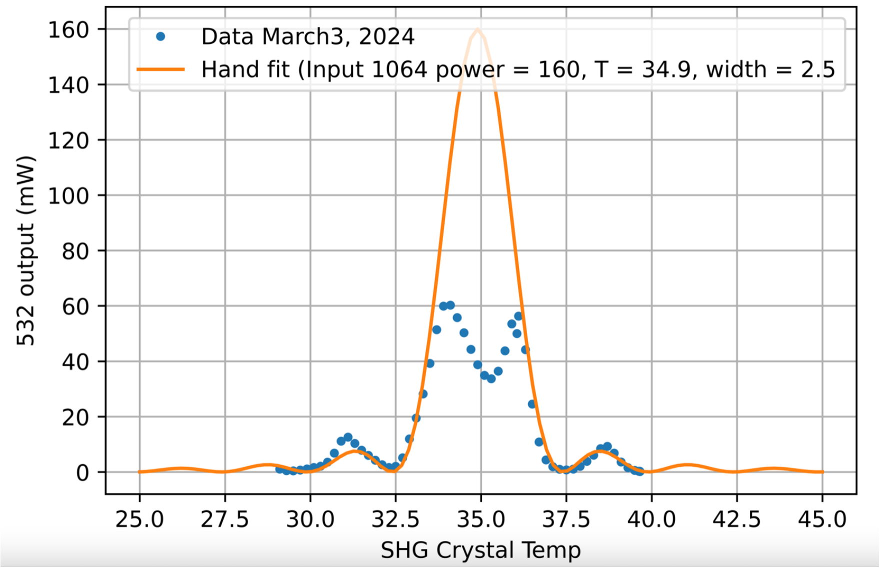
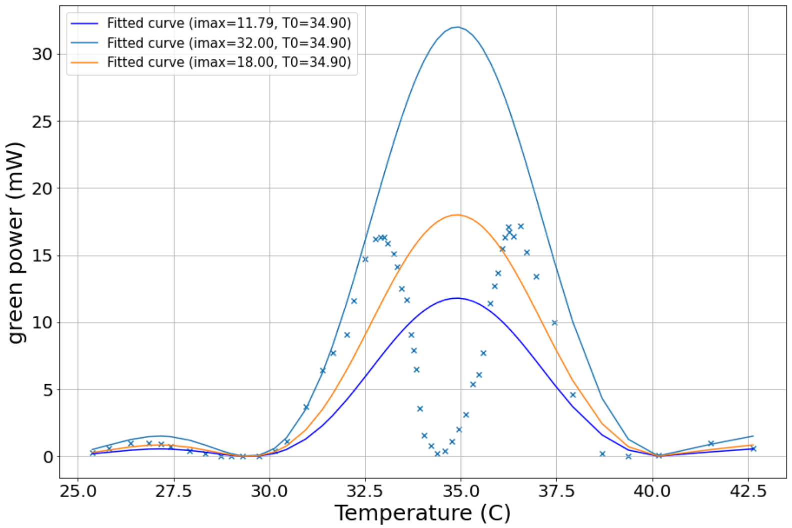
PNW mountains





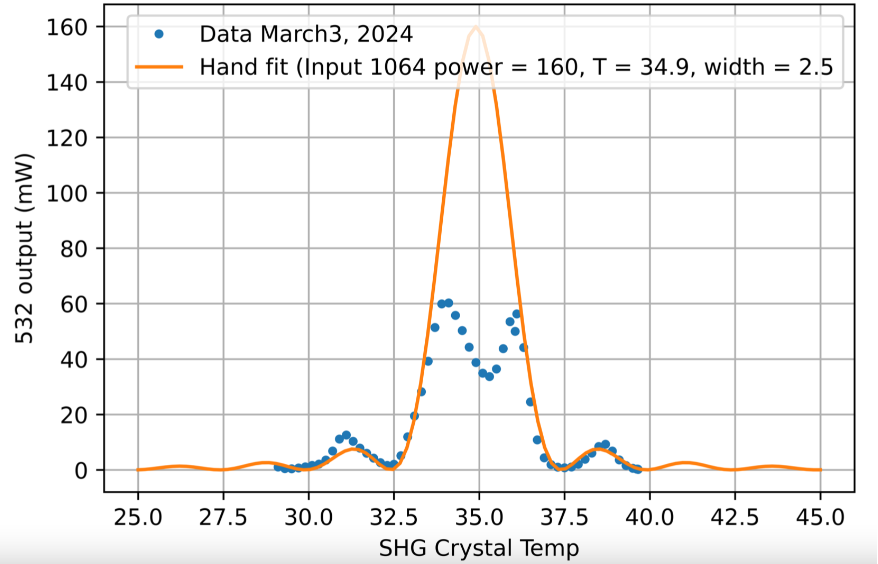
Not all of them are gray-track

Daniel, Nutsinee

In brand new SHGs, it’s the phase mismatch!

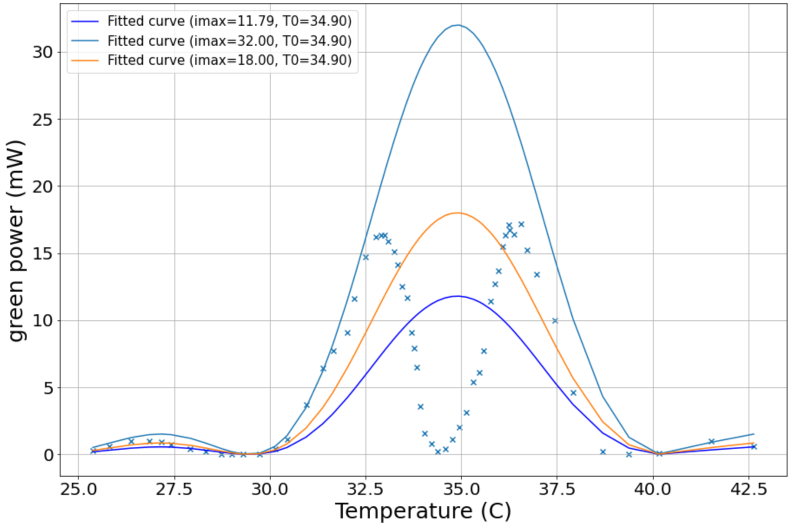
- Well turns out it’s the phase mismatch between the green and the red
- https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=78785
- https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=78715
- https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=78686

Mount Saint Helens. This one can’t be explained by the phase mismatch →



Twin Sisters Rock in a brand new SHG here

This can’t be the bloody gray-track now.



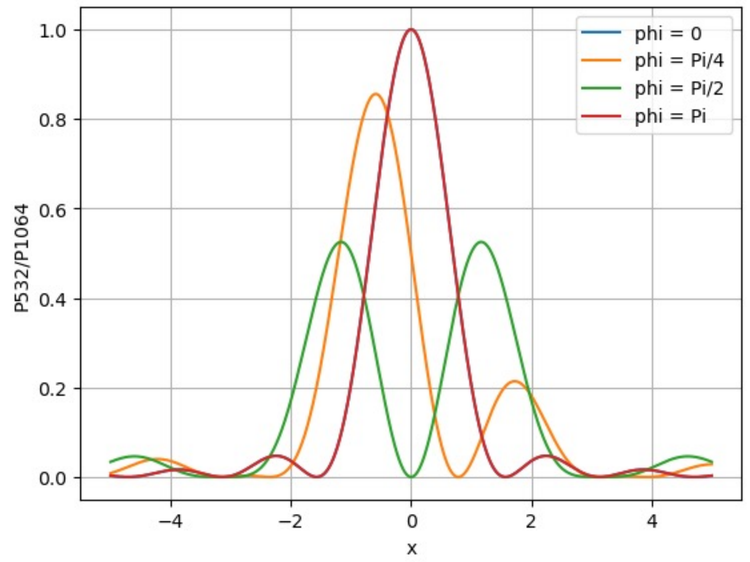
Daniel’s explanation

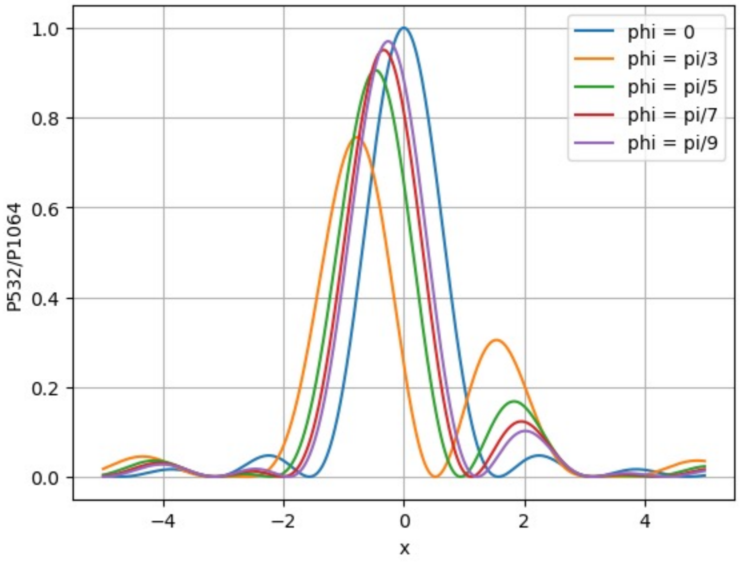
- “The dispersion that matters is between the first path and the second path in the crystal. In a simple dual path system, green light will be generated in phase with the red light in the first path, then the two frequencies propagate in air towards the rear mirror, where they are reflected with potentially different phases, and then propagate back to the crystal. According to Gonzalez et al. the dispersion in air for 1064/532 is 27.4°/cm (double passed). Our SHG length is about 5 cm long with a 1 cm crystal. So, the in-air path is of order 4 cm. You need a 90° phase shift to explain the double peak. However, the phase difference of our mirror reflectivity is unknown.” alog78939

Virgo SHG paper (2018)

- Leonardi et al. 2018 eq. 2 taken into account a phase difference
- https://iopscience.iop.org/article/10.1088/1555-6611/aad84d

$$\frac{P_{2\omega}}{P_{\omega}^{2}} \propto \operatorname{sinc}^2(x)\ \cos^2(x+\delta\phi)\,. \qquad (2)$$





Virgo SHG paper (2018)

- They saw similar things we did in the double pass

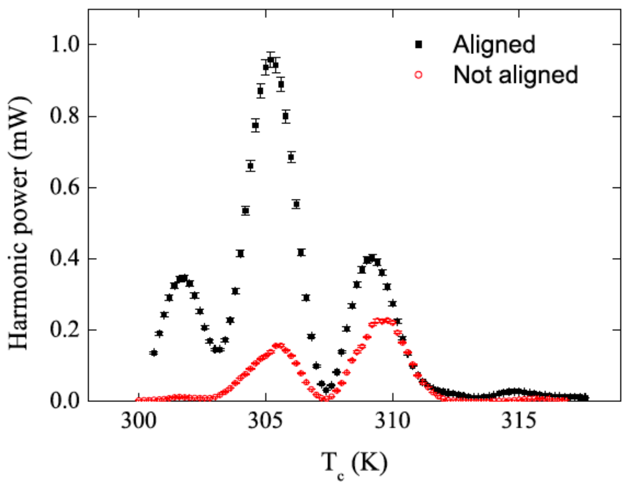


Figure 3. Second harmonic power in the double-pass configuration as a function of the PPKTP crystal temperature T_c, at two different beam alignments with respect to the PPKTP crystal: best overlap (black full squares) and slightly misaligned input beam (red open circles). The power data correspond to the time average of the PD3 signal, while the error bars represent the standard deviation. For each point we waited for the temperature to stabilize before acquiring the data. With the best alignment, maximum conversion is achieved at $T_c = T_m \equiv 305.5$ K.

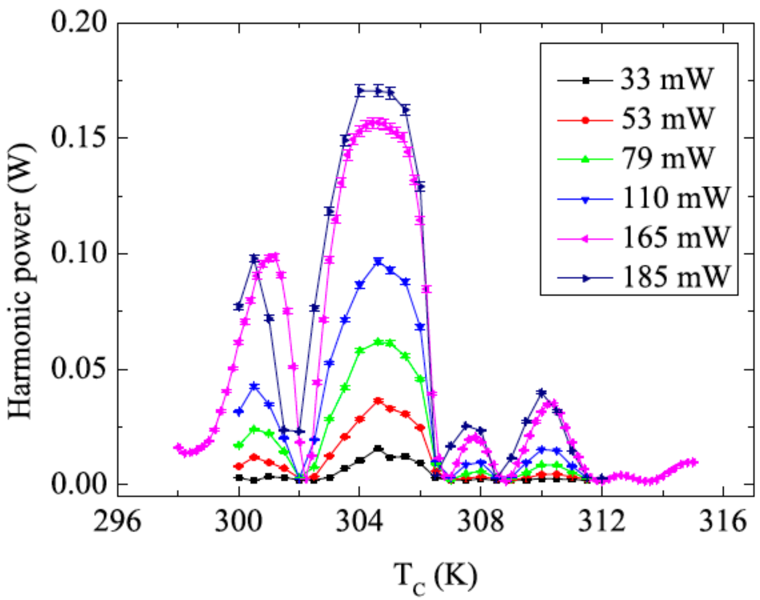
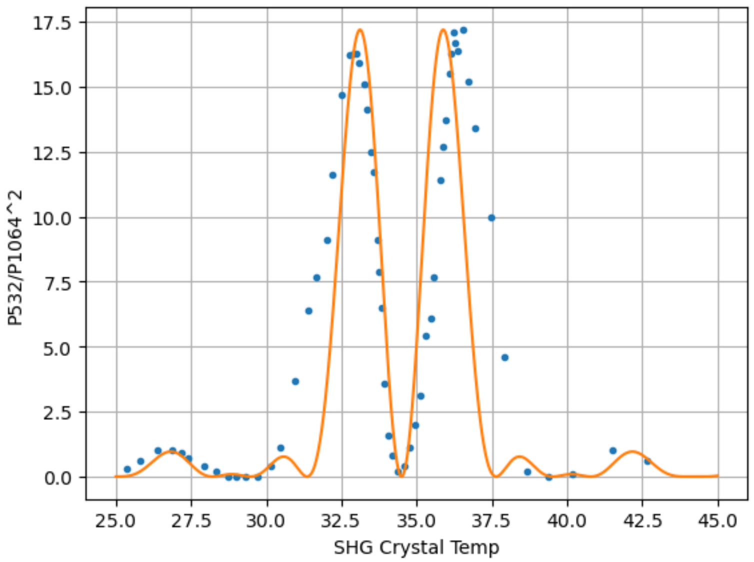


Figure 7. The phase-matching curve of the SHG cavity at various (mode matched) input fundamental powers as a function of the PPKTP crystal temperature T_c. In the legend: the input fundamental power for each curve.

Virgo SHG paper

- An actual first attempt to fit to Karmeng's data




```
def SHGsinccos(T, A, T0, width, phi):
    return A*((np.sinc((T-T0)/(width*np.pi)))**2)*(np.cos(((T-T0)/width)+phi)**2)
```

```
t = np.linspace(25, 45, 1000)
```

```
plt.plot(ktemp, kgreen, '.')

#plt.plot(temp, green, '.', t, SHGsinc(t,*Popt))
plt.plot(t, SHGsinccos(t,23,34.5,1.7, np.pi/2))
#plt.plot(t, SHGsinc(t,60,34.2,2))
#plt.plot(t, SHGsinc(t,60,36,1.5))

plt.grid()
plt.xlabel('SHG Crystal Temp')
plt.ylabel('P532/P1064^2')
plt.rcParams["figure.dpi"] = 100
```

Double pass using prism (1973)

- Ref 10 of the previous paper, Gonzalez, Nieh, and Steier 1973 took air dispersion into account
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1077338

$$P_2 = K_0 P_1{}^2 \operatorname{sinc}^2\left(\frac{\Delta k l_c}{2}\right) \cos^2\left(\frac{\Delta k l_c}{2} + \frac{2\omega}{c}\, l\, \Delta n + \Delta\phi\right), \quad (1)$$

where

$K_0 \quad = 2\eta^3\omega^2\,(d^2 l_c{}^2/\pi W^2)$,
ω angular frequency of fundamental,
d nonlinear coefficient,
$\Delta n \quad = n(2\omega) - n(\omega)$—dispersion of air,
$\Delta\phi \quad = (\phi/2)(2\omega) - \phi(\omega)$ phase shifts due to mirror coatings,
W radius of fundamental beam,
$\Delta k \quad = k_1(\omega) - 2k_2(2\omega)$,
$\eta \quad = 377/n$.

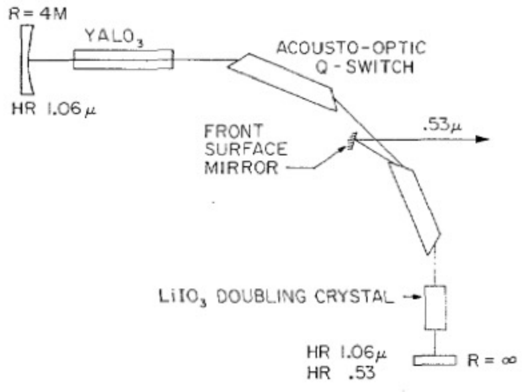


Fig. 1. Experimental configuration.

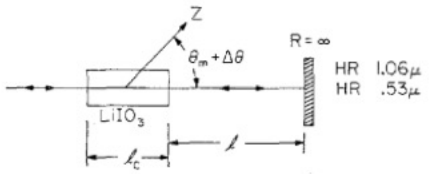


Fig. 2. Two-pass second-harmonic configuration.

Double pass using prism (1973)

- These guys saw similar behavior we saw. And the measurements don’t necessary match the model

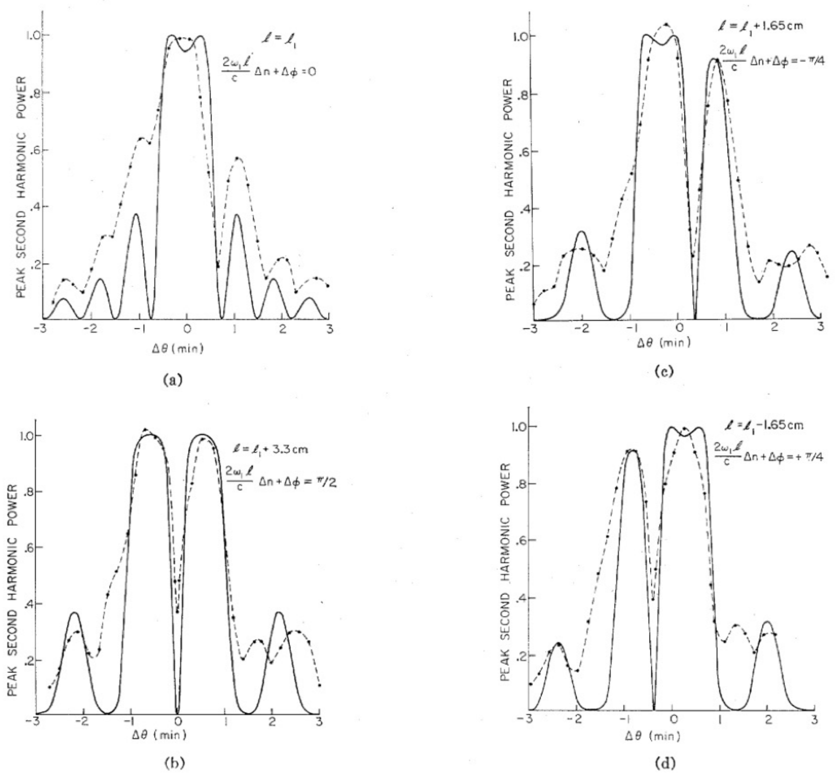


Fig. 3. Peak second-harmonic power versus crystal orientation.

Kato and Takaoka 2002

- Eq 1 in the 1973 paper is a sinc^2 function in terms of phase (delta K*l). K is related to n which relates to T. We need this equation in terms of temperature.
- Kato and Takaoka 2002 gives you n of ppktp at 1064 and dn/dT
- Kiyoshi Kato and Eiko Takaoka, "Sellmeier and thermo-optic dispersion formulas for KTP," Appl. Opt. **41**, 5040-5044 (2002) https://opg.optica.org/ao/abstract.cfm?URI=ao-41-24-5040

$$n_x^2 = 3.29100 + \frac{0.04140}{\lambda^2 - 0.03978} + \frac{9.35522}{\lambda^2 - 31.45571},$$
$$n_y^2 = 3.45018 + \frac{0.04341}{\lambda^2 - 0.04597} + \frac{16.98825}{\lambda^2 - 39.43799},$$
$$n_z^2 = 4.59423 + \frac{0.06206}{\lambda^2 - 0.04763} + \frac{110.80672}{\lambda^2 - 86.12171}, \quad (1)$$

This is PPKTP n at T=20C
This is f(a) in the Taylor series expansion

$$f(a) + \frac{f'(a)}{1!}(x-a).$$

(x-a) is (T-20)
Where T is the thing we want to sweep

$$\frac{dn_x}{dT} = \left(\frac{0.1717}{\lambda^3} - \frac{0.5353}{\lambda^2} + \frac{0.8416}{\lambda} + 0.1627\right) \times 10^{-5}(°C^{-1})\ (0.43 \le \lambda \le 1.58),$$
$$\frac{dn_y}{dT} = \left(\frac{0.1997}{\lambda^3} - \frac{0.4063}{\lambda^2} + \frac{0.5154}{\lambda} + 0.5425\right) \times 10^{-5}\ (0.43 \le \lambda \le 1.58),$$
$$\frac{dn_z}{dT} = \left(\frac{0.9221}{\lambda^3} - \frac{2.9220}{\lambda^2} + \frac{3.6677}{\lambda} - 0.1897\right) \times 10^{-5}\ (0.53 \le \lambda \le 1.57)$$
$$= \left(\frac{-0.5523}{\lambda} + 3.3920 - 1.7101\lambda + 0.3424\lambda^2\right) \times 10^{-5}\ (1.32 \le \lambda \le 3.53), \quad (2)$$

This is f'(a)

Delta K is defined in Leonardi et al (the Virgo SHG paper), page 3 bottom right.

on the phase mismatch $x = \Delta k L/2$. For crystals based on the quasi-phase-matching (QPM) technique, as in our PPKTP crystal, $\Delta k = 4\pi/\lambda_\omega(n_{2\omega} - n_\omega) - \Delta k_{QPM}$, $\Delta k_{QPM} = 2\pi/\Lambda$

deltaK and the sinc^2 function becomes

```
def deltaKz(T):
    lambdagr = 0.532
    lambdair = 1.064
    pollength = 8.99608

    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)

    deltaKz = (4*np.pi/(lambdair))*((ngr1+ngr2)-(nir1+nir2))-(2*np.pi/pollength)
    return deltaKz

def SHGsinccosair(T, A, n, phi):
    lcry = 10000 #crystal length 10000 um

    return A* np.sinc(deltaKz(T)*lcry/(2*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/2 + 2*n + phi)**2
```

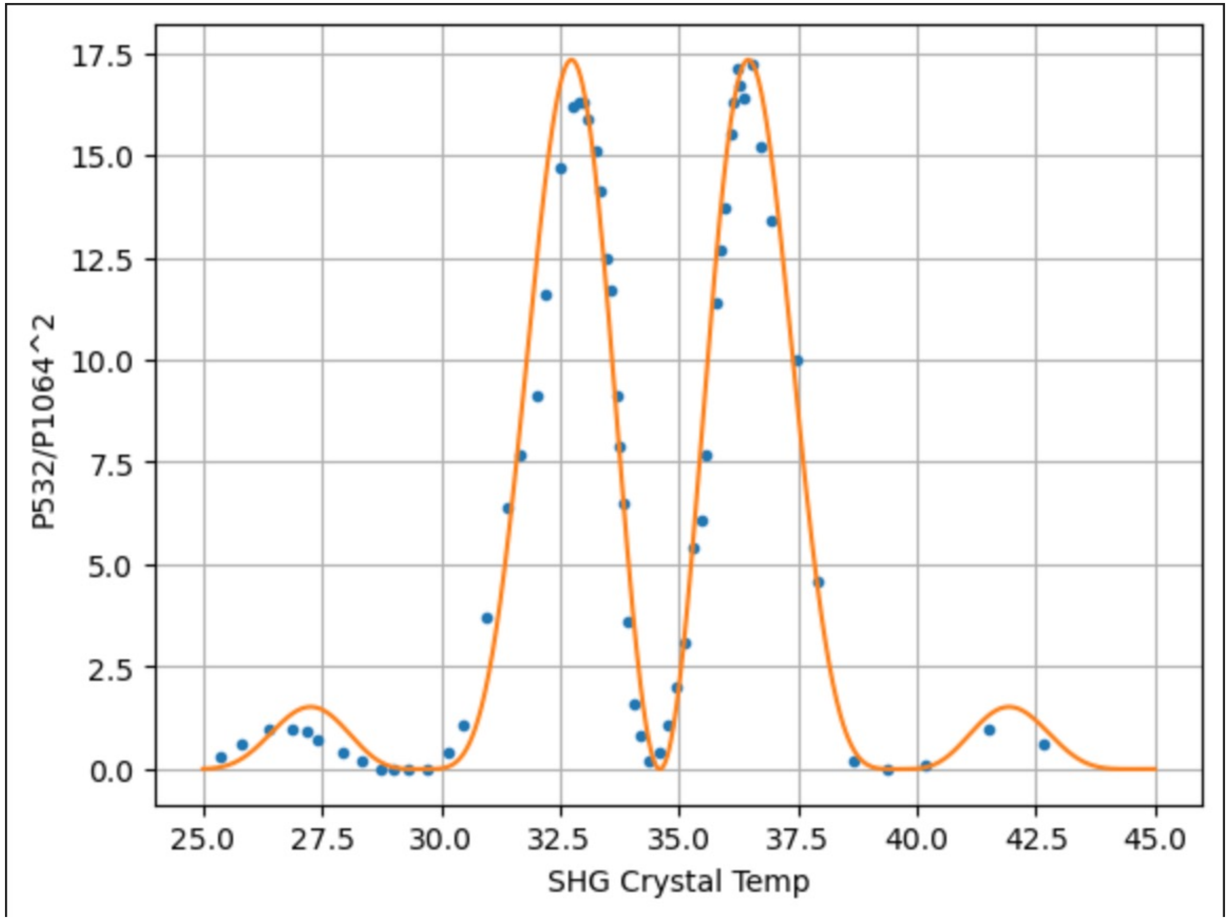
Fitting the actual data

- The function is super sensitive to the polling lenth. Raicol calims 9um polling length but that’s probably not exact.
- Karmeng’s data supposed to peak at at 34.6 C. This corresponds to a polling length = 8.99608 um (deltaK = 0 at this T).
- Add a phase mismatch of pi/2 and you have a self-destruc sinc^2 in the middle

Fitting the actual data

```
plt.plot(ktemp, kgreen, '.')

#plt.plot(temp, green, '.', t, SHGsinc(t,*Popt))
plt.plot(t, SHGsinccosair(t,33, 0, np.pi/2))


plt.grid()
plt.xlabel('SHG Crystal Temp')
plt.ylabel('P532/P1064^2')
plt.rcParams["figure.dpi"] = 100
```

✓ 0.1s



The nonlinear part

- Right now the unit of K0 makes no sense.
- This paper sounds more sensible
- https://www.sciencedirect.com/science/article/pii/S0030401897003386?ref=cra_js_challenge&fr=RR-1

Green and ultraviolet quasi-phase-matched second harmonic generation in bulk periodically-poled $KTiOPO_4$

A. Arie [a], G. Rosenman [a], V. Mahal [a], A. Skliar [a], M. Oron [b], M. Katz [b], D. Eger [b]

[a] *Department of Electrical Engineering – Physical Electronics, Faculty of Engineering, Tel Aviv University, 69978 Tel Aviv, Israel*

[b] *NRC, 81800 Soreq, Israel*

Received 20 March 1997; revised 12 June 1997; accepted 17 June 1997

$$\eta_{SH} = \frac{P_{2\omega}}{P_{\omega}} = \left(\frac{2\omega^2 d_{eff}^2 k_{\omega} P_{\omega}}{\pi n_{\omega}^2 n_{2\omega} \epsilon_0 c^3} \right) L h(B,\xi). \qquad (1)$$

Boyd-Kleinman factor

- G. D. Boyd and D. A. Kleinman, “Parametric interactions of focused Gaussian light beams,’
- I don’t have access to J. Appl Phys. Daniel sent one to me. Why is Adelaide Library so shit?
- Apparently there’s a thing call the focusing parameter l/b = 2.84 where l is the length of the crystal and b is the confocal parameter (2*Rayleigh length) and the Boyd-Kleinman factor h(B,l/b) is 1 ish in an optimal SHG.

JOURNAL OF APPLIED PHYSICS VOLUME 39, NUMBER 8 JULY 1968

Parametric Interaction of Focused Gaussian Light Beams

G. D. Boyd
Bell Telephone Laboratories, Incorporated, Holmdel, New Jersey
AND
D. A. Kleinman
Bell Telephone Laboratories, Incorporated, Murray Hill, New Jersey

(Received 5 February 1968)

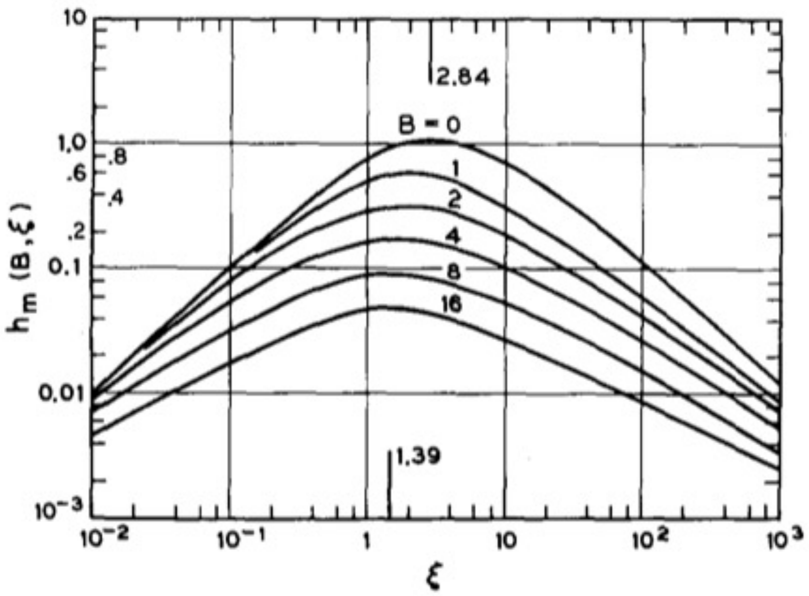


FIG. 2. SHG power (2.22) represented by the function $h_m(B, \xi)$ (2.29) for optimum phase matching as a function of focusing parameter $\xi = l/b$ for several values of double-refraction parameter $B = \rho(lk_1)^{1/2}/2$. Vertical lines indicate optimum focusing in the limits of small and large B.

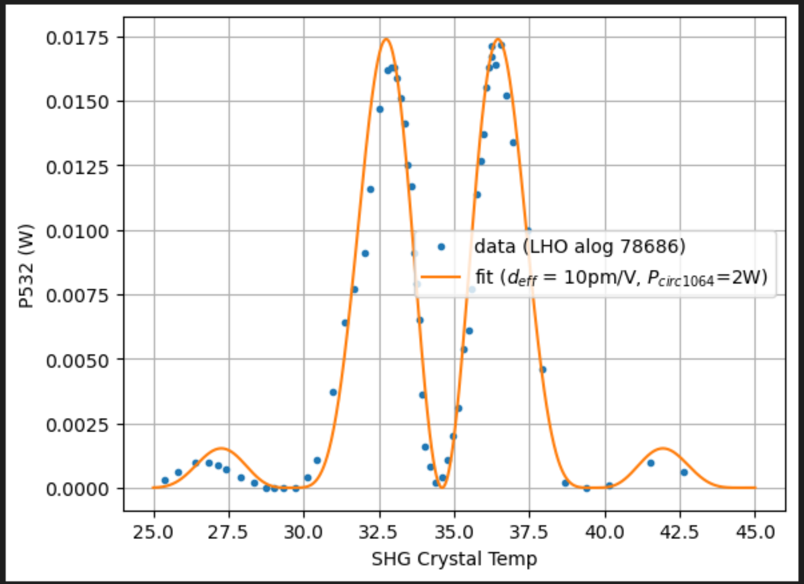
After everything was understood…

https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=78993

```python
def deltaKz(T):
    lambdagr = 0.532
    lambdair = 1.064
    pollength = 8.99608

    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)

    deltaKz = (4*np.pi/(lambdair))*((ngr1+ngr2)-(nir1+nir2))-(2*np.pi/pollength)
    return deltaKz

def SHGsinccosair(T, A, phi):
    lcry = 10000 #crystal length 10000 um

    return A* np.sinc(deltaKz(T)*lcry/(2*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/2 + phi)**2

def SHGsinccosKappa(T,deff,Pw, h, phi):
    lcry = 10000 #crystal length 10000 um
    lcry_m = 10e-3 #crystal length 10000 um
    nw = nir1+nir2
    n2w = ngr1+ngr2
    eta = 377/n
    lambda_m = 1064e-9
    c = 299792458
    f = c/lambda_m
    omega = 2*np.pi*f
    W = 140 #um
    kw = 2*np.pi/lambda_m
    eps = 8.8541878188e-12 #vacuum permittivity  C/(V*m)

    #d = 10e-6 #10pm/V Virgo SHG paper
    #Kappa = 2*eta**3*omega**2*(deff**2*lcry**2/(np.pi*W**2)) #1973 paper
    Kappa = ((2*omega**2*deff**2*kw*Pw**2)/(np.pi*nw**2*n2w*eps*c**3))*lcry_m*h


    return Kappa*np.sinc(deltaKz(T)*lcry/(2*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/2 + phi)**2
```
✓ 0.0s

with nonlinear coefficient

```python
plt.plot(ktemp, kgreen, '.')

#plt.plot(temp, green, '.', t, SHGsinc(t,*Popt))
plt.plot(t, SHGsinccosKappa(t, 10e-12, 2, 0.9, np.pi/2))


plt.grid()
plt.xlabel('SHG Crystal Temp')
plt.ylabel('P532 (W)')
plt.legend(['data (LHO alog 78686)', 'fit ($d_{eff}$ = 10pm/V, $P_{circ1064}$=2W)'],loc="best")
plt.rcParams["figure.dpi"] = 100
```
[691] ✓ 0.1s



Update Jul 11 24

- Fit to a single pass measurement
- Alog https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79017
- Added a factor of 4 to K0 in double peak case (double pass power = 4 times the single pass
- Need 1.5W circ to fit. Model suggests 1.7W circ with escape efficiency measurement taken into account
- Modulation depth unknown.

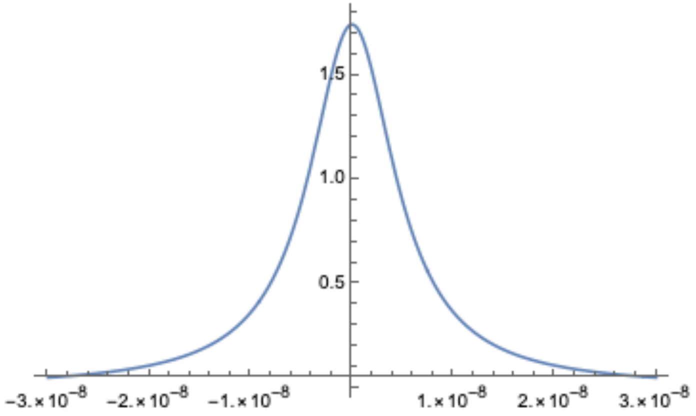
SHG Cavity Response (1064)

```
In[∗]:= ClearAll["Global`∗"]

In[325]:= R1 = .9;
R2 = 0.99850 - .014; (*Karmeng's escape efficiency measurement, alog 78571*)
(*R2 = 0.9985;*)
Rloss = R2 - 0.096;
T1 = 1 - R1;
T2 = 1 - R2;
t1 = √T1;
t2 = √T2;
r1 = √R1;
r2 = √R2;
rloss = √Rloss;

In[335]:= λcr = 1064 × 10^-9;
fcr = 0;
c = 299 792 458;
L = 0.05; (*cavity length*)
FSR = c/(2 L);
Fin = (Pi √(r1 r2))/(1 - r1 r2); (*based on Finesse Living Review*)
Finwithloss = (Pi √(r1 r2 rloss))/(1 - r1 r2 rloss)
FWHM = FSR/Fin;
FWHM2024 = (1.8 × 3 × 10^9)/54
FWHMwithloss = FSR/Finwithloss
Pin = 60 × 10^-3;
Ein = √Pin;

Out[341]= 26.2516

Out[343]= 1. × 10^8

Out[344]= 1.142 × 10^8

In[347]:= Ecirccr = (Ein t1)/(1 - r1 r2 E^(-2*I*(Pi*fcr/FSR + 2*Pi*δl/λcr)));
Etranscr = (Ein t1 t2 E^(-I*(Pi*fcr/FSR + 2*Pi*δl/λcr)))/(1 - r1 r2 E^(-2*I*(Pi*fcr/FSR + 2*Pi*δl/λcr)));
Ereflcr = -r1 Ein + t1 r2 Ecirccr E^(-2*I*(Pi*fcr/FSR + 2*Pi*δl/λcr));

Pcirc = Ecirccr Ecirccr*;
Ptrans = Etranscr Etranscr*;
Prefl = Ereflcr Ereflcr*;
```

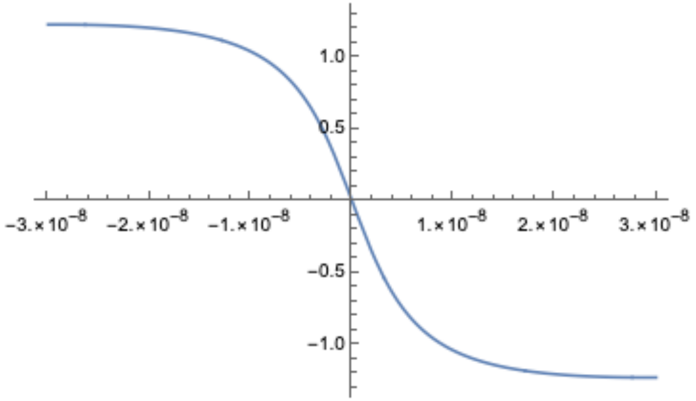
Circ

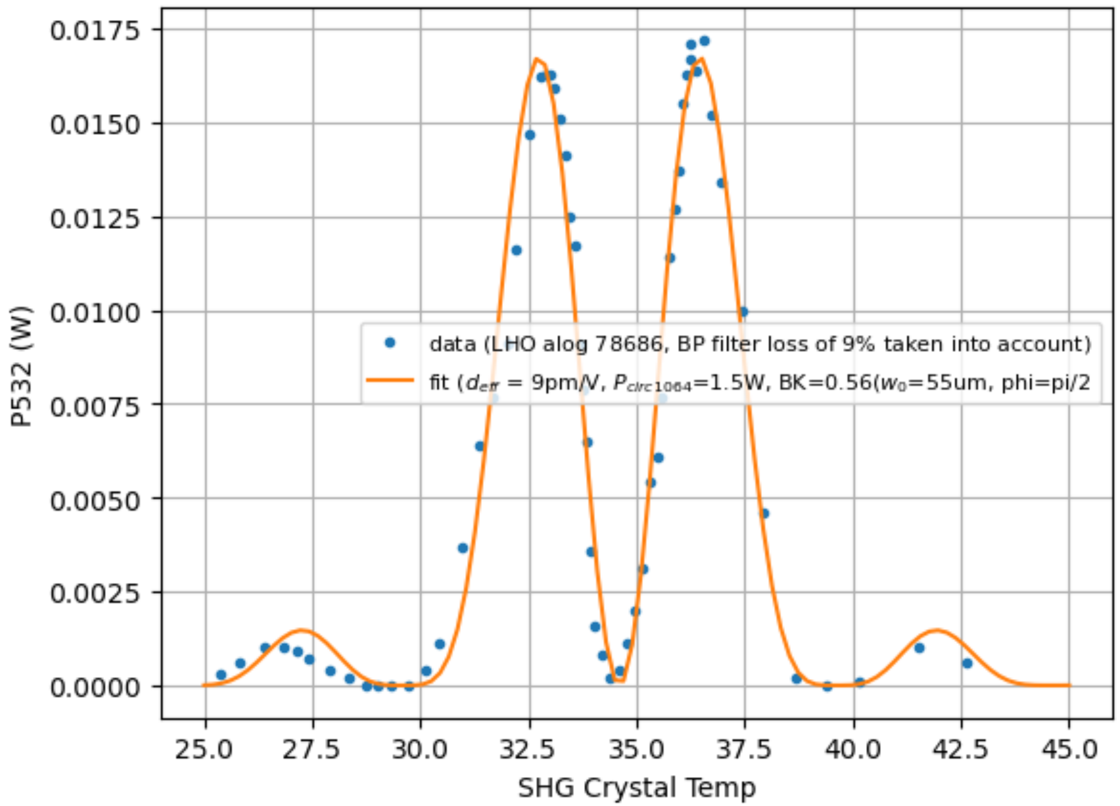
```
In[353]:= Plot[Pcirc, {δl, -3 × 10^-8, 3 × 10^-8}, PlotRange → All]
Plot[Arg[Ecirccr], {δl, -3 × 10^-8, 3 × 10^-8}, PlotRange → All]
```

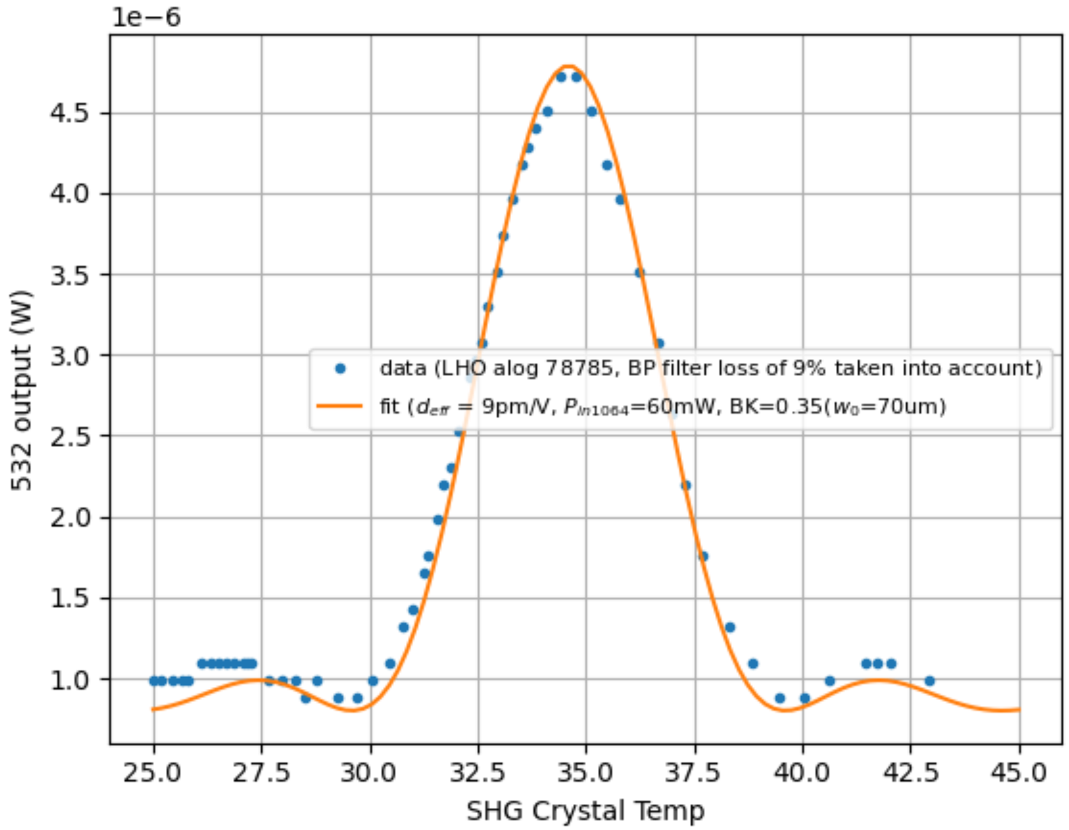
Out[353]=



Out[354]=







Different Boyd-Klienman factor because cavity vs no cavity

single pass sinc^2

```
lambdagr = 0.532
lambdair = 1.064
pollength = 8.99608
lcry = 10000 #crystal length 10000 um
lcry_m = 10e-3 #crystal length 10000 um

def deltaKz(T):
    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)

    deltaKz = (4*np.pi/(lambdair))*((ngr1+ngr2)-(nir1+nir2))-(2*np.pi/pollength)
    return deltaKz

#single pass SHG
def SHGsincsingle(T,deff,Pw, h, DN):
    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)
    nw = nir1+nir2
    n2w = ngr1+ngr2
    #eta = 377/n
    lambda_m = 1064e-9
    c = 299792458
    f = c/lambda_m
    omega = 2*np.pi*f
    W = 140 #um
    kw = 2*np.pi/lambda_m
    eps = 8.8541878188e-12 #vacuum permittivity  C/(V*m)

    #d = 10e-6 #10pm/V Virgo SHG paper
    #Kappa = 2*eta**3*omega**2*(deff**2*lcry**2/(np.pi*W**2)) #1973 paper
    Kappa = ((2*omega**2*deff**2*kw*Pw**2)/(2*np.pi*nw**2*n2w*eps*c**3))*lcry_m*h


    return Kappa*np.sinc(deltaKz(T)*(lcry)/(4*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/4)**2 +DN
```

Double Pass Sinc^2

+ Code + Markdown

```
lambdagr = 0.532
lambdair = 1.064
pollength = 8.99608
lcry = 10000 #crystal length 10000 um
lcry_m = 10e-3 #crystal length 10000 um
l = 20000 # distance between crystal and back mirror

def deltaKz(T):
    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)

    deltaKz = (4*np.pi/(lambdair))*((ngr1+ngr2)-(nir1+nir2))-(2*np.pi/pollength)
    return deltaKz

def SHGsinccosair(T, A, phi):

    return A* np.sinc(deltaKz(T)*lcry/(2*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/2 + phi)**2

#double pass SHG
def SHGsinccosKappa(T,deff,Pw, h, phi):
    ngr1 = np.sqrt(4.59423+(0.06206/(lambdagr**2-0.04763))+(110.80672/(lambdagr**2-86.12171)))
    ngr2 = (((0.9221/lambdagr**3)-(2.922/lambdagr**2)+(3.6677/lambdagr)-0.1897)*1e-5)*(T-20)
    nir1 = np.sqrt(4.59423+(0.06206/(lambdair**2-0.04763))+(110.80672/(lambdair**2-86.12171)))
    nir2 = (((0.9221/lambdair**3)-(2.922/lambdair**2)+(3.6677/lambdair)-0.1897)*1e-5)*(T-20)
    nw = nir1+nir2
    n2w = ngr1+ngr2
    #eta = 377/n
    lambda_m = 1064e-9
    c = 299792458
    f = c/lambda_m
    omega = 2*np.pi*f
    W = 140 #um
    kw = 2*np.pi/lambda_m
    eps = 8.8541878188e-12 #vacuum permittivity  C/(V*m)

    #d = 10e-6 #10pm/V Virgo SHG paper
    #Kappa = 2*eta**3*omega**2*(deff**2*lcry**2/(np.pi*W**2)) #1973 paper
    Kappa = 4*((2*omega**2*deff**2*kw*Pw**2)/(np.pi*nw**2*n2w*eps*c**3))*lcry_m*h #extra factor of 4 in power dur to double pass


    return Kappa*np.sinc(deltaKz(T)*lcry/(2*np.pi))**2 * np.cos(  (deltaKz(T)*lcry)/2 + phi)**2
```

Beam waist

- Used Finesse to determined to beam waist inside of the SHG cavity with the crystal inside given known parameters
- A beam waist is required to compute Boyd-Klienman constant
- Single pass waist with or without the crystal is differ by 2um. Negligible.

```
kat = finesse.Model()
kat.parse(
    """
    # Add a Laser named L0 with a power of 1 W.
    l L0 P=1

    # Space attaching L0 <-> m1 with length of 0 m (default).
    s s0 L0.p1 m1.p1 L=1

    # Input mirror of cavity.
    m m1 R=0.9 T=0.09999999999999998 Rc=-0.025

    # m1 to c1 (crystal)
    s s1 m1.p2 c1.p1 L=0.0185

    # PPKTP crystal input c1
    m c1 R=0 T=1

    # PPKTP length
    s crystal c1.p2 c2.p1 L=0.01 nr=1.8298794813610717

    # PPKTP crystal output c2
    m c2 R=0 T=1

    # c2 to m2
    s s2 c2.p2 m2.p1 L=0.0185

    # End mirror of cavity.
    m m2 R=0.9985 T=.001499 Rc=0.025


    # Power detectors on reflection, circulation and transmission.
    pd refl m1.p1.o
    pd circ m2.p1.i
    pd trns m2.p2.o

    # Scan over the detuning DOF of m1 from -180 deg to +180 deg with 400 points.
    xaxis(m1.phi, lin, -180, 180, 10000)
    """
)

# Add a cavity object
kat.add(components.Cavity("SHG", kat.m1.p2.o, kat.m2.p1.i))
```

```
[43]
print(kat.SHG.draw())
```

```
o m1.p2.o
└─o c1.p1.i
  └─o c1.p2.o
    └─o c2.p1.i
      └─o c2.p2.o
        └─o m2.p1.i
          └─o m2.p1.o
            └─o c2.p2.i
              └─o c2.p1.o
                └─o c1.p2.i
                  └─o c1.p1.o
                    └─o m1.p2.i
```

```
(function) SHG: Any
[96]
kat.SHG.q
```

```
array([<BeamParam (w0=55.038 um, w=141.78 um, z=-21.232 mm, nr=1, λ=1.064 um) at 0x16e4f9990>,
       <BeamParam (w0=55.038 um, w=141.78 um, z=-21.232 mm, nr=1, λ=1.064 um) at 0x16e4f9ab0>],
      dtype=object)
```

```
(function) parse: Any
[46]
# De
kat.parse("cavity cavity1 source=m1.p2.o via=m2.p1.i priority=1")

# Plot the beam trace, starting from cavity the eigenmode.
tsy = finesse.tracing.tools.propagate_beam(
    to_node=kat.m1.p2.i, from_node=kat.m2.p1.o, direction="y"
)
tsy.plot();
```