Z-Scan Simulation using Gaussian Beam Propagation

Overview

This notebook presents a numerical simulation of the Z-scan technique, based on Gaussian beam propagation through a Kerr nonlinear medium and implemented using ray-transfer (ABCD) matrix analysis.

1. Physical constants and parameters

This section defines all physical, optical, and numerical parameters used in the Z-scan simulation.

In[ ]:=
```
ClearAll["Global`*"];
```
borra todo

In[ ]:=
```
(*=======Beam and laser parameters=======*)
λ = 800 * 10^-9 ;      (*Wavelength[m]*)
ω0 = 0.4 * 10^-3;     (*Initial beam waist[m]*)
p = 100 * 10^3;       (*Laser power[W]*)


(*=======Optical system parameters=======*)
f = 0.13 ;             (*Lens focal length[m]*)
d = 1 * 10^-3;         (*Kerr medium thickness[m]*)
n0 = 1.76;             (*Linear refractive index*)
n2 = 3 * 10^-20; (*3*10^-3;             Nonlinear refractive index[m^2/W]*)
L1 = 0.001;            (*Distance between laser and lens ~ 0 *)


(*=======Scan parameters=======*)
          explora
z0 = -0.05;            (*Initial z position[m]*)
zf = 0.05;             (*Final z position[m]*)
nSteps = 200;          (*Number of scan points*)
                         número


(*=======Detector parameters=======*)
Rdet = 0.5 * ω0;       (*Aperture radius[m]*)


(*=======Derived quantities=======*)
I0[w_] := 2 p / (π w^2); (*Peak intensity*)
q0 = -I π ω0^2 n0 / λ    (*Initial complex beam parameter*)
      número i
```

Out[ ]= 0. − 1.10584 i

2. ABCD matrices

In[ ]:=
```
FreeSpace[L_] := {{1, L}, {0, 1}};

ThinLens[f_] := {{1, 0}, {-1 / f, 1}};

Interface[n1_, n2_] := {{1, 0}, {0, n1 / n2}};

KerrLens[z_, w_] := Module[{hk2}, hk2 = (8 n2 p) / (n0 π w^4);
                      módulo
   {{1, d}, {-hk2 d, 1}}];
```

3. Gaussian beam propagation utilities

```
In[•]:= qPropagate[q_, M_] := (M[[1, 1]] q + M[[1, 2]]) / (M[[2, 1]] q + M[[2, 2]]);

        BeamWaist[q_] := Sqrt[λ / (π n0 * Im[1 / q])];
                         raíz cuadrada   parte imaginaria
```

4. Power through aperture

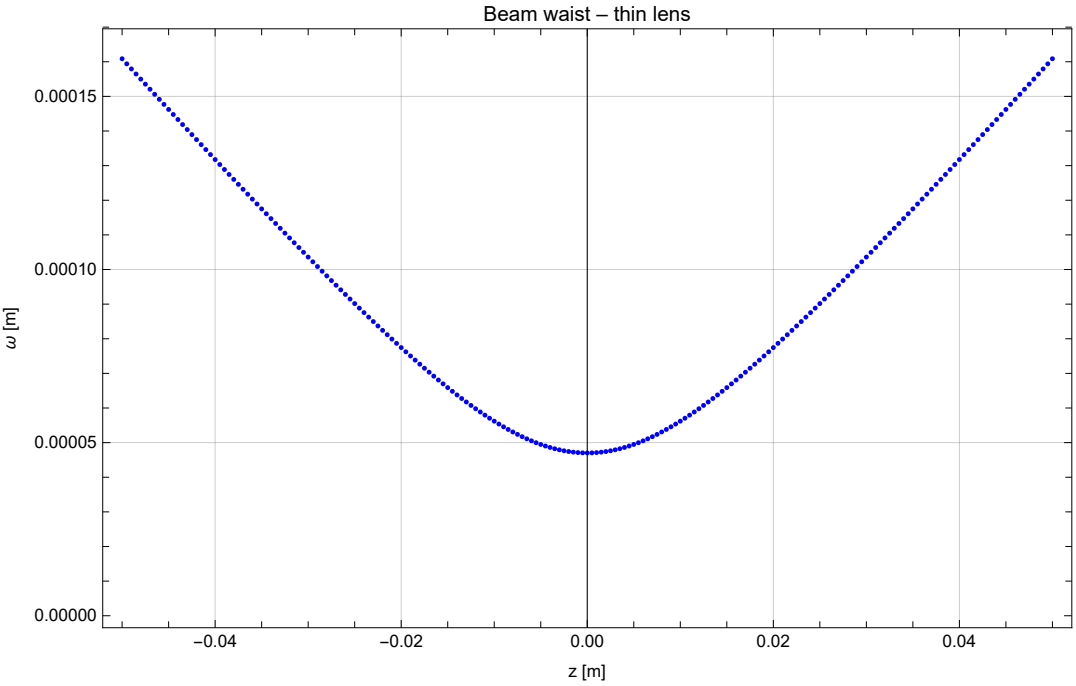
```
In[•]:= DetectedPower[w_] := p (1 - Exp[-2 Rdet^2 / w^2]);
                                 exponencial
```

5. Test case: single thin lens

```
In[•]:= LensOnly[z_] := Module[{M, qf}, M = FreeSpace[f + z].ThinLens[f].FreeSpace[f];
                        módulo
          qf = qPropagate[q0, M];
          BeamWaist[qf]];


In[•]:= zVals = Subdivide[z0, zf, nSteps];
              subdivide

        waistLens = LensOnly /@ zVals;
```

In[•]:=
```
ListPlot[Transpose[{zVals, waistLens}],
 represent··· transposición
 FrameLabel → {"z [m]", "ω [m]"}, PlotStyle → {PointSize[0.005], Blue},
 etiqueta de marco  estilo de repre··· tamaño de punto  azul
 PlotLabel → "Beam waist – thin lens", GridLines → Automatic, Frame → True]
 etiqueta de representación  parrilla de lín··· automático  marco  verdade
```

Out[•]=



6. Z-scan simulation with Kerr medium

In[•]:=
```
ZScan[z_] := Module[{Mpre, qIn, wIn, hk2, Mkerr, Mtot, qDet, wDet},
              módulo
  (*Propagation up to the Kerr medium*)
  Mpre = Interface[1, n0].FreeSpace[f + z].ThinLens[f].FreeSpace[L1];
  qIn = qPropagate[q0, Mpre];
  wIn = BeamWaist[qIn];
  hk2 = (n2 * 8 * p) / (n0 * π * wIn^4);  (*Kerr parameter evaluated INSIDE the medium*)
  Mkerr = {{1, d}, {-d * hk2, 1}};

  (*Full propagation to detector*)
    completo
  Mtot = FreeSpace[f - z].Interface[n0, 1].Mkerr.Mpre;
  qDet = qPropagate[q0, Mtot];
  wDet = BeamWaist[qDet];
  {wDet, DetectedPower[wDet]}

 ];
```

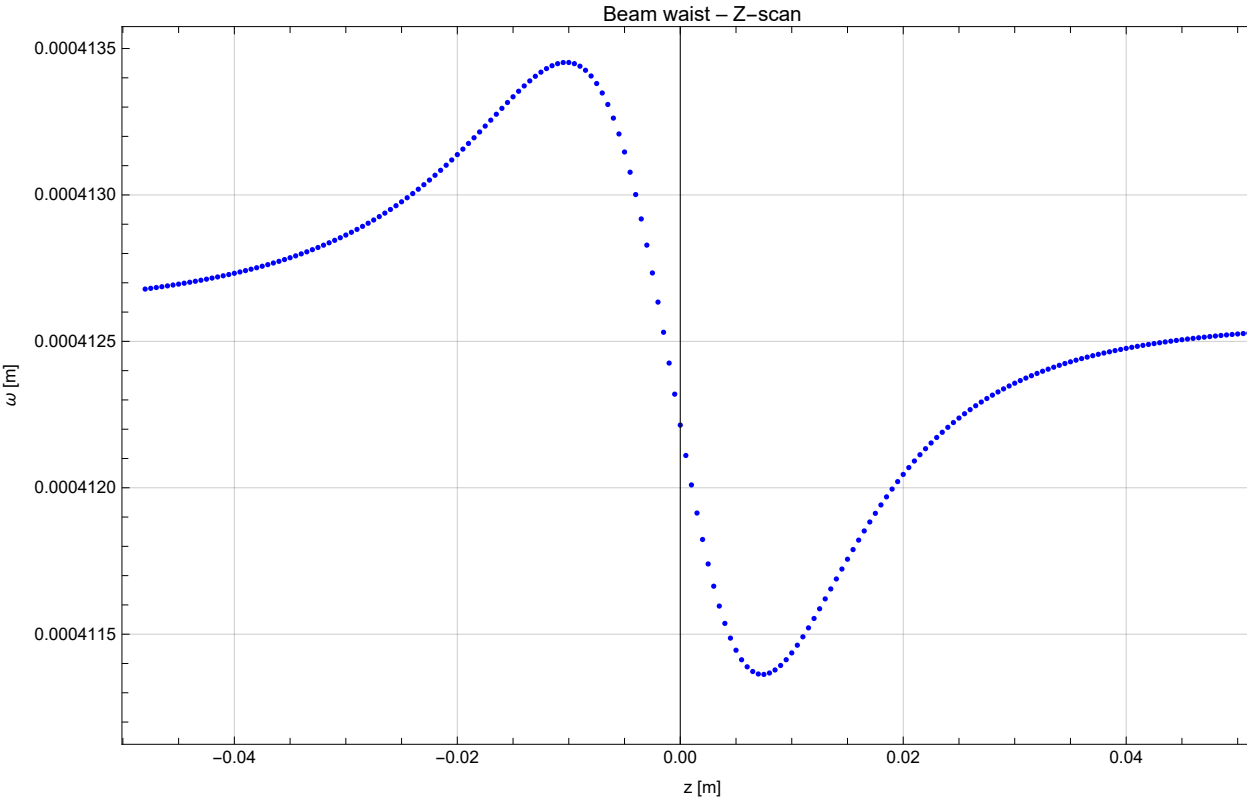
```
In[ ]:= zVals = Subdivide[z0, zf, nSteps];
        subdivide
        zData = ZScan /@ zVals;

        waistZ = zData[[All, 1]];
                 todo
        powerZ = zData[[All, 2]];
                 todo
```

7. Plots

```
In[ ]:= waistZ;
```

```
In[ ]:=
        ListPlot[Transpose[{zVals + 2 * d, waistZ}],
        represent··· transposición
          FrameLabel → {"z [m]", "ω [m]"}, PlotStyle → {PointSize[0.0045], Blue},
          etiqueta de marco                estilo de repre··· tamaño de punto  azul
          GridLines → Automatic, Frame → True, PlotLabel → "Beam waist – Z-scan"]
          parrilla de lín··· automático  marco  verd··· etiqueta de representación
```

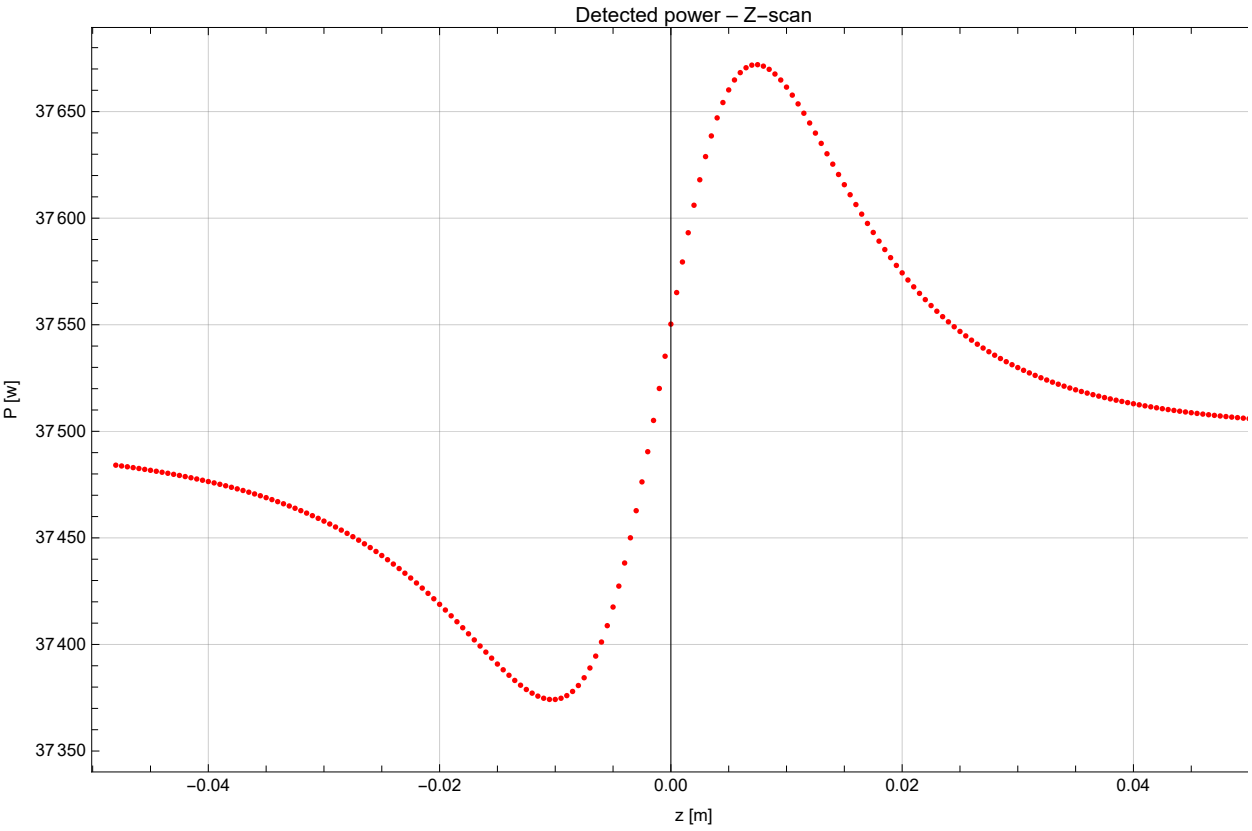
Out[ ]=



In[ ]:=

```
ListPlot[Transpose[{zVals + 2 d, powerZ}],
 FrameLabel → {"z [m]", "P [w]"}, PlotStyle → {PointSize[0.0045], Red},
 GridLines → Automatic, Frame → True, PlotLabel → "Detected power – Z-scan"]
```

Out[ ]=



In[8]:=

```
Export["C:\\Users\\the_w\\OneDrive\\Documentos\\github_portfolio\\zscan.pdf",
 EvaluationNotebook[]]
```

Out[7]= C:\Users\the_w\OneDrive\Documentos\github_portfolio\zscan.pdf