9; (*2.22 eV*)*)
        (*λ=642*10^-9; (*1.931 eV*) *)

In[•]:= wi0 = N[2 * Pi * c / λ]
Out[•]=
        3.03854 × 10^15

In[•]:=
        Δ = 0.3665 * 10^-3; (*offset from Bragg, mrad [21 mdeg] *)
        (*Δ =0.3982*10^-3; (*offset from Bragg, mrad [27 mdeg] *) *)
        (*Δ =0.2443*10^-3; (*offset from Bragg, mrad [14.0166 mdeg] *)*)
        ws0 = wp0 - wi0



        kp0 = wp0 / c * n[wp0]; ks0 = ws0 / c * n[ws0]; ki0 = wi0 * nvisible[wi0] / c;
        L = 1 * 10^-4;
Out[•]=
        1.36689 × 10^19

In[•]:= ΔEp = 1; ΔEi = 1;
```

$$\frac{n[wp0 + q * (\Delta Ep / hb)] * (wp0 + q * (\Delta Ep / hb))}{c} - \frac{n[wp0] * (wp0)}{c}$$

$$\frac{nvisible[wi0 + q * (\Delta Ei / hb)] * (wi0 + q * (\Delta Ei / hb))}{c} - \frac{nvisible[wi0] * (wi0)}{c}$$

```
Out[•]=
        5.06722 × 10^6
Out[•]=
        1.29112 × 10^7

In[•]:= Es0 = ws0 * hb / q
Out[•]=
        8998.

In[•]:=
```

In[ ]:=

In[ ]:=
```
branch = 2; (*Take first or second PM solution*)
```

Find Base values

In[ ]:=
```
θB = Abs[ArcSin[G / (2 * kp0)]];

θB
```

Out[ ]=
```
0.577911
```

In[ ]:=
```
kpvec1 = {kp0 * Sin[(θp0)], kp0 * Cos[(θp0)]};
ksvec1 = {ks0 * Sin[θs], ks0 * Cos[θs]};
kivec1 = {ki0 * Sin[θi], ki0 * Cos[θi]};
vecG1 = {G, 0};
```

In[ ]:=
```
eq1 = kp0 * Sin[(θp0)] + G - ks0 * Sin[θs] - ki0 * Sin[θi]
```

Out[ ]=

$$2.48984\times 10^{10} - 2.44628\times 10^{7}\,\mathrm{Sin}[\theta i] - 4.5594\times 10^{10}\,\mathrm{Sin}[\theta s]$$

In[ ]:=
```
eq2 = kp0 * Cos[(θp0)] - ks0 * Cos[θs] - ki0 * Cos[θi]
```

Out[ ]=

$$3.81892\times 10^{10} - 2.44628\times 10^{7}\,\mathrm{Cos}[\theta i] - 4.5594\times 10^{10}\,\mathrm{Cos}[\theta s]$$

In[ ]:=
```
sol = NSolve[{eq1 == 0, eq2 == 0}, {θs, θi}]
```

NSolve: Inverse functions are being used by NSolve, so some solutions may not be found; use Reduce for complete solution information.

Out[ ]=
```
{{θs → 0.577239, θi → 2.36085}, {θs → 0.578288, θi → -1.20532}}
```

In[ ]:=
```
θs0 = θs /. sol[[branch, 1]]
```

Out[ ]=
```
0.578288
```

In[ ]:=
```
θi0 = θi /. sol[[branch, 2]]
```

Out[ ]=
```
-1.20532
```

In[ ]:=
```
θi01 = ArcSin[Sin[θi0] * nvisible[wi0]]
```

Out[ ]=
```
-1.5708 + 1.45265 i
```

In[ ]:=
```
detector1 = θs0 - θp0 + Δ
```

Out[ ]=
```
1.15693
```

In[ ]:=
```
θi01out = ArcSin[Sin[θi0] * nvisible[wi0]]
```

Out[ ]=
```
-1.5708 + 1.45265 i
```

In[•]:=
```
detector2 = θi01out + θp0
```
Out[•]=
```
-2.14907 + 1.45265 i
```

Relation between θi, θs and θiy, θsy

In[•]:=

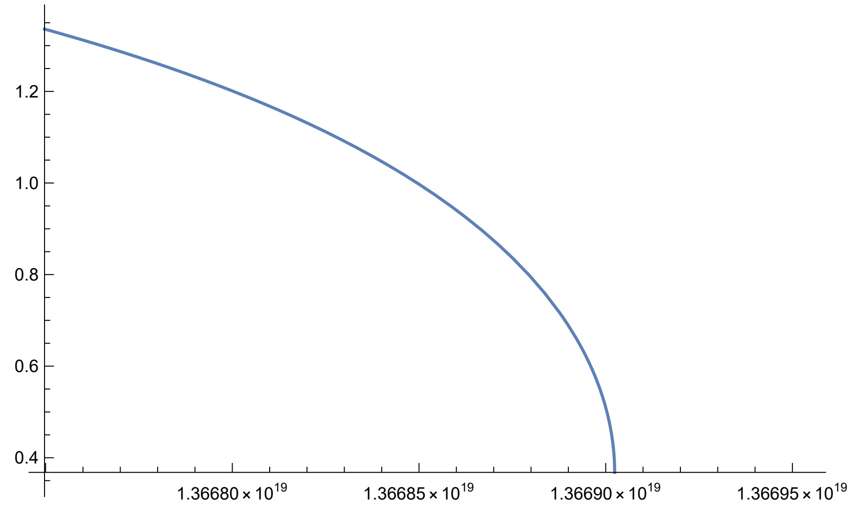
In[•]:=
```
A1 = kp0 * Sin[θp0 + δp] + G;
```
In[•]:=
```
A2 = ws / c * n[ws] * Sin[θs0 + qq];
```
In[•]:=
```
A3 = (wp0 - ws) / c * nvisible[wp0 - ws];
```
In[•]:=
```
δθi[ws_, qq_, δp_] = ArcSin[(A1 - A2) / A3] + θi0 + Pi;
```
In[•]:=
```
δθi[ws0, 0, 0]
```
Out[•]=
```
0.730949
```
In[•]:=
```
δix[ws_, qq_, δp_] = δθi[ws, qq, δp];
```
In[•]:=

In[•]:=

In[•]:=

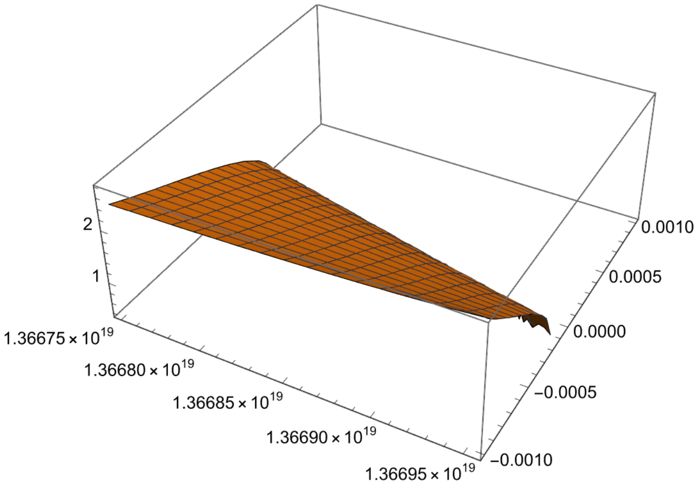
In[•]:=
```
58.7713 - 58.90511051585671`
```
Out[•]=
```
-0.133811
```
In[•]:=

In[•]:=
```
Plot[δix[ws, 0, 0], {ws, .9999 * ws0, 1.00005 * ws0}]
```
Out[•]=



In[•]:=

In[•]:=
```
Plot3D[δix[ws, qq, 0], {ws, .9999 * ws0, 1.00005 * ws0}, {qq, -0.001, 0.001}]
```
Out[•]=



In[•]:=
```
δiy[ws_, qqy_] = -ArcSin[ws * n[ws] * Sin[qqy] / ((wp0 - ws) * nvisible[wp0 - ws])];
```

In[•]:=
```
δiy[ws0, 0]
```
Out[•]=
```
0.
```

CALCULATE relations between apertures

In[•]:=

In[•]:=
```
δkxlim = wp0 * n[wp0] * Sin[θp0] / c + G -
  ws0 * n[ws0] * Sin[θs0 + θslim] / c - wi0 * nvisible[wi0] * Sin[θi0 + qq] / c
```
Out[•]=
```
2.48984 × 10^10 + 2.44628 × 10^7 Sin[1.20532 - qq] - 4.5594 × 10^10 Sin[0.578288 + θslim]
```

In[•]:=
```
sollim = Solve[δkxlim == 0, θslim];
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

In[•]:=
```
θslim[qq_] = Abs[θslim /. sollim[[1, 1]]] // FullSimplify
```
Out[•]=
```
Abs[0.578288 - 1. ArcSin[0.54609 + 0.000536536 Sin[1.20532 - 1. qq]]]
```

In[•]:=
```
θslim[3 * 10^-3] * 10^3
```
Out[•]=
```
0.000689654
```

In[•]:=
```
δkylim = ws0 * n[ws0] * Sin[θslimy] - wi0 * nvisible[wi0] * Sin[qqy];
```

In[•]:=
```
sollimy = Solve[δkylim == 0, θslimy];
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

In[ ]:=
```
θslimy[qqy_] = θslimy /. sollimy[[1, 1]] // FullSimplify
```
Out[ ]=
```
ArcSin[0.000536536 Sin[qqy]]
```

In[ ]:=
```
θslimy[1 * 10^-3] * 10^3
```
Out[ ]=
```
0.000536536
```

Phase Matching ; From Laue; Deltak; Signal and Idler

In[ ]:=
```
Δkzsig[ws_, δsx_, δsy_, δp_] =
  wp0 * n[wp0] * Cos[θp0 + δp] / c - ws * n[ws] * Cos[θs0 + δsx] * Cos[δsy] / c -
   (wp0 - ws) * nvisible[wp0 - ws] * Cos[θi0 + δθi[ws, δsx, δp]] * Cos[δiy[ws, δsy]] / c;
```

In[ ]:=
```
Δkzsig2[ws_, δsx_, δsy_, δp_] =
  wp0 * n[wp0] * Cos[θp0 + δp] / c - ws * n[ws] * Cos[θs0 + δsx] * Cos[δsy] / c -
   (wp0 - ws0) * nvisible[wp0 - ws0] * Cos[θi0 + δθi[ws, δsx, δp]] * Cos[δiy[ws, δsy]] / c;
```

In[ ]:=
```
δθi[ws0, 0, 0]
```
Out[ ]=
```
0.730949
```

In[ ]:=
```
δiy[ws0, 0]
```
Out[ ]=
```
0.
```

In[ ]:=

In[ ]:=

In[ ]:=
```
Δkzsig[ws0, 0, 0, 0] * L
```
Out[ ]=
```
-1301.88
```

Sensitivity

In[ ]:=
```
Δkzsig[ws0, 10^-4, 0, 0]
```
Out[ ]=

$-6.56583 \times 10^{6} - 7.90044 \times 10^{6}\, i$

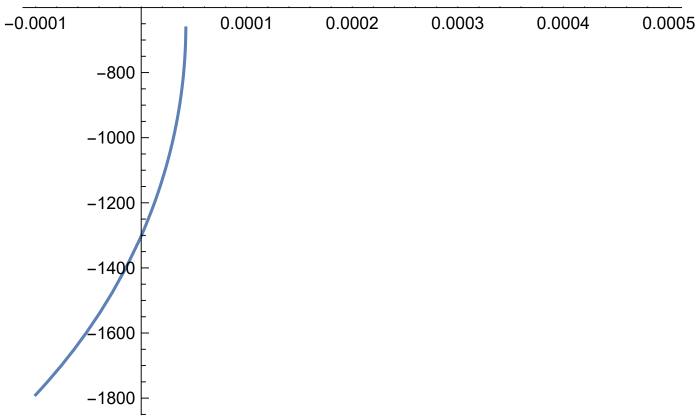
In[ ]:=
```
Δkzsig[ws0, 10^-4, 10^-4, 0]
```
Out[ ]=

$-6.25372 \times 10^{6} - 7.762 \times 10^{6}\, i$

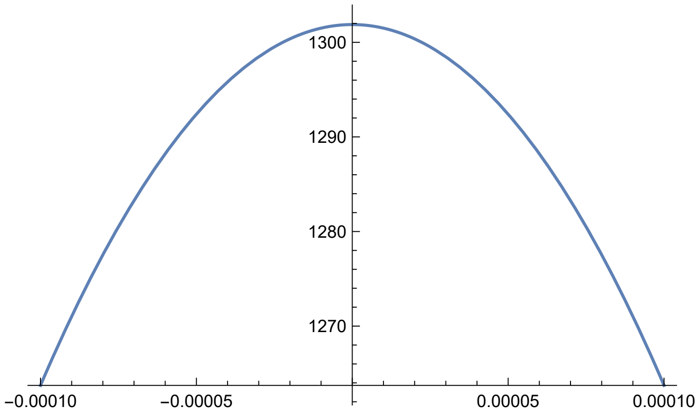
In[ ]:=
```
Δkzsig[1.00001 * ws0, 0, 0, 0]
```
Out[ ]=

-9.52667×10^{6}

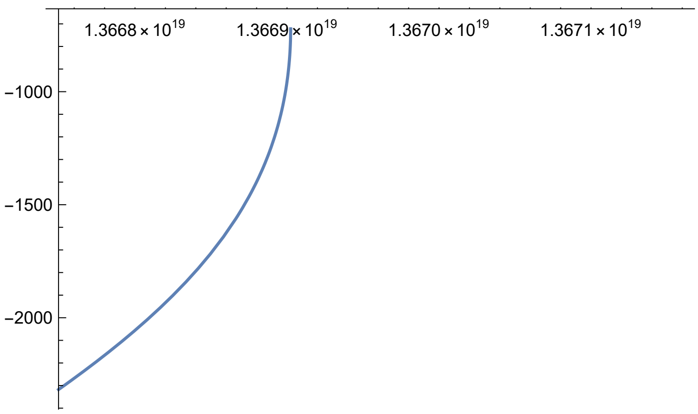
In[•]:=
```
p1 = Plot[L * Δkzsig[ws0, δsx, 0, 0], {δsx, -10^-4, 5 * 10^-4}]
```
Out[•]=



In[•]:=
```
Plot[Abs[L * Δkzsig[ws0, 0, δsy, 0]], {δsy, -10^-4, 10^-4}]
```
Out[•]=

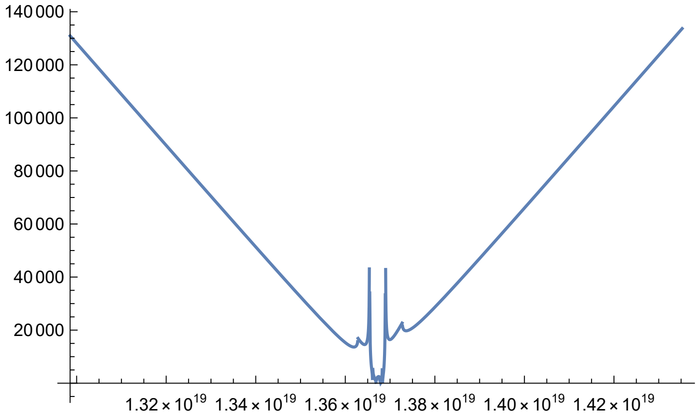


In[•]:=
```
Plot[L * Δkzsig[ws, 0, 0, 0], {ws, .9999 * ws0, 1.0002 * ws0}]
```
Out[•]=



In[∘]:=
```
Plot[Abs[L * Δkzsig[ws, 0, 0, 0] - L * Δkzsig[ws0, 0, 0, 0]], {ws, .95 * ws0, 1.05 * ws0}]
```

Out[∘]=



Nonlinearity

In[∘]:=
```
η = Sqrt[u0 / ϵ0];
```

In[∘]:=
```
J = I * q * (n^2 - 1) * normstructurefactor * G * ϵ0 * Cos[θi0 + δix[ws, δsx, δp]] *
    Cos[2 * θB] / (2 * m * ws0) /. {n → nvisible[λv[wp0 - ws]]};
```

In[∘]:=
```
deff = -I * J / (2 * ϵ0 * ws0)
```

Out[∘]=

$$\left(4.7318\times10^{-18}+0.\,i\right)\left(\frac{0.3306\,\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}{-30\,625+\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}+\frac{4.3356\,\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}{-11\,236+\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}\right)$$

$$\mathrm{Cos}\Bigg[0.730949+\mathrm{ArcSin}\Bigg[\Bigg(299\,792\,458\,\Bigg(4.98249\times10^{10}-4.56041\times10^{10}\,\mathrm{Sin}[0.578278-\delta p]-\frac{1}{299\,792\,458}\,\mathrm{ws}\,\mathrm{Sin}[0.578288+\delta sx]\,\Bigg(1-\frac{2.80753\times10^{32}\,\mathrm{InterpolatingFunction}\left[\text{Domain: }\{\{4.45\times10^{16},4.56\times10^{19}\}\}\ \text{Output: scalar}\right][\mathrm{ws}]}{\mathrm{ws}^{2}}\Bigg)\Bigg)\Bigg)\Bigg/$$

$$\left(\sqrt{\left(1+\frac{1.17302\times10^{36}}{\left(-30\,625+\frac{359\,502\,071\,494\,727\,056\,000\,000\,000\,000\,000\,000\,\pi^{2}}{\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}\right)\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}+\frac{1.53833\times10^{37}}{\left(-11\,236+\frac{359\,502\,071\,494\,727\,056\,000\,000\,000\,000\,000\,000\,\pi^{2}}{\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}\right)\left(1.36719\times10^{19}-\mathrm{ws}\right)^{2}}\right)}\left(1.36719\times10^{19}-\mathrm{ws}\right)\right)\Bigg]\Bigg]$$

In[∘]:=
```
(*κ1[ws_,δsx_,δp_]=
  (2*hb*ws*(wp0-ws)*wp0/(c*ϵ0)^3/n^2)^0.5*ϵ0*deff/.{n→nvisible[λv[wp0-ws]]};*)
κ1[ws_, δsx_, δp_] = (2 * hb * ws * (wp0 - ws) * wp0 / (c * ϵ0) ^ 3 / n) ^ 0.5 * ϵ0 * deff /.
   {n → nvisible[λv[wp0 - ws]]};
```

In[∘]:=
```
κsig[ws_, δsx_, δp_] := If[Im[δix[ws, δsx, δp]] == 0, κ1[ws, δsx, δp], 0]
```

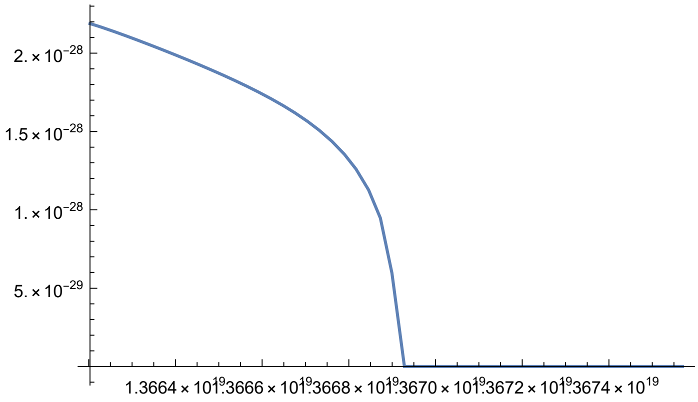
In[∘]:=
```
Abs[κsig[ws0, 0, 0]]^2
```

Out[∘]=

$$8.14009\times10^{-29}$$

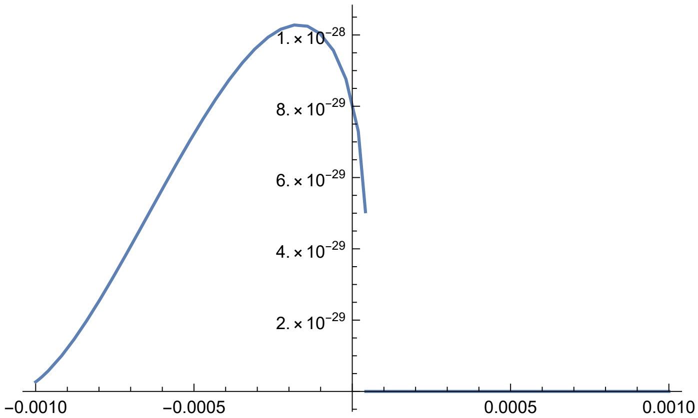
In[•]:=
```
Plot[Abs[κsig[ws, 0, 0]]^2, {ws, 0.9995 * ws0, 1.0005 * ws0}]
```

Out[•]=



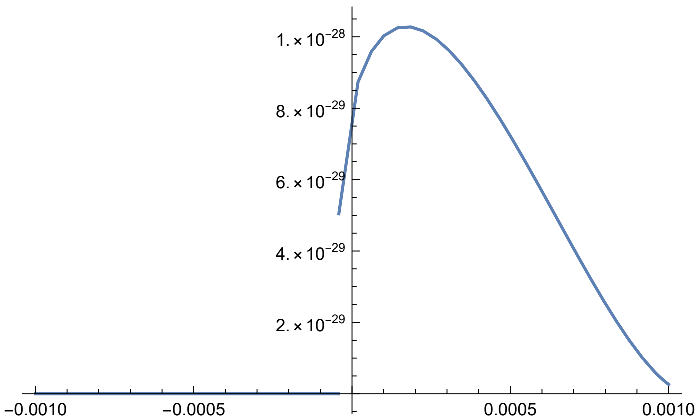
In[•]:=
```
Plot[Abs[κsig[ws0, δsx, 0]]^2, {δsx, -1 * 10^-3, 1 * 10^-3}]
```

Out[•]=

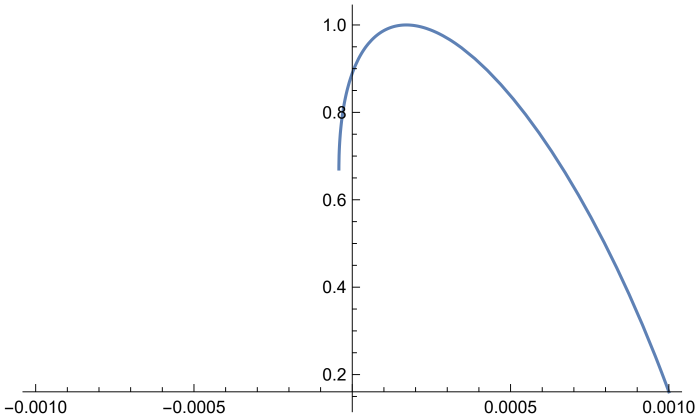


In[•]:=
```
Plot[Abs[κsig[ws0, 0, δp]]^2, {δp, -1 * 10^-3, 1 * 10^-3}]
```

Out[•]=



In[ ]:=
```
Plot[Cos[θi0 + δix[ws0, 0, δp]], {δp, -1 * 10^-3, 1 * 10^-3}]
```
Out[ ]=



In[ ]:=
```
λv[wp0 - 0.9995 * wp0]
```
Out[ ]=

275.551

In[ ]:=
```
λv[wp0 - 0.99989 * wp0]
```
Out[ ]=

1252.5

In[ ]:=
```
r = 1 / (Sqrt[Abs[Cos[θs0] * Cos[θi0 + δix[ws, δsx, δp]]]]);
```

In[ ]:=
```
κsig1 = κsig[ws0, 0, 0]
```
Out[ ]=

$9.02224 \times 10^{-15} + 0.\ i$

Formuli (from Laue Matrix 7)

Signal

In[ ]:=
```
κsig[ws0]
```
Out[ ]=

$\kappa\text{sig}[1.36689 \times 10^{19}]$

In[ ]:=
```
signalintegrand[ws_, δsx_, δsy_, δp_] = -1 * 10^13 * 1 / (2 * Pi)^3 * Cos[θs0] *
```
$$\text{ws}^2/\text{c}^2 * \frac{4\, r^2\, \text{Sin}\left[\frac{1}{2}\, L\, \Delta\text{kzsig}[\text{ws}, \delta\text{sx}, \delta\text{sy}, \delta\text{p}]\right]^2 \kappa\text{sig}[\text{ws}, \delta\text{sx}, \delta\text{p}]^2}{\Delta\text{kzsig}[\text{ws}, \delta\text{sx}, \delta\text{sy}, \delta\text{p}]^2};$$

(*signalintegrand2theta[ws_,δsx_,δsy_]=
$$-2*10^{13}*1/(2*\text{Pi})^3*\text{Cos}[\theta\text{s0}]*\text{ws}^2/\text{c}^2*\frac{4\, r^2\, \text{Sin}\left[\frac{1}{2}\, L\, \Delta\text{kzsig}[\text{ws},\delta\text{sx},\delta\text{sy},\Delta]\right]^2 \kappa\text{sig}[\text{ws},\delta\text{sx},\Delta]^2}{\Delta\text{kzsig}[\text{ws},\delta\text{sx},\delta\text{sy},\Delta]^2};*)$$

In[ ]:=
```
Clear[signalintegrand, signalintegrand1, signalintegrand2]
```

Signal

In[ ]:=
```
δp = 0;
```

In[ ]:= κsig[ws0, 0, 0]

Out[ ]=

$9.02224 \times 10^{-15} + 0.\, i$

```
In[ ]:= (*signalintegrand[ws_,δsx_,δsy_,δ_]=
  10^13*1/(2*Pi)^3*Cos[θs0]*ws^2/c^2*16 Abs[(r Cos[1/2 (δ-L Δkzsig[ws,δsx,δsy,0])]
        Sin[1/2 L Δkzsig[ws,δsx,δsy,0]] κsig[ws,0,0])/Δkzsig[ws,δsx,δsy,0]]^2;*)

In[ ]:= signalintegrand[ws_, δsx_, δsy_, δ_] =
  ((10^13 * 1 / (2 * Pi)^3 * Cos[θs0] * ws^2 / c^2) * Abs[2 * L^2 * ((I * r * κsig[ws0, 0, 0])^2) *
      (1 + Cos[L * Δkzsig[ws, δsx, δsy, 0] - 2 δ]) * Sinc[(L * Δkzsig[ws, δsx, δsy, 0])/2]^2]);

In[ ]:= signalintegrand1[ws_, δsx_, δsy_, δ_] :=
  If[Im[δix[ws, δsx, 0]] == 0, signalintegrand[ws, δsx, δsy, δ], 0]
 signalintegrand2[ws_, δsx_, δsy_, δ_] :=
  If[Im[δiy[ws, δsy]] == 0, signalintegrand1[ws, δsx, δsy, δ], 0]

In[ ]:= signalintegrand2[ws0, 0, 0, 0]
```

Out[ ]=

6.76316×10^{-11}

Phase difference dependence

```
In[ ]:= wlow = 0.99991; whigh = 1.00009; numbersteps = 10; stepsize = (whigh - wlow) / numbersteps;

In[ ]:= ϕsignalmax1x = θslim[0.1]
```

Out[ ]=

0.0000258499

```
In[ ]:= ϕsignalmax1y = θslimy[0.1]
```

Out[ ]=

0.0000535642

```
In[ ]:= ϕsignalmax1x * 10^4
```

Out[ ]=

0.258499

```
In[ ]:= Chop[NIntegrate[signalintegrand2[ws, δsx, δsy, 0],
   {δsx, -1 * 0.00004, 0.00004}, {δsy, -1 * 0.00003, 0.00003},
   {ws, wlow * ws0, whigh * ws0}, Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]
```

Out[ ]=

0.000368051

```
In[ ]:= phase[δ_] := Chop[NIntegrate[signalintegrand2[ws, δsx, δsy, δ],
    {δsx, -1 * 0.00003, 0.00003}, {δsy, -1 * 0.00003, 0.00003},
    {ws, wlow * ws0, whigh * ws0}, Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]

In[ ]:= phase[π / 2]
```

Out[ ]=

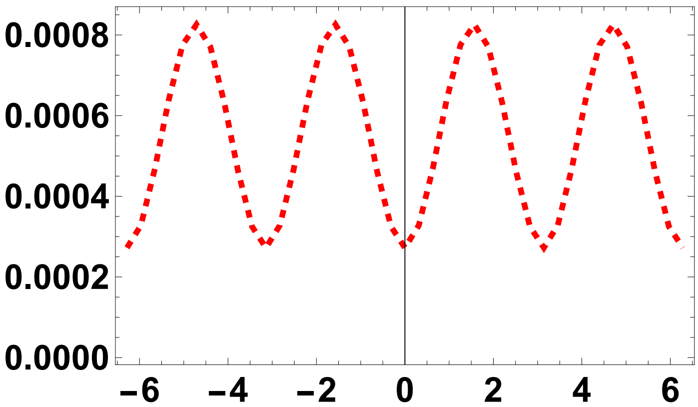
0.000825807

In[∘]:=
```
Clear[phasedata]
phasedata = ParallelTable[{δ1, phase[δ1]}, {δ1, -2 * Pi, 2 * Pi, 0.1 * Pi}]
```

Out[∘]=
```
{{-6.28319, 0.00027234}, {-5.96903, 0.000327855},
 {-5.65487, 0.000467528}, {-5.34071, 0.000638442}, {-5.02655, 0.000775582},
 {-4.71239, 0.000825807}, {-4.39823, 0.000770598}, {-4.08407, 0.000632238},
 {-3.76991, 0.000461424}, {-3.45575, 0.000324829}, {-3.14159, 0.00027234},
 {-2.82743, 0.000327855}, {-2.51327, 0.000467528}, {-2.19911, 0.000638442},
 {-1.88496, 0.000775582}, {-1.5708, 0.000825807}, {-1.25664, 0.000770598},
 {-0.942478, 0.000632238}, {-0.628319, 0.000461424}, {-0.314159, 0.000324829},
 {0., 0.00027234}, {0.314159, 0.000327855}, {0.628319, 0.000467528},
 {0.942478, 0.000638442}, {1.25664, 0.000775582}, {1.5708, 0.000825807},
 {1.88496, 0.000770598}, {2.19911, 0.000632238}, {2.51327, 0.000461424},
 {2.82743, 0.000324829}, {3.14159, 0.00027234}, {3.45575, 0.000327855},
 {3.76991, 0.000467528}, {4.08407, 0.000638442}, {4.39823, 0.000775582},
 {4.71239, 0.000825807}, {5.02655, 0.000770598}, {5.34071, 0.000632238},
 {5.65487, 0.000461424}, {5.96903, 0.000324829}, {6.28319, 0.00027234}}
```

In[∘]:=
```
ListPlot[phasedata, Joined → True,
 PlotStyle → {{RGBColor[1, 0, 0], Dashed, AbsoluteThickness[3]}},
 Frame → True, BaseStyle → {FontWeight → "Bold", FontSize → 18}]
```

Out[∘]=



In[∘]:=
```
visibility = (Max[Last /@ Level[Cases[%, _Line, Infinity], {-2}]] -
     Min[Last /@ Level[Cases[%, _Line, Infinity], {-2}]]) /
   (Max[Last /@ Level[Cases[%, _Line, Infinity], {-2}]] +
     Min[Last /@ Level[Cases[%, _Line, Infinity], {-2}]])
```

Out[∘]=
```
0.504001
```

In[∘]:=
```
Clear[wlow, whigh]
wlow[Δrelative_] := 1 - Δrelative; whigh[Δrelative_] := 1 + Δrelative;
Eiwhigh[Δrelative_] := (wp0 - ws0 * wlow[Δrelative]) * (hb / q)
Eiwlow[Δrelative_] := (wp0 - ws0 * whigh[Δrelative]) * (hb / q)
ΔEi[Δrelative_] := Eiwhigh[Δrelative] - Eiwlow[Δrelative]
```

In[∘]:=
```
Eiwhigh[Δrelative]
Eiwlow[Δrelative]
```

Out[∘]=

$6.58284 \times 10^{-16} \left(1.36719 \times 10^{19} - 1.36689 \times 10^{19} (1 - \triangle\text{relative})\right)$

Out[∘]=

$6.58284 \times 10^{-16} \left(1.36719 \times 10^{19} - 1.36689 \times 10^{19} (1 + \triangle\text{relative})\right)$

In[∘]:=
```
phaseΔrel[Δrelative_, δ_] :=
 Chop[NIntegrate[signalintegrand2[ws, δsx, δsy, δ], {δsx, -1 * 0.00003, 0.00003},
   {δsy, -1 * 0.00003, 0.00003}, {ws, wlow[Δrelative] * ws0, whigh[Δrelative] * ws0},
   Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]
```

In[∘]:=
```
Clear[phaseΔrelDiscrt]
phaseΔrelDiscrt[Δrelative_] := Table[phaseΔrel[Δrelative, δ], {δ, 0, 2 * Pi, 0.1 * Pi}];
```

In[∘]:=
```
numberstepsΔEi = 2;
ΔrelativeLow = 0.00001;
ΔrelativeHigh = 0.00009;
stepsizeΔEi = (ΔrelativeHigh - ΔrelativeLow) / numberstepsΔEi
```

Out[∘]=

0.00004

In[∘]:=
```
Clear[contrastDiscrt]
contrastDiscrt[Δrelative_] :=
  (Max[phaseΔrelDiscrt[Δrelative]] - Min[phaseΔrelDiscrt[Δrelative]]) /
   (Max[phaseΔrelDiscrt[Δrelative]] + Min[phaseΔrelDiscrt[Δrelative]]);
```

In[∘]:=
```
Δrelative = 0.05;
contrastDiscrt[Δrelative]
```

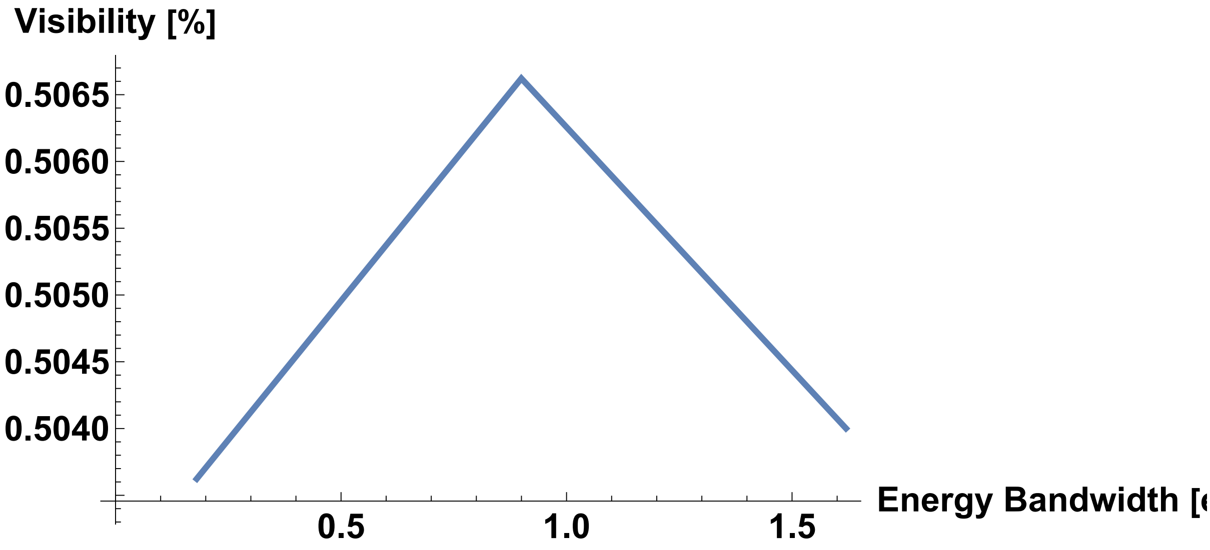
Out[∘]=

0.943877

In[∘]:=
```
Clear[dataVisVsΔEi]
dataVisVsΔEi = ParallelTable[{ΔEi[q2], contrastDiscrt[q2]},
   {q2, ΔrelativeLow, ΔrelativeHigh, stepsizeΔEi}];
```

In[∘]:=
```
ListPlot[dataVisVsΔEi, Joined → True, PlotStyle → {Thickness[0.008]},
 BaseStyle → {FontWeight → "Bold", FontSize → 18}]
```

In[•]:=
```
Show[%, AxesLabel → {HoldForm["Energy Bandwidth [eV]"], HoldForm["Visibility [%]"]},
 PlotLabel → None, LabelStyle → {18, GrayLevel[0]}]
```
Out[•]=

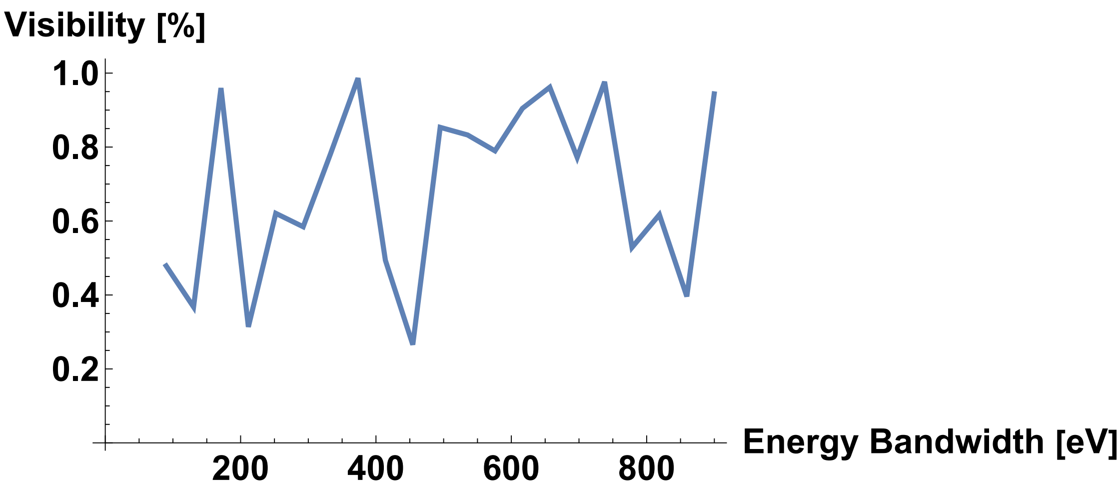


In[•]:=
```
numberstepsΔEs = 20;
ΔrelativeSLow = 0.005;
ΔrelativeSHigh = 0.05;
stepsizeΔEs = (ΔrelativeSHigh - ΔrelativeSLow) / numberstepsΔEs
```
Out[•]=
```
0.00225
```

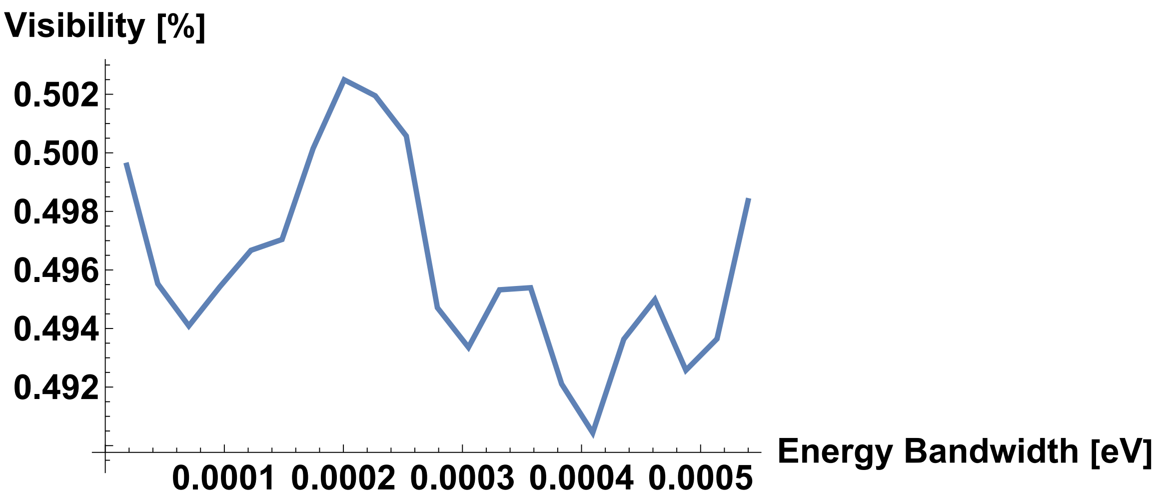
In[•]:=
```
Clear[dataVisVsΔEiSig]
dataVisVsΔEiSig = ParallelTable[{ΔEi[q3], contrastDiscrt[q3]},
   {q3, ΔrelativeSLow, ΔrelativeSHigh, stepsizeΔEs}];
```

In[•]:=
```
ListPlot[dataVisVsΔEiSig, Joined → True, PlotStyle → {Thickness[0.008]},
 BaseStyle → {FontWeight → "Bold", FontSize → 18}]
```

In[•]:=
```
Show[%, AxesLabel → {HoldForm["Energy Bandwidth [eV]"], HoldForm["Visibility [%]"]},
 PlotLabel → None, LabelStyle → {18, GrayLevel[0]}]
```
Out[•]=



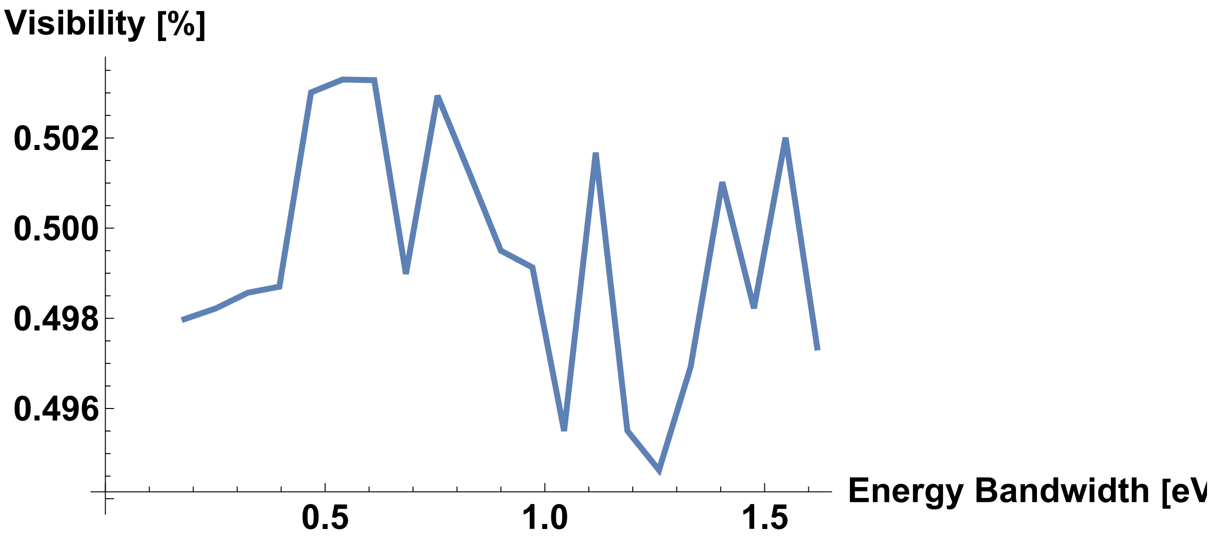
```
In[•]:= phaseΔrel2[Δrelative_, δ_] := Chop[NIntegrate[
            signalintegrand2[ws, δsx, 0, δ], {ws, wlow[Δrelative] * ws0, whigh[Δrelative] * ws0},
            {δsx, -0.00004, 0.00004}, Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]

In[•]:= Clear[phaseΔrelDiscrt2]
        phaseΔrelDiscrt2[Δrelative_] :=
          Table[phaseΔrel2[Δrelative, δ], {δ, 0, 2 * Pi, 0.1 * Pi}];

In[•]:= numberstepsΔEi2 = 20;
        ΔrelativeLow2 = 0.000000001;
        ΔrelativeHigh2 = 0.00000003;
        stepsizeΔEi2 = (ΔrelativeHigh2 - ΔrelativeLow2) / numberstepsΔEi2
Out[•]=
        1.45 × 10^-9

In[•]:= Clear[contrastDiscrt2]
        contrastDiscrt2[Δrelative_] :=
          (Max[phaseΔrelDiscrt2[Δrelative]] - Min[phaseΔrelDiscrt2[Δrelative]]) /
           (Max[phaseΔrelDiscrt2[Δrelative]] + Min[phaseΔrelDiscrt2[Δrelative]]);

In[•]:= Clear[dataVisVsΔEi2]
        dataVisVsΔEi2 = ParallelTable[{ΔEi[q2], contrastDiscrt2[q2]},
            {q2, ΔrelativeLow2, ΔrelativeHigh2, stepsizeΔEi2}];

In[•]:= ListPlot[dataVisVsΔEi2, Joined → True, PlotStyle → {Thickness[0.008]},
         BaseStyle → {FontWeight → "Bold", FontSize → 18}]

In[•]:= Show[%, AxesLabel → {HoldForm["Energy Bandwidth [eV]"], HoldForm["Visibility [%]"]},
         PlotLabel → None, LabelStyle → {18, GrayLevel[0]}]
Out[•]=
```



without integrating over δsx,δsy

```
In[•]:= phaseΔrel2[Δrelative_, δ_] :=
         Chop[NIntegrate[signalintegrand2[ws, 0, 0, δ], {ws, wlow[Δrelative] * ws0,
            whigh[Δrelative] * ws0}, Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]
```

```
In[=]:= Clear[phaseΔrelDiscrt2]
        phaseΔrelDiscrt2[Δrelative_] :=
          Table[phaseΔrel2[Δrelative, δ], {δ, 0, 2 * Pi, 0.1 * Pi}];

In[=]:= numberstepsΔEi2 = 20;
        ΔrelativeLow2 = 0.00001;
        ΔrelativeHigh2 = 0.00009;
        stepsizeΔEi2 = (ΔrelativeHigh2 - ΔrelativeLow2) / numberstepsΔEi2

Out[=]=
        4. × 10^-6

In[=]:= Clear[contrastDiscrt2]
        contrastDiscrt2[Δrelative_] :=
          (Max[phaseΔrelDiscrt2[Δrelative]] - Min[phaseΔrelDiscrt2[Δrelative]]) /
           (Max[phaseΔrelDiscrt2[Δrelative]] + Min[phaseΔrelDiscrt2[Δrelative]]);

In[=]:= Clear[dataVisVsΔEi2]
        dataVisVsΔEi2 = ParallelTable[{ΔEi[q2], contrastDiscrt2[q2]},
           {q2, ΔrelativeLow2, ΔrelativeHigh2, stepsizeΔEi2}];

In[=]:= ListPlot[dataVisVsΔEi2, Joined → True, PlotStyle → {Thickness[0.008]},
         BaseStyle → {FontWeight → "Bold", FontSize → 18}]

In[=]:= Show[%, AxesLabel → {HoldForm["Energy Bandwidth [eV]"], HoldForm["Visibility [%]"]},
         PlotLabel → None, LabelStyle → {18, GrayLevel[0]}]

Out[=]=
```



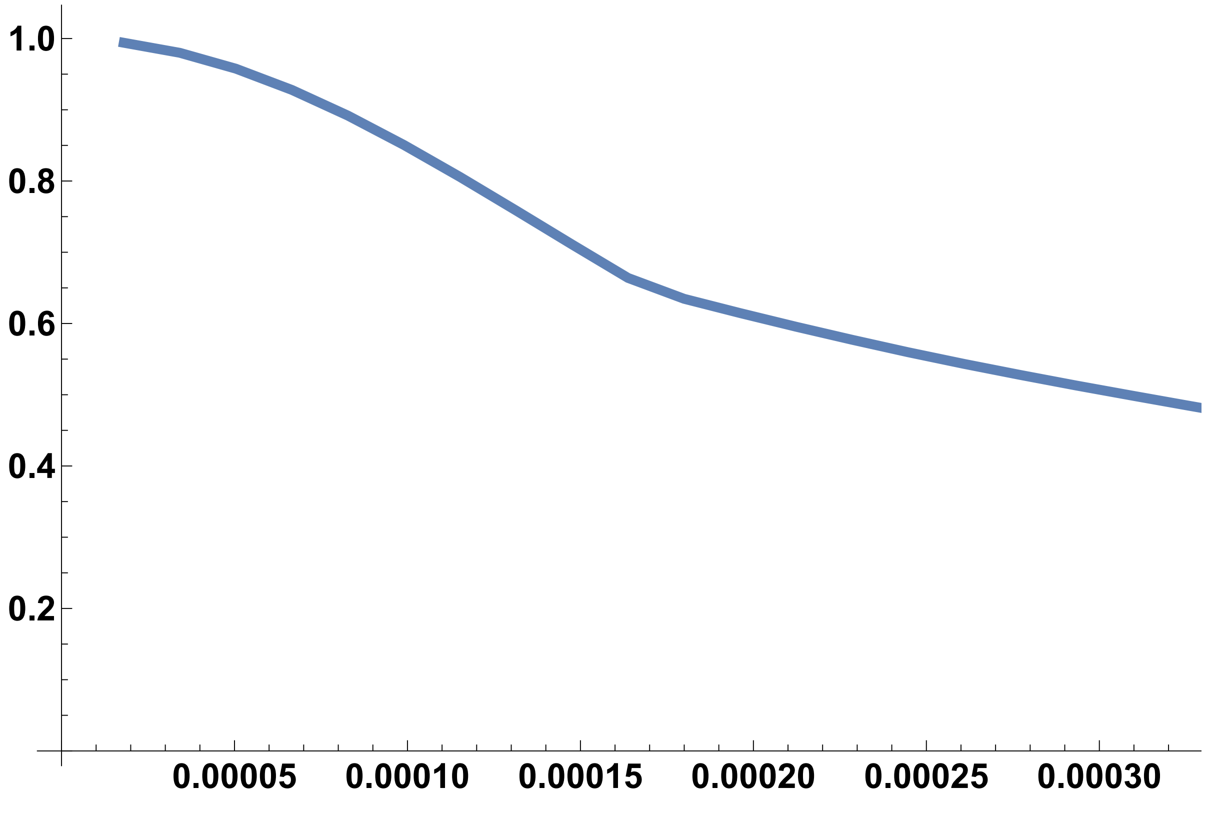
without integrating over δsx,δsy

```
In[=]:= phaseΔrel2[Δrelative_, δ_] :=
         Chop[NIntegrate[signalintegrand2[ws, 0, 0, δ], {ws, wlow[Δrelative] * ws0,
            whigh[Δrelative] * ws0}, Method → {"QuasiMonteCarlo", "MaxPoints" → 10^5}]]

In[=]:= Clear[phaseΔrelDiscrt2]
        phaseΔrelDiscrt2[Δrelative_] :=
          Table[phaseΔrel2[Δrelative, δ], {δ, 0, 2 * Pi, 0.1 * Pi}];
```

In[ ]:=
```
numberstepsΔEi2 = 20;
ΔrelativeLow2 = 0.000000001;
ΔrelativeHigh2 = 0.000000019;
stepsizeΔEi2 = (ΔrelativeHigh2 - ΔrelativeLow2) / numberstepsΔEi2
```
Out[ ]=

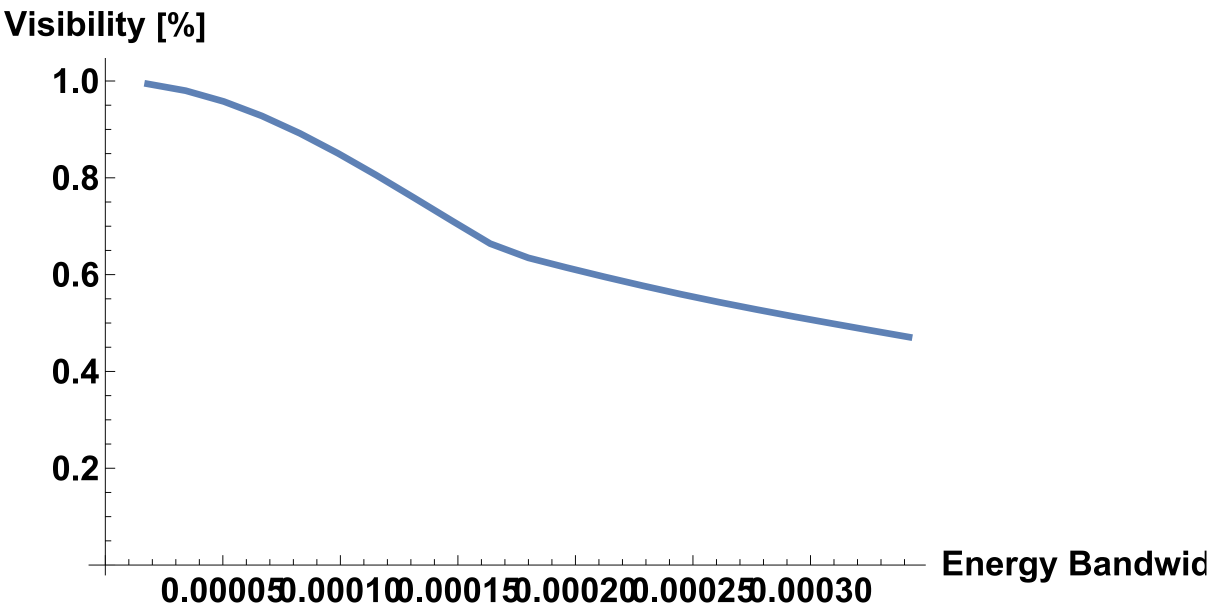
$9. \times 10^{-10}$

In[ ]:=
```
Clear[contrastDiscrt2]
contrastDiscrt2[Δrelative_] :=
  (Max[phaseΔrelDiscrt2[Δrelative]] - Min[phaseΔrelDiscrt2[Δrelative]]) /
   (Max[phaseΔrelDiscrt2[Δrelative]] + Min[phaseΔrelDiscrt2[Δrelative]]);
```
In[ ]:=
```
Clear[dataVisVsΔEi2]
dataVisVsΔEi2 = ParallelTable[{ΔEi[q2], contrastDiscrt2[q2]},
   {q2, ΔrelativeLow2, ΔrelativeHigh2, stepsizeΔEi2}];
```
In[ ]:=
```
ListPlot[dataVisVsΔEi2, Joined → True, PlotStyle → {Thickness[0.008]},
 BaseStyle → {FontWeight → "Bold", FontSize → 18}]
```
Out[ ]=



```
In[◦]:= Show[%, AxesLabel → {HoldForm["Energy Bandwidth [eV]"], HoldForm["Visibility [%]"]},
         PlotLabel → None, LabelStyle → {18, GrayLevel[0]}]
```

Out[◦]=



with Wp gaussian without integrating over δsx,δsy

```
In[◦]:= (*signalintegrand[ws0,0,0,0]*)

In[◦]:= signalintegrand1[ws_, δsx_, δsy_, δp_] :=
         If[Im[δix[ws, δsx, δp]] == 0, signalintegrand[ws, δsx, δsy, δp], 0]
        signalintegrand2[ws_, δsx_, δsy_, δp_] :=
         If[Im[δiy[ws, δsy]] == 0, signalintegrand1[ws, δsx, δsy, δp], 0]
        (*signalintegrand2theta1[ws_,δsx_,δsy_]:=
         If[Im[δix[ws,δsx]]==0,signalintegrand2theta[ws,δsx,δsy],0]
           signalintegrand2theta2[ws_,δsx_,δsy_]:=
          If[Im[δiy[ws,δsy]]==0,signalintegrand2theta1[ws,δsx,δsy],0]*)
```

Idler

```
idlerintegrand[ws_, δsx_, δsy_, δp_] = -1 * 10^13 * 1 / (2 * Pi)^3 * Cos[θs0] *
```

$$\text{ws}^2 / \text{c}^2 * \frac{4\,r^2 \,\text{Sin}\left[\frac{1}{2}\,\text{L}\,\Delta\text{kzsig}[\text{ws},\,\delta\text{sx},\,\delta\text{sy},\,\delta\text{p}]\right]^2 \kappa\text{sig}[\text{ws},\,\delta\text{sx},\,\delta\text{p}]^2}{\Delta\text{kzsig}[\text{ws},\,\delta\text{sx},\,\delta\text{sy},\,\delta\text{p}]^2};$$

```
idlerintegrand[ws0, 0, 0, 0]
```

$-1.6211\times10^{-10} + 0.\,i$

```
idlerintegrand1[ws_, δsx_, δsy_, δp_] :=
 If[Im[δix[ws, δsx, δp]] == 0, idlerintegrand[ws, δsx, δsy, δp], 0]
idlerintegrand2[ws_, δsx_, δsy_, δp_] :=
 If[Im[δiy[ws, δsy]] == 0, idlerintegrand1[ws, δsx, δsy, δp], 0]
```

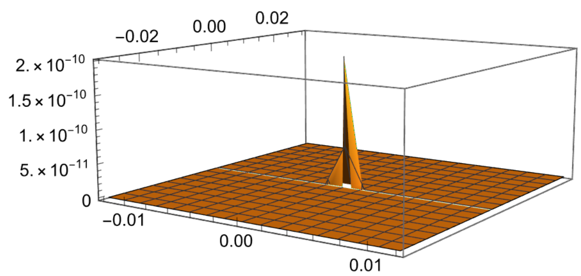
Coincidence

```
coin1integrand[ws_, δsx_, δsy_, δp_] = 1 * 10^13 * 1 / (2 * Pi)^3 *
   Cos[θs0] * ws^2 / c^2 * Abs[((-1 + e^(-i L Δkzsig[ws,δsx,δsy,δp])) r κsig[ws, δsx, δp]) / Δkzsig[ws, δsx, δsy, δp]]^2;
```

```
coinintegrand1[ws_, δsx_, δsy_, δp_] :=
 If[Im[δix[ws, δsx, δp]] == 0, coin1integrand[ws, δsx, δsy, δp], 0]
coinintegrand2[ws_, δsx_, δsy_, δp_] :=
 If[Im[δiy[ws, δsy]] == 0, coinintegrand1[ws, δsx, δsy, δp], 0]
```

Signal Integration

```
In[•]:= wlow = 1 - 1.102 * 10^-4; whigh = 1 + 1.102 * 10^-4;
        numbersteps = 20;
        stepsize = (whigh - wlow) / numbersteps;
        (*300mev @ 11keV → 0.136e-4*)

In[•]:= (*δsxlow=-12.7119*10^-4; δsxhigh=12.7119*10^-4;*)

In[•]:= sig[ws_] := Chop[NIntegrate[Chop[Abs[signalintegrand2[ws, δsx, δsy, 0]], 10^-10],
           {δsx, -5 * 10^-5, 5 * 10^-5}, {δsy, -3 * 10^-4, 3 * 10^-4},
           Method → {"QuasiMonteCarlo", "MaxPoints" → 10^4}], 10^-19];

In[•]:= signalintegrand2[ws0, 0, 0, 0]
Out[•]=
        -6.28456 × 10^-11 + 0. i

In[•]:= Plot3D[Abs[signalintegrand2[ws0, δsx, δsy, 0]],
         {δsx, -0.012, 0.012}, {δsy, -0.03, 0.03}]
Out[•]=
```




```
data1 = Table[{(1 - q1) * ws0 * hb / q, sig[q1 * ws0]}, {q1, wlow, whigh, stepsize}];
```

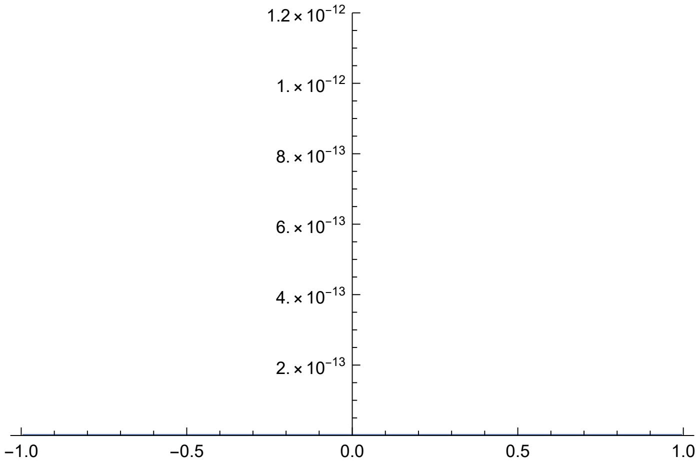
NIntegrate::maxp : The integral failed to converge after 5417 integrand evaluations. NIntegrate obtained 2.8625217499731342`*^-18 and 2.86620318600379`*^-20 for the integral and error estimates. >>

NIntegrate::maxp : The integral failed to converge after 10000 integrand evaluations. NIntegrate obtained 3.4571184268073098`*^-18 and 3.477503486923309`*^-20 for the integral and error estimates. >>

NIntegrate::maxp : The integral failed to converge after 5400 integrand evaluations. NIntegrate obtained 3.961754907534135`*^-18 and 3.9731749141430846`*^-20 for the integral and error estimates. >>

General::stop : Further output of NIntegrate::maxp will be suppressed during this calculation. >>

```
ListPlot[data1, Joined → True, PlotRange → {0, 1.2 * 10^-12}]
```




```
traprule =
 ws0 * stepsize * (Plus @@ Transpose[data1][[2]] - (sig[wlow * ws0] + sig[whigh * ws0]) / 2)
```

NIntegrate::maxp : The integral failed to converge after 5417 integrand evaluations. NIntegrate obtained 2.8625217499731342`*^-18 and 2.86620318600379`*^-20 for the integral and error estimates. ≫

NIntegrate::maxp : The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained 0.` and 0.` for the integral and error estimates. ≫

```
0.00744913
```

```
check = Chop[NIntegrate[Abs[signalintegrand2[ws, δsx, δsy, 0]],
   {δsx, -.0001, .0001}, {δsy, -.003, .003}, {ws, wlow * ws0, whigh * ws0},
   Method → {"QuasiMonteCarlo", "MaxPoints" → 10^3}]]
```

NIntegrate::maxp : The integral failed to converge after 1000 integrand evaluations. NIntegrate obtained 0.24386274583060433` and 0.07318471177408574` for the integral and error estimates. ≫

```
0.243863
```

Rocking curve

```
In[]:= (*sigrock[δ_]:=Chop[NIntegrate[Chop[Abs[signalintegrand2[ws,δsx,δsy,δ]],10^-10],
         {δsx,-1*10^-4,1*10^-4},{δsy,-3*10^-4,3*10^-4},{ws,wlow*ws0,whigh*ws0},
         Method→{"QuasiMonteCarlo","MaxPoints"→10^5}],10^-19];*)
       δlow = -10^-3; δhigh = 1*10^-3; δres = 1*10^-4;
       (*δsxlow=-12.7119*10^-4; δsxhigh=12.7119*10^-4;*) (*1.5mm*)
       (*δsxlow=-20*10^-4; δsxhigh=20*10^-4;*) (*4mm*)
       (*δsxlow=-5*10^-4; δsxhigh=5*10^-4;*) (*1mm*)
       δsxlow = -5*10^-3; δsxhigh = 5*10^-3; (*0.1 mm at 615mm distance=0.1626mrad*)
       (*δsxlow=-6.25*10^-4; δsxhigh=6.25*10^-4;*) (*2.5mm*)
       (*δsxlow=-14.8305*10^-4; δsxhigh=14.8305*10^-4;*) (*1.75mm*)
       δsylow = -5*10^-3; δsyhigh = 5*10^-3; (*2 mm RS2*)
       (*δsylow=-42.37*10^-4; δsyhigh=42.37*10^-4;*)
       (*δsylow=-42.37*10^-4δsyhigh=42.37*10^-4*)
       sigrock[δ_] := Chop[NIntegrate[Chop[Abs[signalintegrand2[ws, δsx, δsy, δ]], 10^-10],
         {δsx, δsxlow, δsxhigh}, {δsy, δsylow, δsyhigh}, {ws, wlow*ws0, whigh*ws0},
         Method → {"QuasiMonteCarlo", "MaxPoints" → 1*10^4}], 10^-19];
       (* Integration over detector plase (angle) δsx(2theta direction) and δsy *)

In[]:= (*datasigrock1=
        ParallelTable[{q1*10^4,sigrock[q1]},{q1,-1*10^-3,1*10^-3,2*10^-4}];*)
       (*labmda_idler 600nm - {δsx,-4.2373*10^-4,4.2373*10^-4}
         (Detector effective aperture - RS2 hor. in radians),
       {δsy,-2.1186*10^-4,2.1186*10^-4}
        (RS1 aperture (min of RS1 and RS2 ver.) in radians)*)

       datasigrock1 = ParallelTable[{q1(**10^3*), sigrock[q1]}, {q1, δlow, δhigh, δres}];
       (* The last value is the rocking curve resolution*)

(kernel 2)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          0.015936017183115096` and 0.002279273614428101` for the integral and error estimates.
(kernel 4)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          0.049641286537873835` and 0.009593997386626044` for the integral and error estimates.
(kernel 6)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          1.206789463527927` and 0.5757647775078254` for the integral and error estimates.
(kernel 1)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          0.01462360968749989` and 0.0020969248645139487` for the integral and error estimates.
(kernel 3)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          0.027777529465448265` and 0.0040945296394549425` for the integral and error estimates.
(kernel 5)
       NIntegrate::maxp :
        The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
          0.08005566479772822` and 0.011851149444331065` for the integral and error estimates.
```

```
(kernel 7)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       3.5994142821240884` and 2.8481796602216853` for the integral and error estimates.
(kernel 8)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       2539.0987275708308` and 2476.4269294374244` for the integral and error estimates.
(kernel 1)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.016027352312612517` and 0.0021525059867223656` for the integral and error estimates.
(kernel 3)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.03592788368130967` and 0.006033676145019395` for the integral and error estimates.
(kernel 4)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.06613585852855253` and 0.011197353988165383` for the integral and error estimates.
(kernel 2)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.02177595866602288` and 0.0031032347562491317` for the integral and error estimates.
(kernel 5)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.1787817937979716` and 0.039762965787924286` for the integral and error estimates.
(kernel 6)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       94.25469247394281` and 62.37273853996543` for the integral and error estimates.
(kernel 8)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       2366.0954885084175` and 2306.139272029837` for the integral and error estimates.
(kernel 7)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       65.9083827779767` and 28.64761099446315` for the integral and error estimates.
(kernel 1)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       1.0814643106083117` and 0.2755571041044932` for the integral and error estimates.
(kernel 4)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.13874575522302252` and 0.020487141722985523` for the integral and error estimates.
(kernel 3)
    NIntegrate::maxp :
     The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
       0.2916901309994567` and 0.05165857466158206` for the integral and error estimates.
```

(kernel 5)

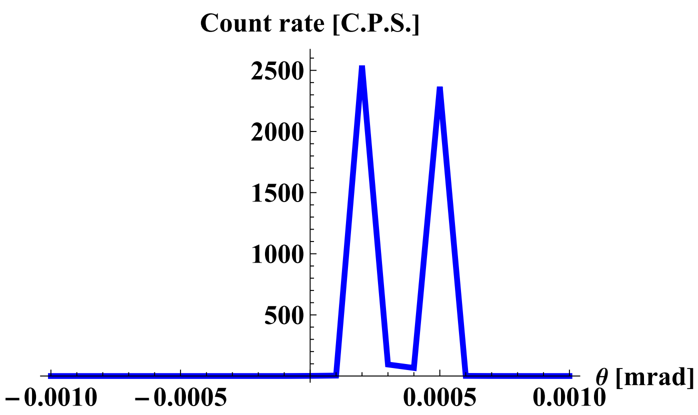
```
NIntegrate::maxp :
 The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
   0.052641166880735356` and 0.007331639067944236` for the integral and error estimates.
```

(kernel 2)

```
NIntegrate::maxp :
 The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained
   0.08910709419077031` and 0.012879448846151779` for the integral and error estimates.
```

In[ ]:=
```
ListPlot[datasigrock1, Joined → True,
 LabelStyle → {FontFamily → "Times", FontSize → 14, FontWeight → "Bold"},
 PlotStyle → {{RGBColor[0, 0, 1], AbsoluteThickness[3]}},
 AxesLabel → {Style["θ [mrad]", 14, Black, Bold],
   Style["Count rate [C.P.S.]", 14, Black, Bold]}, PlotRange → All]
```

Out[ ]=




```
ΔkzSig[ws_, δsx_, δsy_, δp_] =
  wp0 * n[wp0] * Cos[θp0 + δp] / c - ws * n[ws] * Cos[θs0 + δsx] * Cos[δsy] / c -
   (wp0 - ws) * nvisible[wp0 - ws] * Cos[θi0 + δθi[ws, δsx, δp]] * Cos[δiy[ws, δsy]] / c;
```

In[ ]:=
```
SignalIntegrand[wi_, δsx_, δsy_] = ((10^13 * 1 / (2 * Pi)^3 * Cos[θi0] * wi^2 / c^2) *
    Abs[2 * L^2 * ((I * r * κsig[ws0, 0, 0])^2) * Sinc[(L * Δkzsig[wp0 - wi, δsx, δsy, 0]) / 2]]^2);
```

In[ ]:=
```
SignalIntegrand1[wi_, δsx_, δsy_] :=
 If[Im[δix[wp0 - wi, δsx, 0]] == 0, SignalIntegrand[wi, δsx, δsy], 0]
SignalIntegrand2[wi_, δsx_, δsy_] :=
 If[Im[δiy[wp0 - wi, δsy]] == 0, SignalIntegrand1[wi, δsx, δsy], 0]
(*signalintegrand2theta1[ws_,δsx_,δsy_]:=
 If[Im[δix[ws,δsx]]==0,signalintegrand2theta[ws,δsx,δsy],0]
   signalintegrand2theta2[ws_,δsx_,δsy_]:=
  If[Im[δiy[ws,δsy]]==0,signalintegrand2theta1[ws,δsx,δsy],0]*)
```

In[ ]:=
```
datasigRock[δsx_, δsy_] :=
  Chop[NIntegrate[Chop[Abs[SignalIntegrand2[wi, δsx, δsy]], 10^-10],
    {wi, (1 - 0.3) * wi0, (1 + 0.3) * wi0},
    Method → {"QuasiMonteCarlo", "MaxPoints" → 1 * 10^4}], 10^-19];
```

In[•]:=
```
DensityPlot[datasigRock[δsx, δsy], {δsx, -10^-4, 10^-4}, {δsy, -10^-4, 10^-4}]
```

NIntegrate: The integral failed to converge after 6828 integrand evaluations. NIntegrate obtained 0.` and 0.` for the integral and error estimates.

Out[•]=
```
$Aborted
```

```
(*sigrock[δ_]:=Chop[NIntegrate[Chop[Abs[signalintegrand2[ws,δsx,δsy,δ]],10^-10],
    {δsx,-1*10^-4,1*10^-4},{δsy,-3*10^-4,3*10^-4},{ws,wlow*ws0,whigh*ws0},
    Method→{"QuasiMonteCarlo","MaxPoints"→10^5}],10^-19];*)
δlow = -10^-3; δhigh = 1 * 10^-3; δres = 1 * 10^-4;
(*δsxlow=-12.7119*10^-4; δsxhigh=12.7119*10^-4;*) (*1.5mm*)
(*δsxlow=-20*10^-4; δsxhigh=20*10^-4;*) (*4mm*)
(*δsxlow=-5*10^-4; δsxhigh=5*10^-4;*) (*1mm*)
δsxlow = -5 * 10^-3; δsxhigh = 5 * 10^-3; (*0.1 mm at 615mm distance=0.1626mrad*)
(*δsxlow=-6.25*10^-4; δsxhigh=6.25*10^-4;*) (*2.5mm*)
(*δsxlow=-14.8305*10^-4; δsxhigh=14.8305*10^-4;*) (*1.75mm*)
δsylow = -5 * 10^-3; δsyhigh = 5 * 10^-3; (*2 mm RS2*)
(*δsylow=-42.37*10^-4; δsyhigh=42.37*10^-4;*)
(*δsylow=-42.37*10^-4δsyhigh=42.37*10^-4*)
sigRock[δ_] := Chop[NIntegrate[Chop[Abs[signalintegrand2[ws, δsx, δsy, δ]], 10^-10],
    {δsx, δsxlow, δsxhigh}, {δsy, δsylow, δsyhigh}, {ws, wlow * ws0, whigh * ws0},
    Method → {"QuasiMonteCarlo", "MaxPoints" → 1 * 10^4}], 10^-19];
(* Integration over detector plase (angle) δsx(2theta direction) and δsy *)
```