Self-focusing effect in Nonlinear Media

Arnav Metrani

IISER Mohali

April 2025

Objectives of the talk

- Give a theoretical overview of the intensity-dependent refractive index and some results regarding self-focusing.
- Theoretical explanation of the implementation.
- Explanation of the code.
- Examples generated.
- Simulations vs Reality
- Cross-checking with literature.
- Limitations.
- Future directions.

Derivation of n_2

If we consider a system with $\chi^{(1)}$ as the linear term and $\chi^{(3)}$ as the nonlinear term, then:

$$\overrightarrow{P} = \overrightarrow{P}^{(1)} + \overrightarrow{P}^{(3)}, P^{(3)} = \epsilon_0 \chi^{(3)} E^3, E = E_0 e^{i\omega t} + E_0^* e^{-i\omega t}$$

$$P^{(3)} = \epsilon_0 \chi^{(3)} [E_0^3 e^{3i\omega t} + E_0^* e^{-3i\omega t} + 3E_0^2 E_0^* e^{i\omega t} + 3E_0 (E_0^*)^2 e^{-i\omega t}]$$

$$\Rightarrow \epsilon_0 \chi^{(3)} [E_0^3 e^{3i\omega t} + 3E_0^2 E_0^* e^{i\omega t}] + \text{C.C.}$$

$$P = \epsilon_0 [\chi^{(1)} + 3\chi^{(3)} (E_0 E_0^*)] E_0 e^{i\omega t} + \epsilon_0 \chi^{(3)} E_0^3 e^{3i\omega t} + \text{C.C}$$

We now look at only the ω component since the 3ω component is phase mismatched wrt. the input beam.

Derivation of n_2

For the ω term:
Let $\overline{\chi} = \chi^{(1)} + 3\chi^{(3)}(|E_0|^2)$. $n^2 = 1 + \overline{\chi}$

$$n^2 = (n_0^2 + n_2^2 I^2 + 2n_0 n_2 I) = 1 + \chi^{(1)} + 3\chi^{(3)}|E_0|^2$$

Since $n_2 \ll 1$, we drop the n_2^2 term. $I = \frac{1}{2} n_0 \epsilon_0 c \langle E^2 \rangle, \langle E^2 \rangle = 2|E_0|^2$

$$n_2 = \frac{3}{2n_0^2 \epsilon_0 c} \chi^{(3)}$$

[This will not be referred to in the rest of the presentation.]

Derivation of n_2

(There is some discrepancy with respect to the factor, see **[Butcher and Cotter]**).

TABLE 1. Notation for K in $\delta n = K|\mathscr{E}|^2$, where $\mathscr{E}$ is the peak amplitude of a linearly polarized field. Equation numbers are for cited references

Author (reference)	K
This work	$\frac{1}{2}n_2 = \epsilon_2/4n_0 = 2\pi\eta/n_0$
Chiao *et al.*[9]	n_2 in eqn. (1)
Chiao *et al.*[9]	$\epsilon_2/2n_0$ in eqn. (3)
Chiao *et al.*[9]	$\epsilon_2/4n_0$ subsequent eqns.
Talanov[10]	$\epsilon'/8n_0$
Kelley[11]	n_2', $\epsilon_2'/2n_0$
Akhmanov *et al.*[8]	n_2, $\epsilon_2/2n_0$
Shen[55]	$\frac{1}{2}K_0\lambda_0$, $\frac{1}{3}K_x\lambda_0$
Hellwarth[34]	$2\pi\eta/n_0$
Dawes and Marburger[47]	$\epsilon_2/4n_0$
C. C. Wang[110]	$6\pi[\chi_3^{1122} + \chi_3^{1212} + \chi_3^{1221}]/n_0 = 6\pi\chi_3^{1111}/n_0 = \pi(2A + B)/n_0$
Wagner *et al.*[46]	$\epsilon_2'/2n_0$ (mks units)

Figure: Different definitions. (From **[Marburger(1975)]**)

Implementation

The phase changes due to the linear and non-linear term are not sptaially uniform, thus the overall effect is of a divergence due to the diffraction and convergence due to n_2. We begin with the Helmholtz equation. The spatial Helmholtz equation is given by $(\nabla^2 + k^2)E(x, y, z) = 0$,

$$k = \frac{n\omega}{c}, k_0 = \frac{n_0\omega}{c}$$

$$n(I) = n_0 + n_2 I = n_0 + n_2 \cdot \frac{1}{2} n_0 \epsilon_0 c |E|^2$$

$$k^2 = k_0^2 (1 + \frac{2n_2 I}{n_0})$$

$$\Rightarrow (\nabla^2 + k_{\mathsf{vac}}^2 n^2(I))E(x, y, z) = 0$$

Implementation

Paraxial approximation: We assume that the beam is well collimated and travels along a fixed wavevector. Thus, we assume $E(x, y, z) = A(x, y, z)e^{ik_0 z}$, where the phase variation is isolated and amplitude is slowly varying. [We could take $e^{ik(I(x,y,z))z}$ instead but this would be difficult to model, ignored for now.]
Subbing gives:

$$\nabla_T^2 A + (\frac{\partial^2 A}{\partial z^2} + 2ik_0 \frac{\partial A}{\partial z} - k_0^2 A) + k_0^2 A = 0$$

We take $\frac{\partial^2 A}{\partial z^2} \ll \frac{\partial A}{\partial z}$, thus getting:

$$\nabla_T^2 A + \partial z^2 + 2ik_0 \frac{\partial A}{\partial z} = 0$$

Implementation

Similarly, from $(\nabla^2 + k_{\text{vac}}^2 n^2(I))E = 0$ we get:

$$\nabla_T^2 \overrightarrow{E} + 2ik_0 \frac{d\overrightarrow{E}}{dz} + (k_0^2 \frac{2n_2 I}{n_0})\overrightarrow{E} = 0$$

Solving this nonlinear equation is generally quite complicated and computationally intensive.
To tackle this, we use the Beam Propagation Method.

Beam Propagation Method (BPM)

We break up the propagation length L into N steps. $\Delta z = \frac{L}{N\text{steps}}$
Then, we find $E(x, y, z)$ at $z = 0, \Delta z, ..., L$.
Let E at the mth step be denoted by $E_m = E(x, y, z_m)$.
We need to find E_{m+1} from E_m.
We split the process into two: The linear diffraction part and the nonlinear part. We then apply each process one after the other for the same step, assuming that within one step there is no appreciable change.

Linear diffraction part

Let $f(x)$ be a 1D function. $\mathcal{F}(f(x)) = F(k) = \int_{-\infty}^{\infty} f(x)e^{-ikx}dx$.

$$\mathcal{F}(f'(x)) = \int_{-\infty}^{\infty} \frac{df(x)}{dx} e^{-ikx}dx = f(x)e^{-ikx}|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} ikf(x)e^{-ikx}dx$$

.

If the function disappears at infinities, then $\mathcal{F}(f'(x)) = ikF(k)$. This makes it much easier to solve such equations. In this step we turn off the nonlinear term.

Taking the FT of $E(x, y, z)$ in x and y and subbing gives:

$$\frac{\partial E(k_x, k_y, z)}{\partial x} = \frac{-i(k_x^2 + k_y^2)}{2k_0} E(k_x, k_y, z)$$

Let $E = C(k_x, k_y)e^{\alpha z}$, then $E_{m+1}(k_x, k_y) = E_m(k_x, k_y)e^{\alpha(z - z_m)}$. Thus, we get:

$$E_{m+1}(k_x, k_y) = E_m(k_x, k_y)e^{\frac{-i(k_x^2+k_y^2)}{2k_0}\Delta z}$$

Linear diffraction part

We then take the IFT to come back to the spatial domain.

$$E_{\text{eff}}(x, y, z_{m+1}) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E_{\text{eff}}(k_x, k_y, z_{m+1}) e^{i(k_x + k_y)} dk_x \; dk_y$$

Nonlinear term

Here we turn off the diffraction term.

$$I(x, y, z_m) = \frac{1}{2} n_0 \epsilon_0 c |E_m|^2 \ , \ \Delta n = n_2 I$$

$$\frac{\partial E}{\partial z} = i k_0 \frac{\Delta n}{n_0} E$$

Thus, there is a phase shift of $\Delta \phi = k_0 \frac{\Delta n(x,y,z_m)}{n_0} \Delta z$ from z_m to z_{m+1}.

$$\Delta \phi = \frac{1}{2} k_0 \epsilon_0 c |E_m|^2 \Delta z$$

.

Code parameters

- λ_0: Wavelength of the propagating CW laser.
- n_0, n_2: Linear refractive index and the NL term.
- w_0: Beam waist- Distance where intensity falls to $\frac{1}{e^2}$ of peak intensity.
- P_{in}: Input power.
- L: Propagation length.
- num_steps: Number of uniform break-ups of the propagation length.
- Nx, N_y: Number of discrete elements in the x and y plane (akin to number of pixels).
- dx, dy: Size of each element (akin to size of pixel)- Smaller dx,dy gives finer spatial resolution.

Code explanation

```
k0 = 2 * np.pi * n0 / lambda0
dz = L / num_steps
```

This defines the wavevector k_0 and the step size to be used in the split-step method.

```
x = np.arange(-Nx/2, Nx/2) * dx
y = np.arange(-Ny/2, Ny/2) * dy
X, Y = np.meshgrid(x, y)
```

Generates an array that spans the x and y axis, and are scaled by dx and dy respectively.

Code explanation

It creates all possible combinations of (x, y) and is needed to evaluate the intensity, which is a function of x and y. For example:
$x = [1, 2, 3, 4]$,$y = [100, 200, 300]$ This will give an output of:

$$X = [[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]$$

$$Y = [[100, 100, 100, 100], [200, 200, 200, 200], [300, 300, 300, 300]]$$

We see $(X_{ij}, Y_{ij}) = (x_j, y_i)$

Code explanation

```
E0 = np.sqrt(4 * P_in / (np.pi * w0**2 * n0 * c * epsilon))
E_field = E0 * np.exp(-(X**2 + Y**2) / w0**2)
```

The relation b/w E_0 and $\mathsf{P_{in}}$ is for a Gaussian beam. [$\sqrt{2}$ should have be taken instead since RMS field wince we are dealing with time averages and $\langle \cos^2(\omega t) \rangle = 1/2$, but this gives a factor of 2 in z_{SF}. Not sure why this is not considered.] Here $\mathsf{E_{Field}}$ gives the initial transverse beam profile, rest gets modelled by the code.

$\mathsf{E_{Field}}$ is a 2D array where each point (x,y) is associated with a value.

For example let $f(x, y) = x^2 + y^2$:

```
X = [[-1,  0,  1],             Y = [[-1, -1, -1],
     [-1,  0,  1],                  [ 0,  0,  0],
     [-1,  0,  1]]                  [ 1,  1,  1]]

X^2 + Y^2 = [ [2, 1, 2],
              [1, 0, 1],
              [2, 1, 2] ]
```

Code explanation

```
kx = 2 * np.pi * np.fft.fftfreq(Nx, d=dx)
ky = 2 * np.pi * np.fft.fftfreq(Ny, d=dy)
Kx, Ky = np.meshgrid(kx, ky)
```

np.fft.fftfreq(N,d) generates an array from 0 to $N-1$ of

$$[0, \frac{1}{N \cdot d}, \frac{2}{N \cdot d}, ..., \frac{\lceil N/2-1 \rceil}{N \cdot d}, -\frac{\lceil N/2-1 \rceil}{N \cdot d}, ..., \frac{-1}{N \cdot d}]$$

2π term is added to convert k to angular units rad/m.
Eg. if $x = 1$ cm, then k_x repeats every 1 cm.
The total spatial length is $N \cdot d$, we assume the wave "wraps around" beyond this (periodic BC assumed).
From this, we see that reducing d gives us a smaller spatial length, which increases the overall resolution of that patch. However this increases the spacing in the k plane.
Conversely, increasing d gives us a larger POV and better frequency resolution.

Code explanation

```
intensity_profiles = []
z_pos = np.linspace(0, L, num_steps + 1)
inter_indices = np.linspace(0, num_steps, 5, dtype=int)
inter_profiles = []
```

"intensity_profiles" stores the intensity at each step. "z_positions" creates an array with the step spacing Last two lines are specifically for intermediate points.

Code explanation

```
for i in range(num_steps + 1):
    intensity_profiles.append(np.abs(E_field)**2)
```

Within the loop:
Saves the intensity profile.

```
if i in inter_indices:
    inter_profiles.append((z_positions[i], np.copy(np.abs(E_field)**2)))
```

Creates an independent/non-referential copy of the intensity profile at the intermediate steps via np.copy.

Code explanation

```
E_field_fft = np.fft.fft2(E_field)
```

Generates the 2D array $E(k_x, k_y)$ from the FT of the 2D array $E(x, y)$.

```
prop = np.exp(-1j * dz / (2 * k0) * (Kx**2 + Ky**2))
E_field_diff_fft = E_field_fft * prop
```

Implements the propagator and inverse FT.

```
intensity = 0.5 * n0 * c * epsilon * np.abs(E_field_diff)**2
delta_n = n2 * intensity
delta_phi_nl = (k0 / n0) * delta_n * dz
```

Finds $I_{\mathrm{m}+1}$ and obtains the phase shift.

Code explanation

```
E_field_nonlinear = E_field_diff * np.exp(1j * delta_phi_nl)
E_field = E_field_nonlinear
```

Find the final expression and update the field expression.
End of loop.

Code explanation

```
plt.imshow(intensity_profiles[0], extent=[x.min()*1e6, x.max()*1e6,
y.min()*1e6, y.max()*1e6], origin='lower', cmap='jet')
```

Provides the colour scheme for the 2D plot. 2D array is chosen (here it is the initial profile), cmap sets the colour, extents are chosen, and the (0,0) of the array begins at lower-left.

Code explanation (misc.)

```
x_integ_intensity_1D = np.sum(current_intensity_profile_2D, axis=0)
x_integ_intensity_profiles.append(x_integ_intensity_1D)
```

$I_{int}(x) = \int I(x,y)dy \simeq \sum_y I(x,y)$

```
output_intensity_profile = intensity_profiles[-1]
peak_intensity_output = np.max(output_intensity_profile)
e_squared_intensity_level = peak_intensity_output / np.exp(2)
```

Picks out the max.intensity from the output profile and divides by e^2.

Code explanation (misc.)

```
center_y_index = Ny // 2
intensity_x_slice = output_intensity_profile[center_y_index, :]
center_x_index = Nx // 2
```

It then takes a horizontal slice through $y = 0$.

```
indices_x_waist_left = np.where(intensity_x_slice[:center_x_index]
<= e_squared_intensity_level)[0]
indices_x_waist_right = np.where(intensity_x_slice[center_x_index:]
<= e_squared_intensity_level)[0]
```

np.where find elements satisfying the conditional.

Code explanation (misc.)

```
if indices_x_waist_left.size > 0 and indices_x_waist_right.size > 0:
    waist_x_left_index = indices_x_waist_left[-1]
    waist_x_right_index = center_x_index + indices_x_waist_right[0]
    waist_x_radius = (x_coords[waist_x_right_index] -
    x_coords[waist_x_left_index]) / 2.0
```

Calculates the beam waist. Same is done for y and then averaged.

```
axis_intensity=current_intensity_profile_2D[Ny//2,Nx//2]
axis_intensity_values.append(axis_intensity)
```

Stores the intensity at $(0, 0)$ for later plotting vs z.

Formulae

These are specifically for Gaussian beams- For a simple two slit system separated by y, path difference at (r, θ) measured from the centre is $\Delta r = y \sin(\theta)$. The diffraction angle for a circular aperture is the angle of the first minima, given by $\theta_d \sim 0.61 \frac{\lambda/n_0}{w_0}$, obtained via Bessel functions. Fermat's principle shows that at the focus all rays arriving at the focal point must have the same optical path length (neglecting diffraction). From this we get:

$$(n + \Delta n) z_{\mathsf{SF}} = n_0 z_{\mathsf{SF}} / \cos(\theta)$$

Equating them gives:

- Critical power for self-focusing: $P_C \simeq \frac{\pi D^2 I}{4} = \frac{\pi (0.61)^2 \lambda_0^2}{8 n_0 n_2}$

Formulae

Diffraction length is the propagation distance at which the area enclosed within the beam waist doubles.

- Diffraction length= $\frac{\pi n_0 w_0^2}{\lambda_0}$.
- Self-focusing point with diffraction=$z_{\mathsf{SF}} = \frac{\pi n_0 w_0^2}{\lambda_0} \frac{1}{\sqrt{\frac{P}{P_{\mathsf{C}}} - 1}} = \frac{z_{\mathsf{D}}}{\sqrt{\frac{P}{P_{\mathsf{C}}} - 1}}$

  (The formula in Boyd has π missing.) Obtained from **[CREOL College Lecture notes]**.

Example

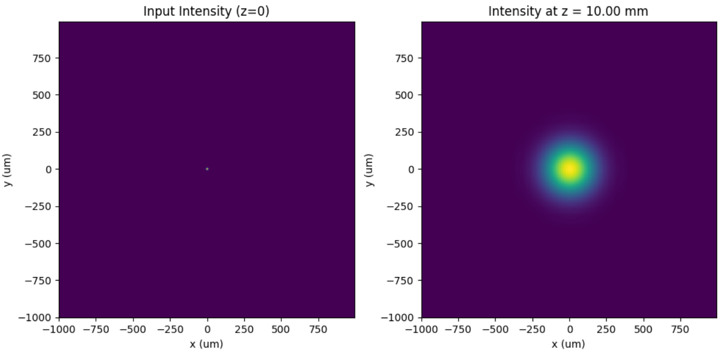


Figure: Test graph. No scale.

This is a test graph with only diffraction to see if the code is working.
$\lambda_0 = 1$ micron, $w_0 = 100$ micron, $n_0 = 1.5, N = 256$

Examples

We start with $\lambda = 1030$ nm, $n_0 = 1.45, n_2 = 3 \cdot 10^{-20}, w_0 = 50\ \mu m$, $\mathsf{P}_{\mathsf{in}} = 10$ MW, 1000 steps, $N_x = N_y = 1024, dx = dy = 2 \cdot 10^{-6}/16$

Examples

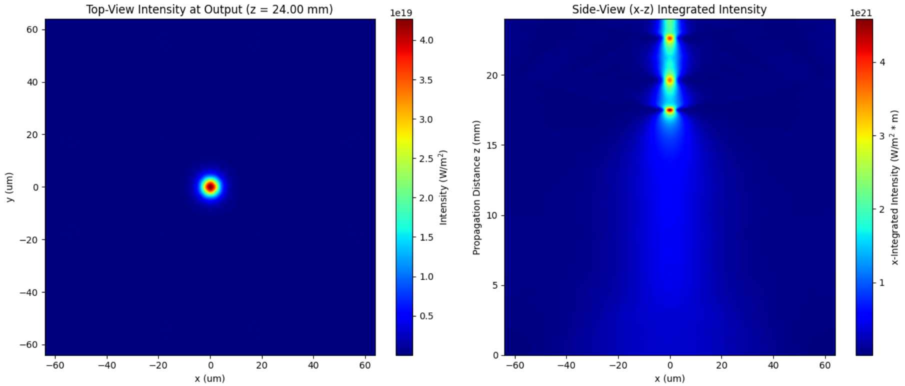


Figure: Output (assuming E_{RMS}) .

Examples

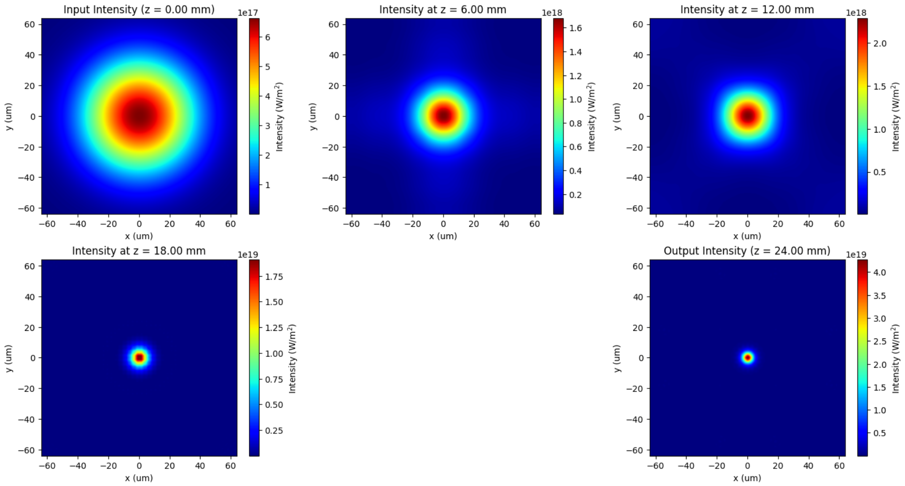


Figure: Output (assuming E_{RMS}).

Examples

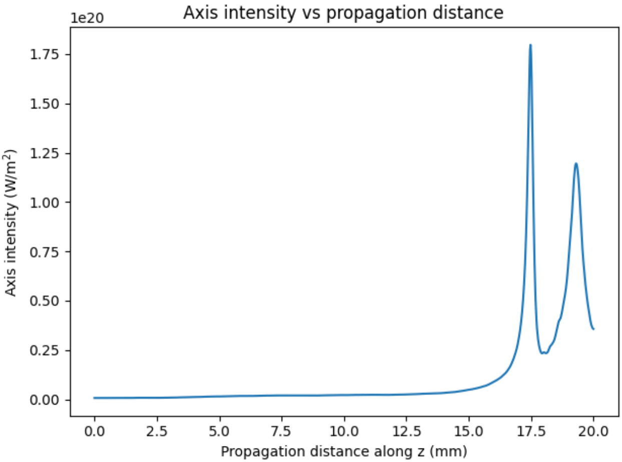


Figure: Central intensity vs propagation(assuming E_{RMS}).

Examples

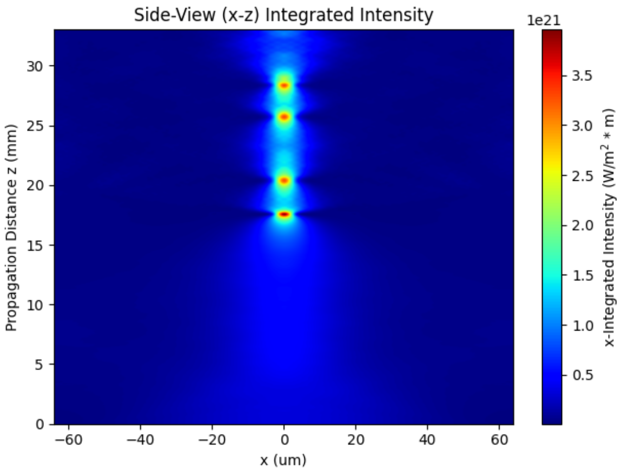


Figure: For longer propagation(assuming E_{RMS}).

Examples

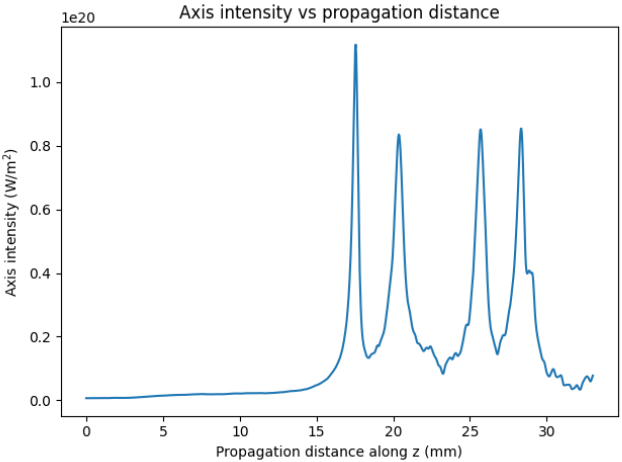


Figure: For longer propagation (assuming E_{RMS}).

Examples

- $\mathsf{P_C} = 3.88$ MW
- We obtain $z_D = 11.05$ mm

Plugging in gives $z_{\mathsf{SF}} = 17.6$ mm. However we see that the self-focusing point is reached before z_{SF} in our simulations. (Further analysis shows this occurs roughly at 17.45 mm). At this point the beam radius is roughly $2.62\ \mu m$.
This is roughly in line with what is seen in numerical simulations of self-focusing. We also see several focusing and defocusing cycles before a final dispersion, and the later foci are shifting wrt. to the propagation length chosen.
This is unexpected.

Examples

If we do not assume E_{RMS}:

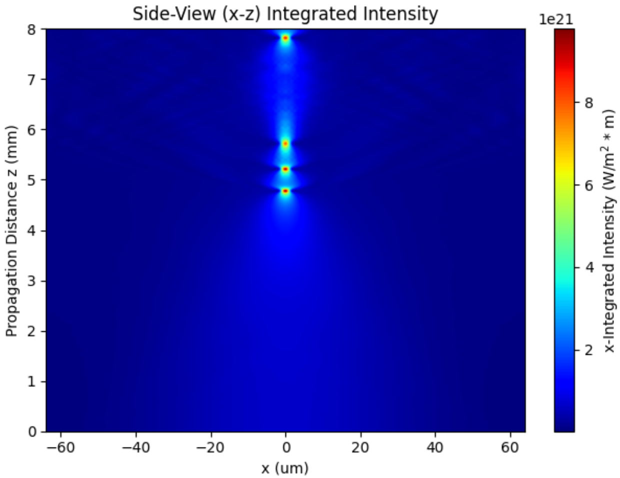


Figure: Without RMS.

Examples

If we set $w_0 = 70\ \mu m$, we expect $z_{\text{SF}} = 17$ mm

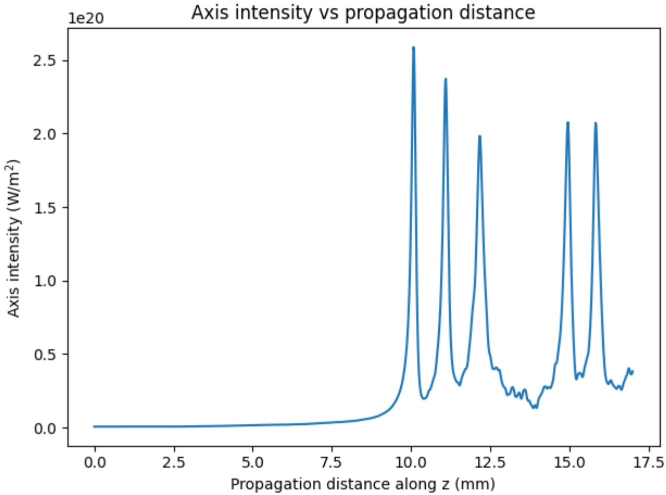


Figure: Without RMS.

What happens physically at z_{SF}?

From **[Self-focusing: Past and Present]**:

> *In air, in particular, the critical power is approximately 2 GW for a diffraction limited Gaussian beam[35]. When the intensity approaches* 10^{13} *to* 10^{14} *W/cm*^{2} *, multiphoton ionization occurs. The resultant intensity-dependent reduction of the refractive index arising from the consequent underdense plasma can stabilize the beam.*

What happens physically at z_{SF}?

The singularity is obviously non-physical. The focus-defocus cycles are not.

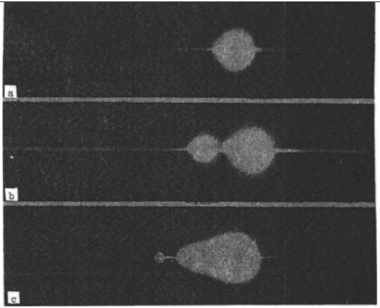


Fig. 5.5 Pattern of scattering of ruby laser radiation (rectangular pulse) in TF-105 glass for various incident powers P_i : **(a)** $P_i \approx P_d$, **(b)** $P_i = 3P_d$, and **(c)** $P_d \approx 6P_d$, where P_d is the laser-induced damage threshold. The distance between scattering center in part (c) is $\approx$5 mm. Bright scattering halos are seen around the damage points, as well as a weak trace of the ordinary scattering from the glass. The laser beam propagates from left to right

Figure: From **[Self-focusing: Past and Present]**.

What happens physically at z_{SF}?

It arises as a direct byproduct of the breakdown of assumptions taken wrt. Helmholtz equation. In this region the nonparaxiality of the Helmholtz equation counteracts the focusing and gives rise to the cycles. This is shown in **[Fibich(1996)]**.

The paper starts from the full Helmholtz equation, parametrises the nonparaxiality and obtains the cycles and the final dispersion.

The paper also models "non-adiabatic" results and shows that even under super-critical laser power we can have dispersion.

What happens when we simulate at $z \to z_{\mathsf{SF}}$?

It is astounding to see a "forcibly" paraxial equation (NLS) still give rise to cycles and a final dispersion.
Before coming to their appearance, their shifting depending on the propagation length might be due to the following factors:

- The beam undergoes severe divergence beyond the first focal point, and artifacts are visible in the xz plane. The total intensity of each z-slice is almost constant (not plotted), so the rays that leak out of the L_x, L_y study are coming back and interfering due to the periodic boundary conditions of FFT.
- N_{steps} was kept constant while the propagation length was changed. As a result the Δz also changed, leading to different extents of phase change in each slice. This effect may have built up.

What happens when we simulate at $z \to z_{\mathsf{SF}}$?

The cycles could occur due to transverse numerical instabilities due to discretization in the FT-IFT process.

$$\nabla_T^2 \overrightarrow{E} + 2ik_0 \frac{d\overrightarrow{E}}{dz} + (k_0^2 \frac{2n_2 I}{n_0})\overrightarrow{E} = 0$$

If we assume a uniform plane wave solution, then transverse Laplacian is zero.

$$2ik_0 \frac{\partial E}{\partial z} = \gamma |E|^2 E$$

$$E(z) = A_0 e^{i\phi z}, \phi = \frac{\gamma |A_0|^2}{2k_0}$$

What happens when we simulate at $z \to z_{\text{SF}}$?

If we now take $E = (A_0 + \alpha(x, y, z))e^{i\phi z}$ and sub into the full equation:

$$2ik_0[i\phi(A_0 + \alpha) + \frac{d\alpha}{dz}] + \nabla_T^2 \alpha + \gamma |A_0 + \alpha|^2 (A_0 + \alpha) = 0$$

Which simplifies to:

$$2ik_0 \frac{d\alpha}{dz} + \nabla_T^2 \alpha + 2\gamma |A_0|^2 \alpha + \gamma A_0^2 \alpha^* = 0$$

If we consider FT-IFT as a mechanism for perturbation then there will be fluctuations in α along z, which may grow exponentially. This process would then repeat itself.

Other approaches

- This was tried using absorbing masks. This was not of much use, plus total intensity was not conserved.
- Filamentation was not observed at high powers.

Cross-check with literature

In [**Kelley(1965)**], the author assumes the paraxial approximation and assumes no diffraction to get the self-focusing point $z_{sf} = \frac{1}{2}a\sqrt{\frac{n_0}{n_2 {E'_m}^2}}$ where E'_m is the peak field value, a is the characteristic transverse beam waist (as stated in the correction; original paper states it as the radius of curvature).

The paper states that for a 1MW Ruby laser with 2mm beam diameter and for CS_2 with $n'_2 = (0.2 - 1.5) \times 10^{11}$ esu, we should see the self focusing point at 40-100 cm.

To determine parameters to be input into the code:

$\lambda_0 = 694$ nm, n_0 for CS_2 at λ_0 is 1.6, $w_0 = 1$mm,

$n_2 = 3.2 \cdot 10^{-20}\ [m^2/W]$ (Boyd Table 4.1.2), P $= 1$MW

Cross-check with literature

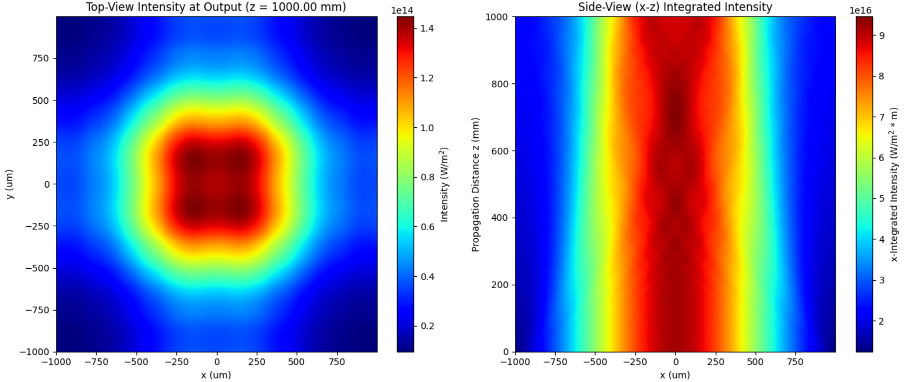


Figure: Output. (1000 steps, $N_x = N_y = 1024$)

Cross-check with literature

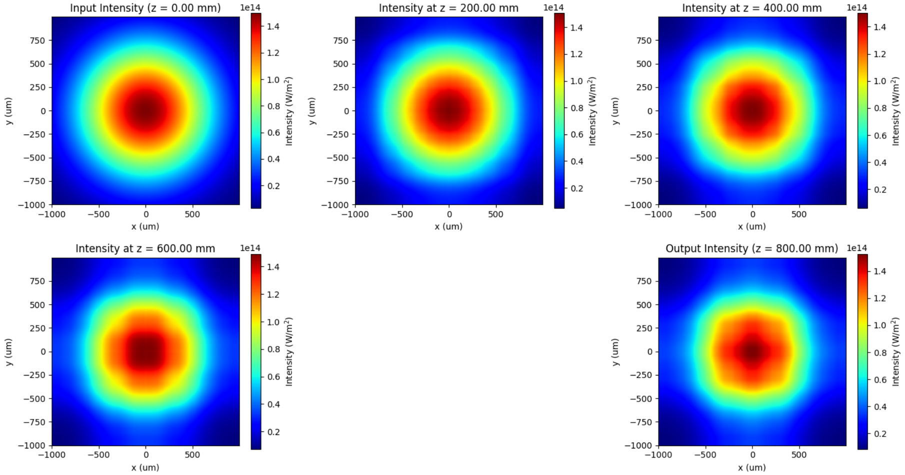


Figure: Output. (1000 steps, $N_x = N_y = 1024$)

Cross-check with literature

We see the self-focusing maxima occur within the range predicted by Kelley.

However, the critical power is given by $\frac{\pi(0.61)^2\lambda_0^2}{8n_0n_2}$, and depending on n_2 taken by Kelley, the 1MW input may fall above or below the threshold, so it is unclear if complete self-focusing is expected or not.

More examples

$\lambda = 1030$ nm, $n_2 = 3 \cdot 10^{-20}$, $w_0 = 50\ \mu m$, $\mathrm{P_{in}} =$ 10 MW, 1000 steps, $N_x = N_y = 1024$, $dx = dy = 2 \cdot 10^{-6}/16$

- $n_0 = 1.1$:

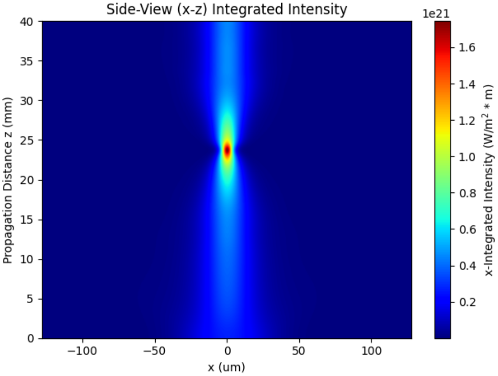
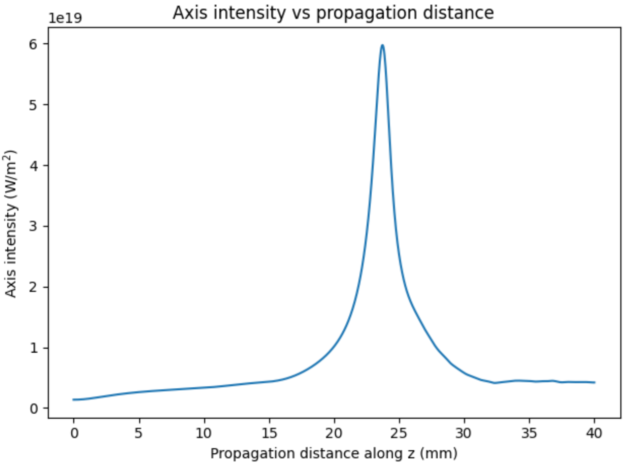


Figure: Output beam waist: $22.5\ \mu m$

More examples

- $n_0 = 1.1$:



More examples

- $n_0 = 1$:

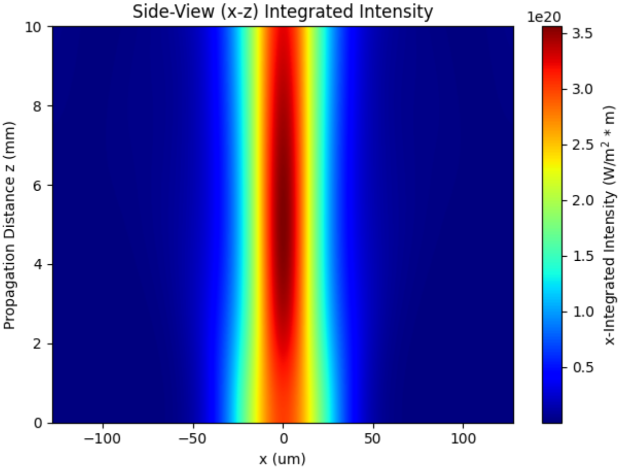
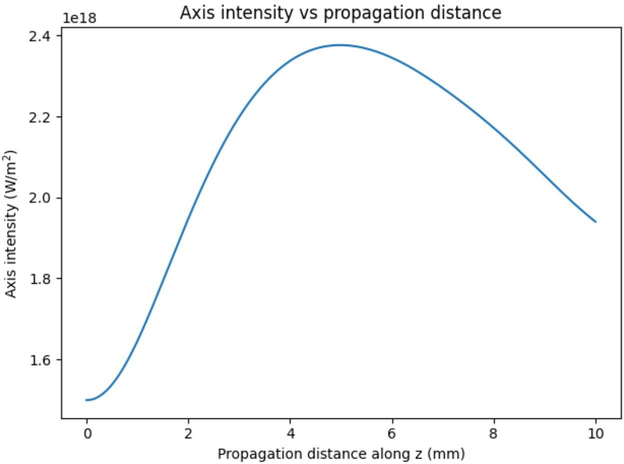


Figure: Output beam waist: $34.5\ \mu m$

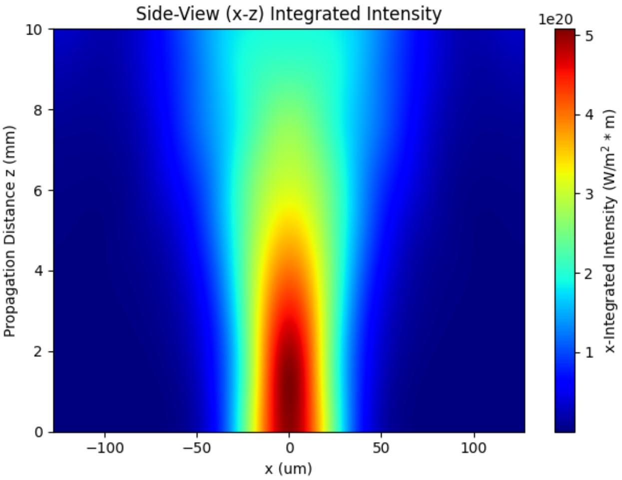
More examples

- $n_0 = 1$:



More examples

- $n_0 = 0.6$ (unphysical):



Limitations

- Square artifacts: We see square artifacts occur whenever the beam waist size is appreciable wrt. the grid size. This is a natural byproduct of low resolution coupled with the periodic boundary conditions applied by **np.fft.fftfreq**. This causes a "wrapping around" effect.
- Low resolution: The system has a limit of 12GB, thus constraining the transverse resolution and the step size for long propagation distances.
- Long compile times: Each output takes 2-10 minutes to compile.
- Constrained to use 2^n: The data format used is **float64** and behaves badly with fractions like $1/3$ in the parameters such as for dx, dy, limiting the flexibility in fine resolving.
- Instability in the beginning: Total intensity of the system seems to rise rapidly in the first 4-5 steps before becoming nearly constant. Although it did not seem to affect much here, it may become problematic.

Further developments

- Going beyond the paraxial approximation.
- Multi frequency driving beam.
- Different beam profiles.
- Anisotropic media.

References

- Nonlinear Optics - Boyd
- [**Kelley(1965)**]
- **[Butcher and Cotter(1990)]**
- **[Marburger(1975)]**
- **[Self-focusing: Past and Present(2009)]**
- **[Fibich(1996)]**
- **[CREOL College Lecture notes]**