Wakefield Acceleration PIC Simulations

$$\vec{F}(\vec{r}(t), t) = q(\vec{E}(\vec{r}(t), t) + \vec{v} \times \vec{B}(\vec{r}(t), t))$$

```
md""" # Wakefield Acceleration PIC Simulations

```math
\vec{F}(\vec{r}(t), t) = q(\vec{E}(\vec{r}(t), t) + \vec{v} \times \vec{B}(\vec{r}(t),
t))
```
"""
```

```
begin
    using Unitful ✓
    using CSV ✓
    using DataFrames ✓
    using Plots ✓
    using LaTeXStrings ✓
    using UnitfulRecipes ✓
    using LsqFit ✓
    using Images ✓
    using PhysicalConstants.CODATA2018 ✓ : c_0, ε_0, m_e, e, m_u
    using FastTransforms ✓
    using Printf ✓

    using SDFResults ✓
    using PICDataStructures ✓
    import CairoMakie ✓
end
```

```
▸ ["GGiubega_VHEE_Results in Physics-6.pdf", "Gas density profile.PNG", "Laser Spectrum.csv
```

```
readdir()
```

First Simulation Parameters

```
md""" ## First Simulation Parameters """
```

```
(35 fs, 35 μm, 1.25, 17.5 μm)
```

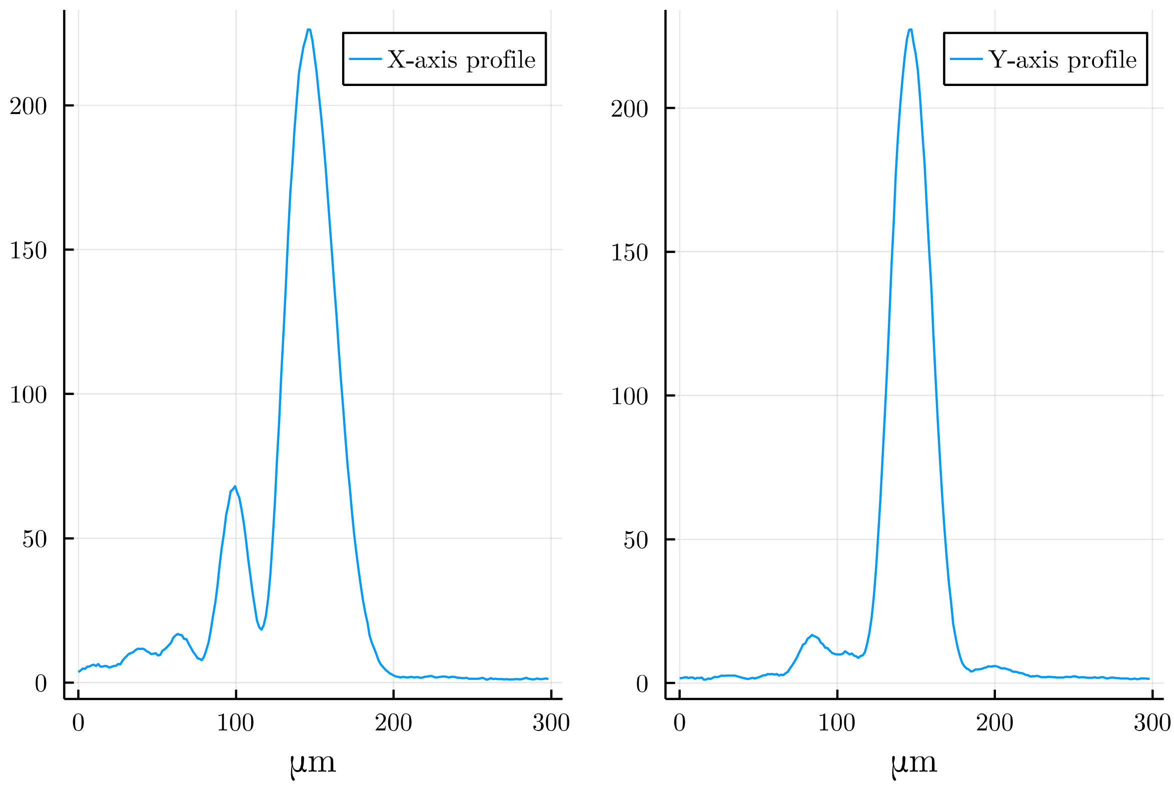
```
begin
    τ = 35u"fs"
    d₀ = 35u"μm"
    a₀ = 1.25
    w₀ = d₀/2
    (τ, d₀, a₀, w₀)
end
```

Reading Laser Intensity Profile Data

Reading X and Y-axis Intensity data and fitting with gaussian functions

	X	Y
1	0.0 μm	3.6
2	1.4 μm	4.202
3	2.8 μm	4.899
4	4.2 μm	4.706
5	5.6 μm	5.5
6	7.0 μm	5.506
7	8.4 μm	6.003
8	9.8 μm	6.192
9	11.2 μm	5.713
10	12.6 μm	6.381
⋮ more		
214	298.2 μm	1.25

	X	Y
1	0.0 μm	1.7
2	1.4 μm	1.7
3	2.8 μm	1.9
4	4.2 μm	1.998
5	5.6 μm	1.702
6	7.0 μm	1.9
7	8.4 μm	1.894
8	9.8 μm	1.505
9	11.2 μm	1.798
10	12.6 μm	1.704
⋮ more		
214	298.2 μm	1.5




```
begin
    plot(
        plot(df_x.X, df_x.Y, label = "X-axis profile", dpi = 360, fontfamily =
"computer modern"),
        plot(df_y.X, df_y.Y, label = "Y-axis profile", dpi = 360, fontfamily =
"computer modern");
        # size = (960,480)
    )
end
```

gaussian (generic function with 1 method)

```
gaussian(x, (a,μ,σ)) = @. a * exp(-(x - μ)^2 / (2 * σ^2))
```

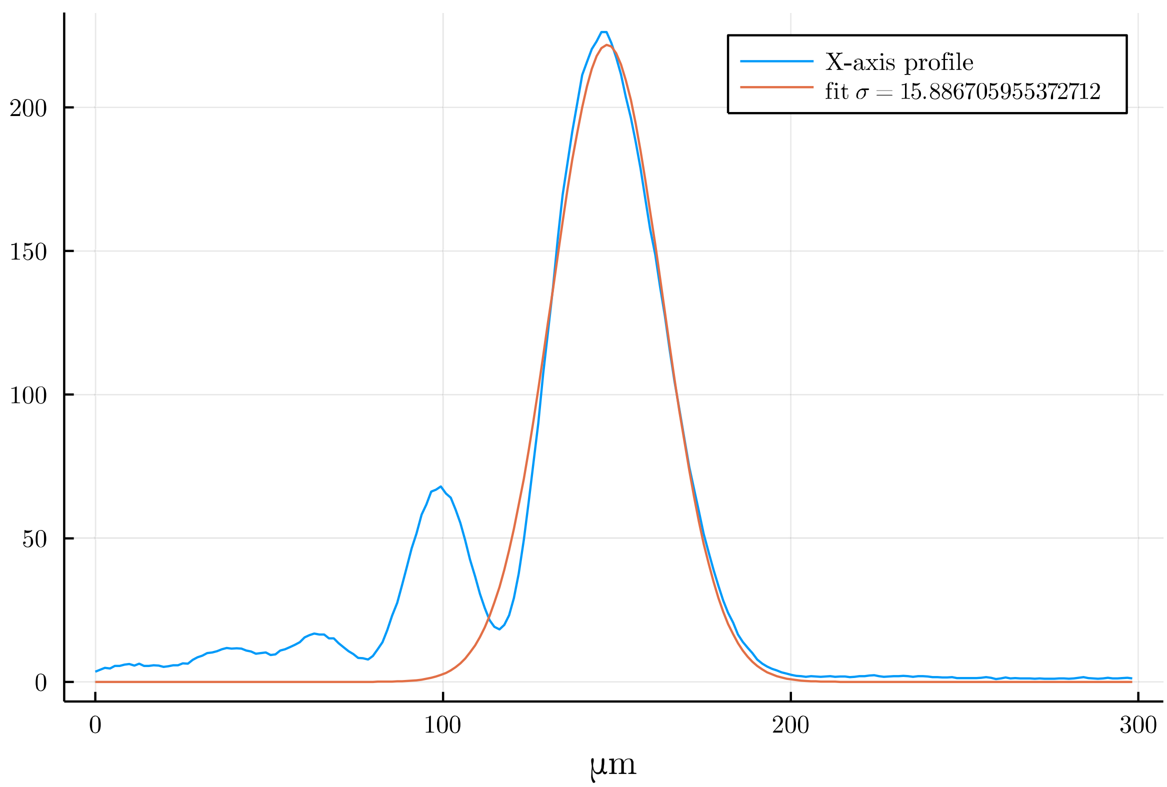
fit_line (generic function with 1 method)

```
function fit_line(x, y)
    p0 = [200, 150, 50.]
    curve_fit(gaussian, ustrip.(x), y, p0)
end
```

fit_x =

▸ LsqFitResult([221.789, 147.222, 15.8867], [-3.6, -4.202, -4.899, -4.706, -5.5, -5.506, -6.

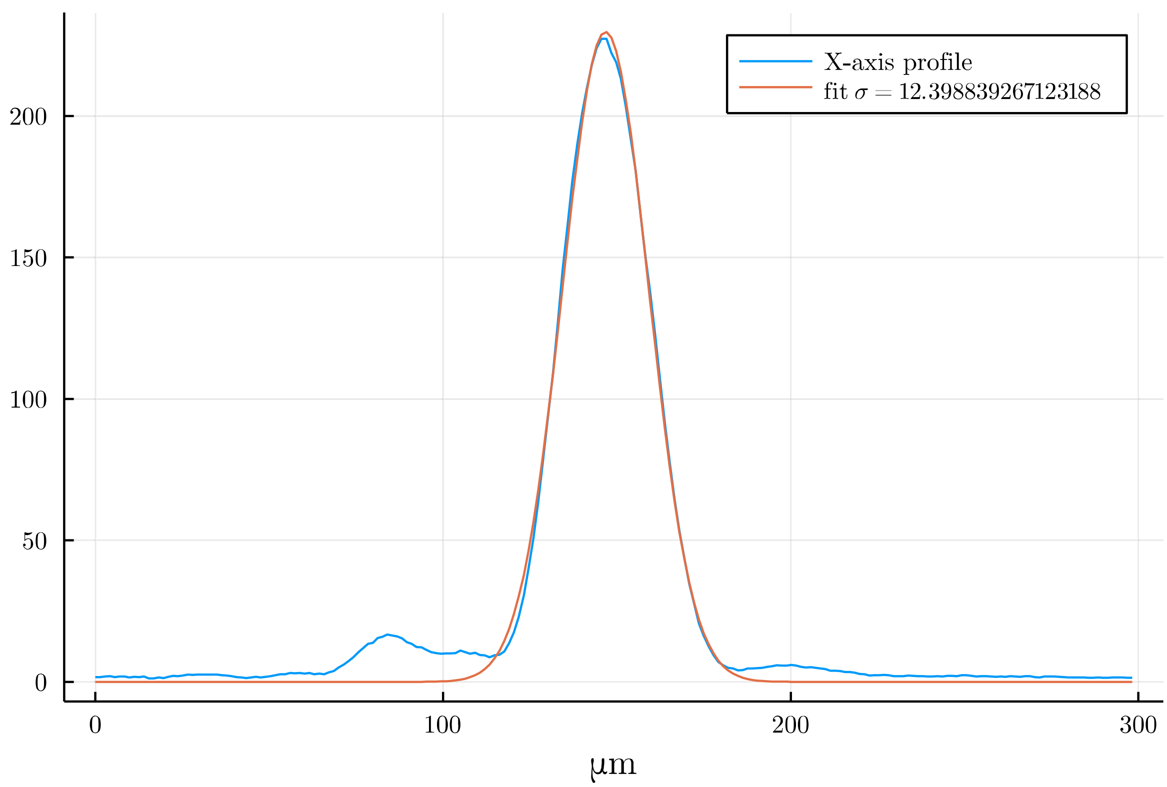
```
fit_x = fit_line(df_x.X, df_x.Y)
```




```
begin
    plot(df_x.X, df_x.Y, label = "X-axis profile", dpi = 360, fontfamily = "computer modern")
    plot!(df_x.X, x->gaussian(ustrip.(x), fit_x.param), label=L"\mathrm{fit}~\sigma = %$(fit_x.param[3])")
end
```

```
fit_y =
▸LsqFitResult([229.722, 146.766, 12.3988], [-1.7, -1.7, -1.9, -1.998, -1.702, -1.9, -1.894,
```

```
fit_y = fit_line(df_y.X, df_y.Y)
```




```
begin
    plot(df_y.X, df_y.Y, label = "X-axis profile", dpi = 360, fontfamily = "computer
modern")
    plot!(df_y.X, x->gaussian(ustrip.(x), fit_y.param), label=L"\mathrm{fit}~\sigma =
%$(fit_y.param[3])")
end
```

▸ (31.7734 μm, 24.7977 μm)

```
begin
    σx = fit_x.param[3] * u"μm"
    σy = fit_y.param[3] * u"μm"
    FWHMx = 2*√(2log(2))*σx
    FWHMy = 2*√(2log(2))*σy
    wx = 2*σx
    wy = 2*σy

    wx, wy
end
```

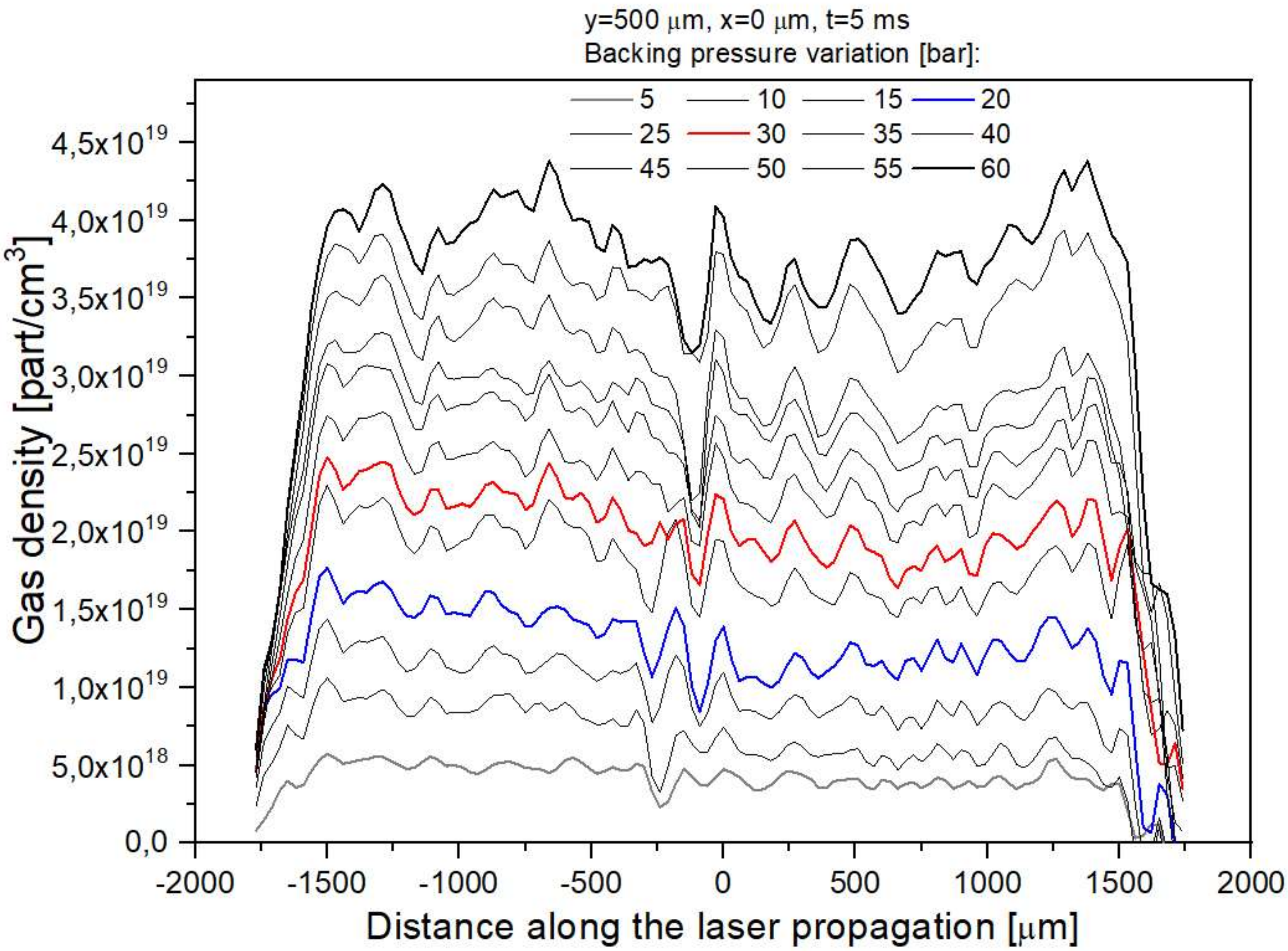
Gas Density Information

- extracting gas cone diameter
- measuring error if assuming periodic boundaries through gas in the laser transverse direction

```
md""" ## Gas Density Information

* extracting gas cone diameter
* measuring error if assuming periodic boundaries through gas in the laser transverse direction
"""
```

img =

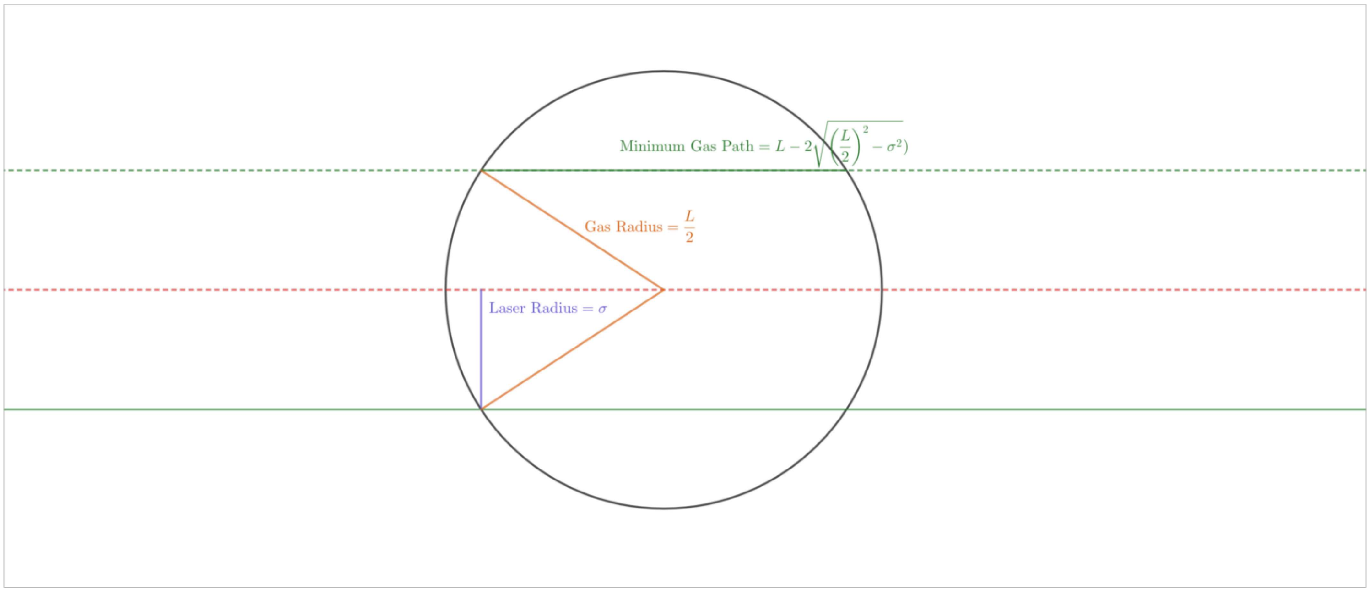



```julia
img = load("Gas density profile.PNG")
```

L = 3000 μm

```julia
L = 3000u"μm"
```

img2 =




```
img2 = load("gas laser intersection.png")
```

gas_err = 0.20417361447744042 μm

```
gas_err = 2(L/2 - √((L/2)^2 - w₀^2))
```

gas_rel_err = 0.00013611574298496028

```
gas_rel_err = gas_err/(L/2)
```

Reading Laser Spectrum Data

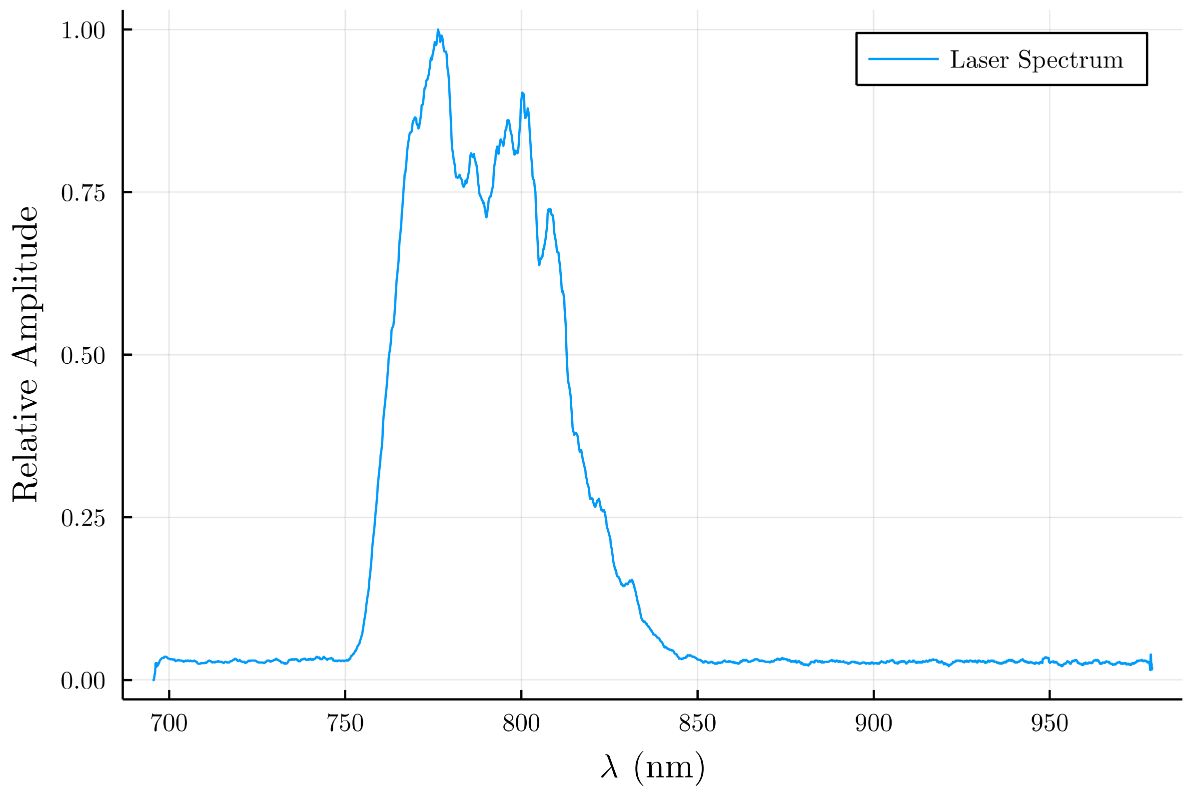
- reading wavelength data of spectrum
- transforming wavelength into frequency
- fitting spectrum with a gaussian

```
md""" ## Reading Laser Spectrum Data

* reading wavelength data of spectrum
* transforming wavelength into frequency
* fitting spectrum with a gaussian
"""
```

	lambda	amp
1	695.323 nm	0.0
2	695.501 nm	0.0
3	695.678 nm	0.0
4	695.856 nm	0.006
5	696.033 nm	0.012
6	696.211 nm	0.026
7	696.388 nm	0.02
8	696.566 nm	0.026
9	696.743 nm	0.022
10	696.92 nm	0.024
⋮ more		
2048	979.102 nm	0.02

```
begin
    fT = CSV.File("Laser Spectrum.csv", delim = "\t")
    df_T = DataFrame(fT, copycols=true)
    df_T.lambda *= u"nm"
    df_T
end
```




```julia
plot(df_T.lambda, df_T.amp, label = "Laser Spectrum", dpi = 360, fontfamily =
"computer modern", xlab = L"\lambda", ylab = "Relative Amplitude")
```

Wavelength to Frequency Spectrum Conversion

The total energy in both expressions of the spectrum must be equal:

$$E = \int^\infty_{0} f_E(\lambda) \mathrm{d}\lambda = \int^\infty_0 \tilde{f}_E(\nu) \mathrm{d} \nu$$

however

$$\lambda\nu = c\longrightarrow\mathrm{d}\lambda = -\frac{c}{\nu^2}\mathrm{d}\nu$$

and because of the limits

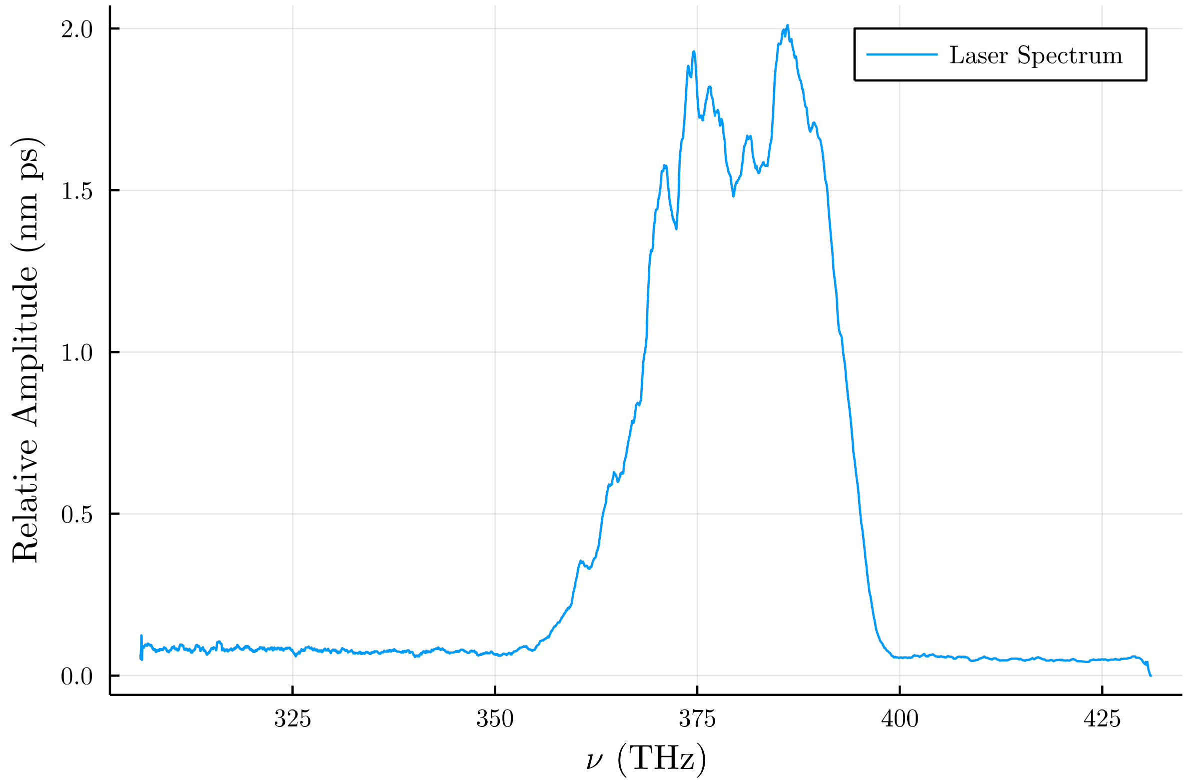
$$\begin{cases} \lambda \rightarrow \infty \longrightarrow \nu \rightarrow 0 \\ \lambda \rightarrow 0 \longrightarrow \nu \rightarrow \infty \end{cases} \longrightarrow \int_0^\infty f_E(\lambda) \mathrm{d} \lambda = \int_\infty^0 f_E\left(\frac{c}{\nu}\right) \left(-\frac{c}{\nu^2}\right)\mathrm{d}\nu = \int_0^\infty f_E\left(\frac{c}{\nu}\right) \left(\frac{c}{\nu^2}\right) \mathrm{d} \nu$$

therefore

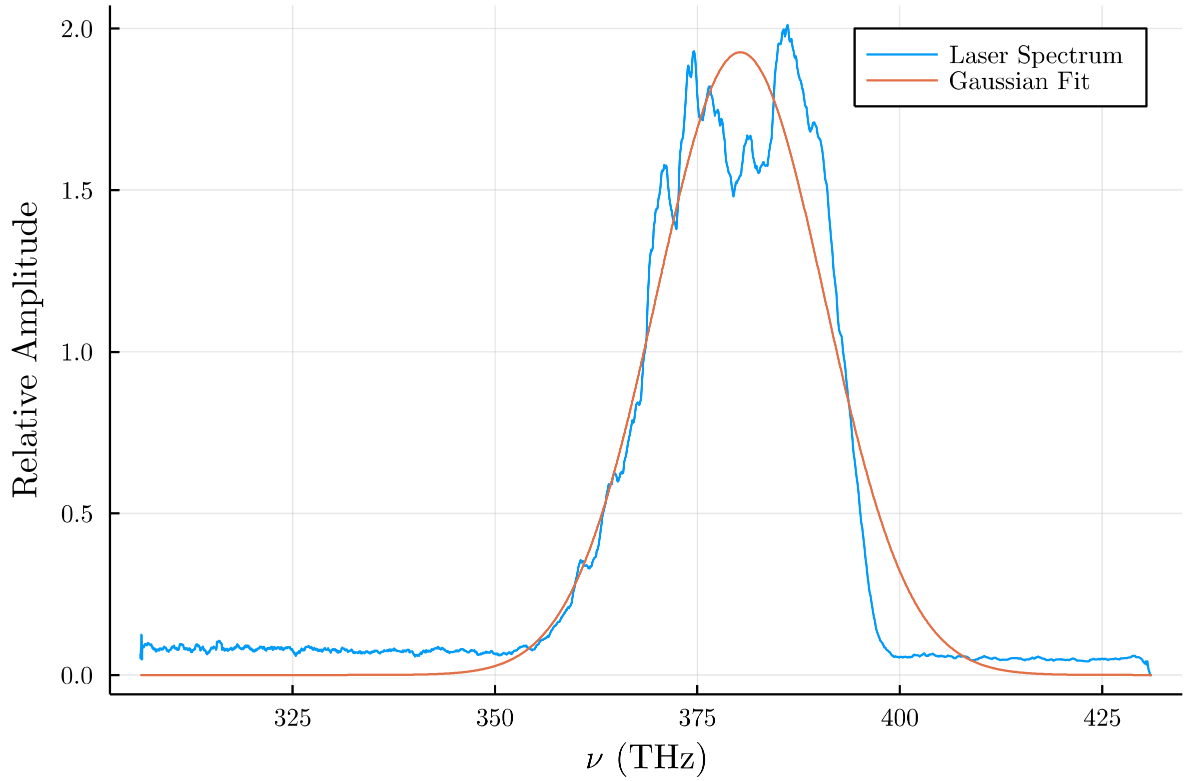
$$\tilde{f}_E(\nu) = f_E\left(\frac{c}{\nu}\right) \left(\frac{c}{\nu^2}\right)$$

```
md""" #### Wavelength to Frequency Spectrum Conversion

The total energy in both expressions of the spectrum must be equal:
```math
E = \int^\infty_{0} f_E(\lambda) \mathrm{d}\lambda = \int^\infty_0 \tilde{f}_E(\nu) \mathrm{d} \nu \\
```
however
```math
\lambda\nu = c\longrightarrow\mathrm{d}\lambda = -\frac{c}{\nu^2}\mathrm{d}\nu
```
and because of the limits
```math
\begin{cases}
\lambda \rightarrow \infty \longrightarrow \nu \rightarrow 0 \\
\lambda \rightarrow 0 \longrightarrow \nu \rightarrow \infty
\end{cases}
\longrightarrow
\int_0^\infty f_E(\lambda) \mathrm{d} \lambda = \int_\infty^0 f_E\left(\frac{c}{\nu}\right) \left(-\frac{c}{\nu^2}\right)\mathrm{d}\nu = \int_0^\infty f_E\left(\frac{c}{\nu}\right) \left(\frac{c}{\nu^2}\right) \mathrm{d} \nu
```
therefore
```math
\tilde{f}_E(\nu) = f_E\left(\frac{c}{\nu}\right) \left(\frac{c}{\nu^2}\right)
```
"""
```




```
begin
    νlist = uconvert.(u"THz", c_0 ./ df_T.lambda)
    νlist² = νlist.^2
    νamplist = uconvert.(u"nm*ps",c_0 * df_T.amp ./ νlist²)
    plot(νlist, νamplist, label = "Laser Spectrum", dpi = 360, fontfamily = "computer
modern", xlab = L"\nu", ylab = "Relative Amplitude")
end
```




```
begin
    p0 = [2, 380, 40.]
    temp_curve_fit = curve_fit(gaussian, ustrip.(νlist), ustrip.(νamplist), p0)
    plot(νlist, [ustrip.(νamplist), x->gaussian(ustrip.(x), temp_curve_fit.param)],
label = ["Laser Spectrum" "Gaussian Fit"], dpi = 360, fontfamily = "computer modern",
xlab = L"\nu", ylab = "Relative Amplitude")
end
```

To do: Supergausiană

```
md"To do: Supergausiană"
```

```
[1.9262, 380.311 THz, 10.4043 THz]
```

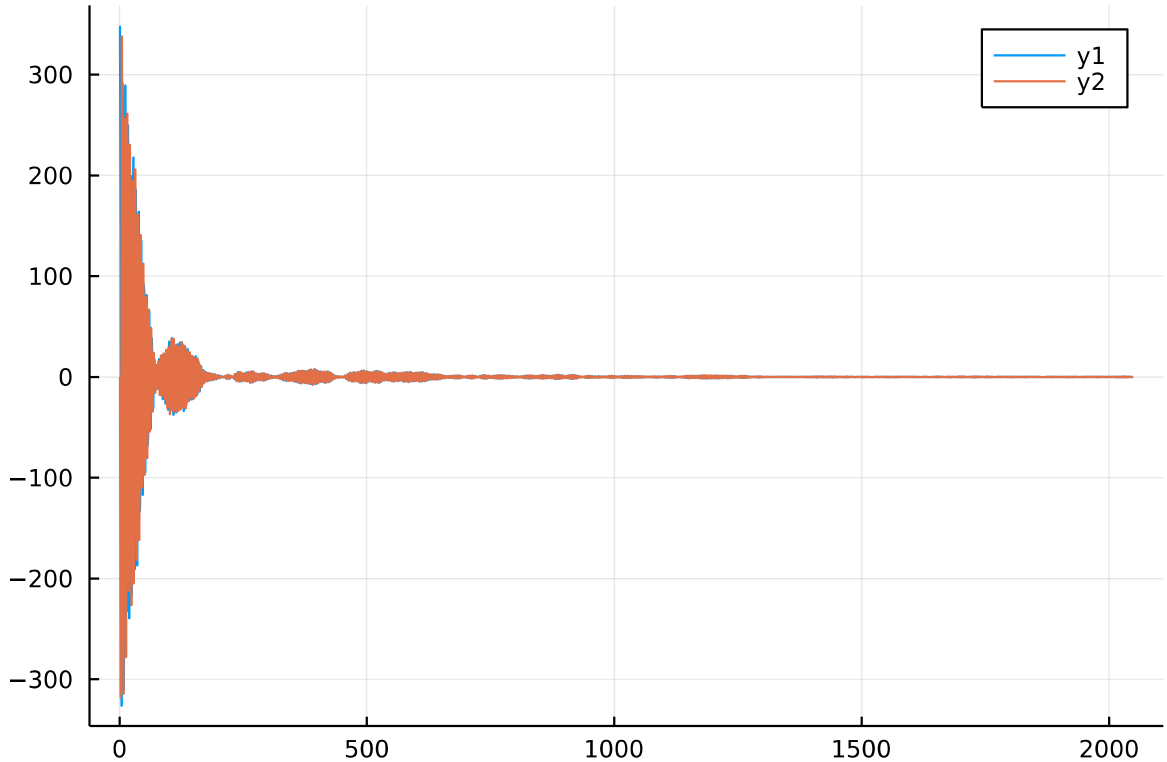
```
temp_amp, νmodel, σmodel = (1,u"THz",u"THz") .* temp_curve_fit.param
```

```
λmodel = 788.2825172427168 nm
```

```
λmodel = uconvert(u"nm",c_0/νmodel)
```

```
temp3 =
[348.27-2.73838e-15im, 141.186-317.751im, -231.991-256.735im, -326.106+107.164im, -33.552
```

```
temp3 = nufft1(complex(df_T.amp), ustrip(uconvert.(u"THz", c_0 ./ df_T.lambda)),eps())
```




```
plot(1:2048, [real(temp3), imag(temp3)])
```

Deck File Set-up

```
res_lin = 0.01522296858995939 nm^-1
res_lin = 12/λmodel
```

Analytic plasma wavelength computation

$$\omega_P = \sqrt{\frac{e^2 n_e}{\varepsilon_0 m_e}}$$

$$\lambda_P \approx \frac{2\pi c}{\omega_P}$$

$$\lambda = \underbrace{2\pi c}_{\mu\mathrm{ms}^{-1}} \underbrace{\sqrt{\frac{\varepsilon_0 m_e}{e^2}}}_{\sqrt{\mathrm{s}^2\mathrm{cm}^{-3}}} \underbrace{\sqrt{\frac{1}{n_e}}}_{\sqrt{\mathrm{cm}^3}}$$

```
λ_conv_const = 3.3389432702154293e10
λ_conv_const = 2π*ustrip(u"μm/s", c_0)*√(ustrip(u"s^2*cm^-3",ε_0 * m_e / e^2))
```

λ_plasma = 6.6778865404308565 μm

```
λ_plasma = uconvert(u"μm", 2π*c_0*√(ε_0*m_e / (n*e^2)))
```

To do: Moving window x-direction 10 lungimi de undă a plasmei

```
md"""To do: Moving window x-direction 10 lungimi de undă a plasmei"""
```

667.7886540430857 μm

```
begin
    wl_mult = 100.
    x_min = 0*u"μm"
    x_max = wl_mult * λ_plasma
    Δx = x_max - x_min
end
```

nx = 10166

```
begin
    nx = uconvert(NoUnits, Δx*res_lin)
    nx = ceil(Int, nx)
    md"""nx = $nx"""
end
```

Choosing a width for simulations centered on the laser

Nota bene: In EPOCH

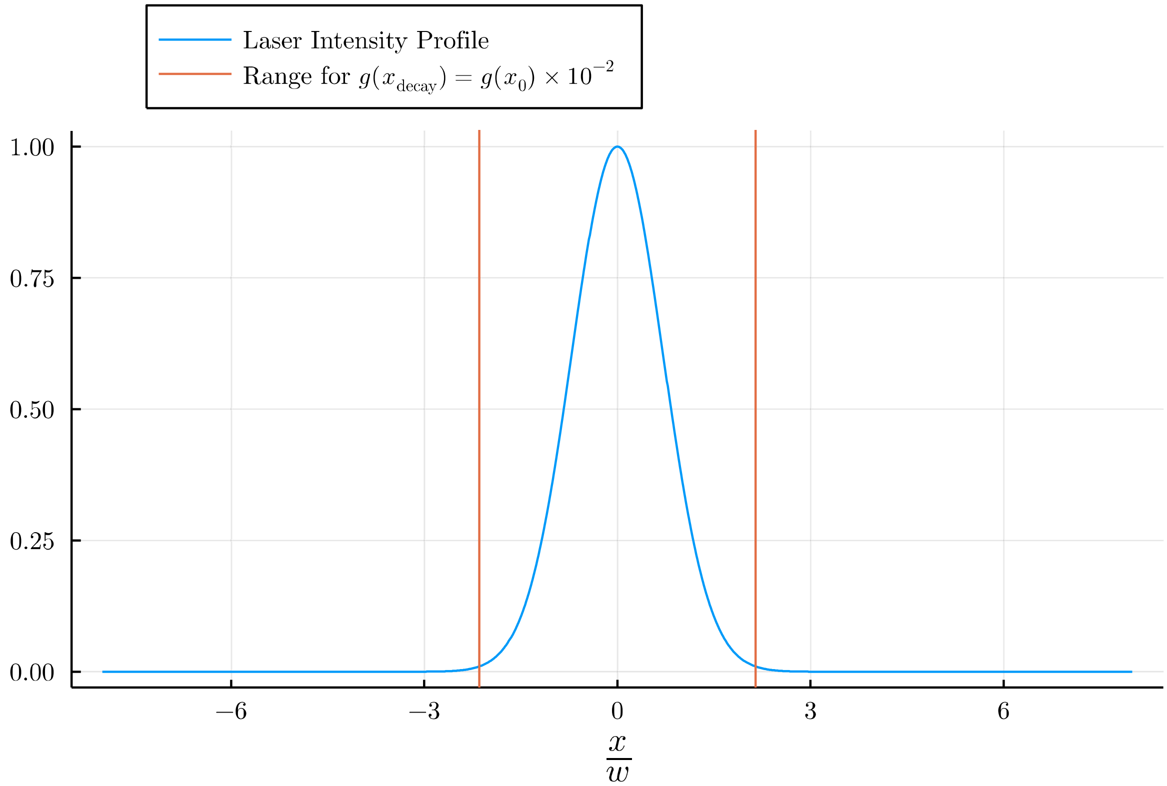
$$\mathrm{gauss}(x, x_0, w) = \exp\left(-\frac{(x-x_0)^2}{w^2}\right) \equiv g(x)$$

then, for a field with $E \propto \mathrm{gauss}(x, x_0, w)$, the intensity is $I \propto E^2 \propto \mathrm{gauss}^2(x, x_0, w)$, such that when $E_{\mathrm{decayed}} = f E_{\mathrm{max}}$ the intensity decays $I_{\mathrm{decayed}} = f^2 I_{\mathrm{max}}$. A field decay of 0.01 becomes an intensity decay of 0.0001.

$$\begin{gathered}
\max(g) = g(x_0) = \exp(0) = 1 \\
g(x_0 + x_{\mathrm{decay}}) = f \longrightarrow \exp\left(-\frac{x_{\mathrm{decay}}^2}{w^2}\right) = f \\
-\frac{x_{\mathrm{decay}}^2}{w^2} = \ln(f) \\
x_{\mathrm{decay}} = w\sqrt{-\ln(f)}
\end{gathered}$$

We want our domain to include space for the field to decay both to the left and to the right:

$$\begin{aligned}
\Delta x &= x_{\mathrm{right}} - x_{\mathrm{left}} \\
&= (x_0 + x_{\mathrm{decay}}) - (x_0 - x_{\mathrm{decay}}) \\
&= 2x_{\mathrm{decay}} \\
&= 2w\sqrt{-\ln(f)}
\end{aligned}$$




```
begin
    w = 1
    g(x) = exp(-x^2/(w^2))
    f = 1/100
    cutoff_x = [-√(-log(f)), √(-log(f))]
    plot(-8:0.01:8, x->g(w*x), label = "Laser Intensity Profile", xlab = L"\frac{x}
{w}")
    vline!(cutoff_x, label = "Range for \$g(x_\\mathrm{decay})= g(x_0) \\times
10^{$(round(Int, log10(f)))}\$", fontfamily = "computer modern", legend = :outertop)
    # (g(0), g(s*cutoff_x[1]), g(s*cutoff_x[1])/g(0), exp(-2))
end
```

75.10881092012715 μm

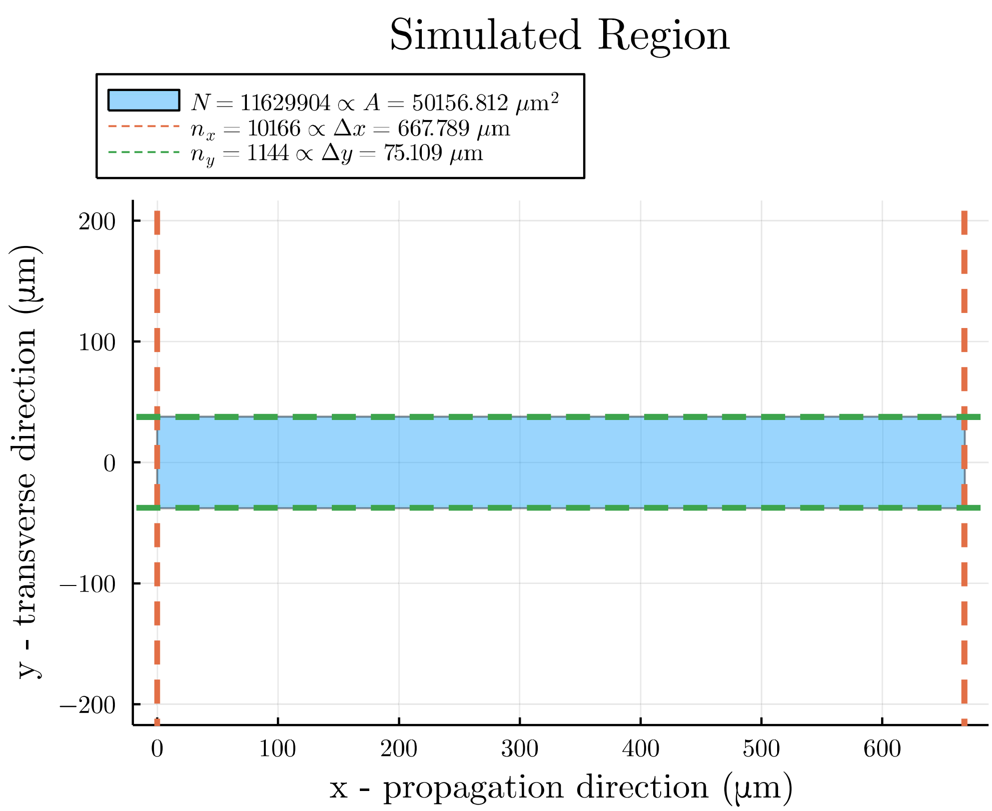
```
begin
    y_min, y_max = (-w₀*√(-log(f)), w₀*√(-log(f)))
    Δy = y_max - y_min
end
```

▶ (y_min = -37.5544 μm, y_max = 37.5544 μm, Δy = 75.1088 μm, ny = 1144)

```
begin
    ny = uconvert(NoUnits, Δy*res_lin)
    ny = ceil(Int, ny)
    (y_min = y_min, y_max = y_max, Δy = Δy, ny = ny)
end
```

Cells used = 11629904 Simulated Area = 50156.81175112833 µm^2

```
begin
    N = nx*ny
    SimulatedArea = Δx*Δy
    md"""
    Cells used = $N
    Simulated Area = $SimulatedArea
    """
end
```



(29.7263 fs, 21.0196 fs, 63.7916 fs, 19.1242 µm)

```
begin
    w_τ = τ/√(2*log(2))
    σ_τ = τ/(2*√log(2))
    invf = 1_00
    t₀ = σ_τ*√(2log(invf))
    (w_τ, σ_τ,t₀,uconvert(u"µm",c_0*t₀))
end
```

t_end = 2.3550862827110864e-12

```
t_end = ustrip(u"s", Δx/c_0) + 2*ustrip(u"s", t₀)
```

```
5.300000000000001e25 m^-3
```

```
begin
    n = uconvert(u"m^-3", 2.5e19u"cm^-3")
    pHelium = 0.99
    pNitrogen = 0.01
    nHe = 0.99*n
    nN  = 2*0.01*n
    nₑ  = 2*nHe + 7*nN
end
```

```
25533.199026445905
```

```
begin
    mHe_e = 4.002602*m_u/m_e
    mN_e = 14.007*m_u/m_e
end
```

Peak Amplitude vs. Reduced Amplitude

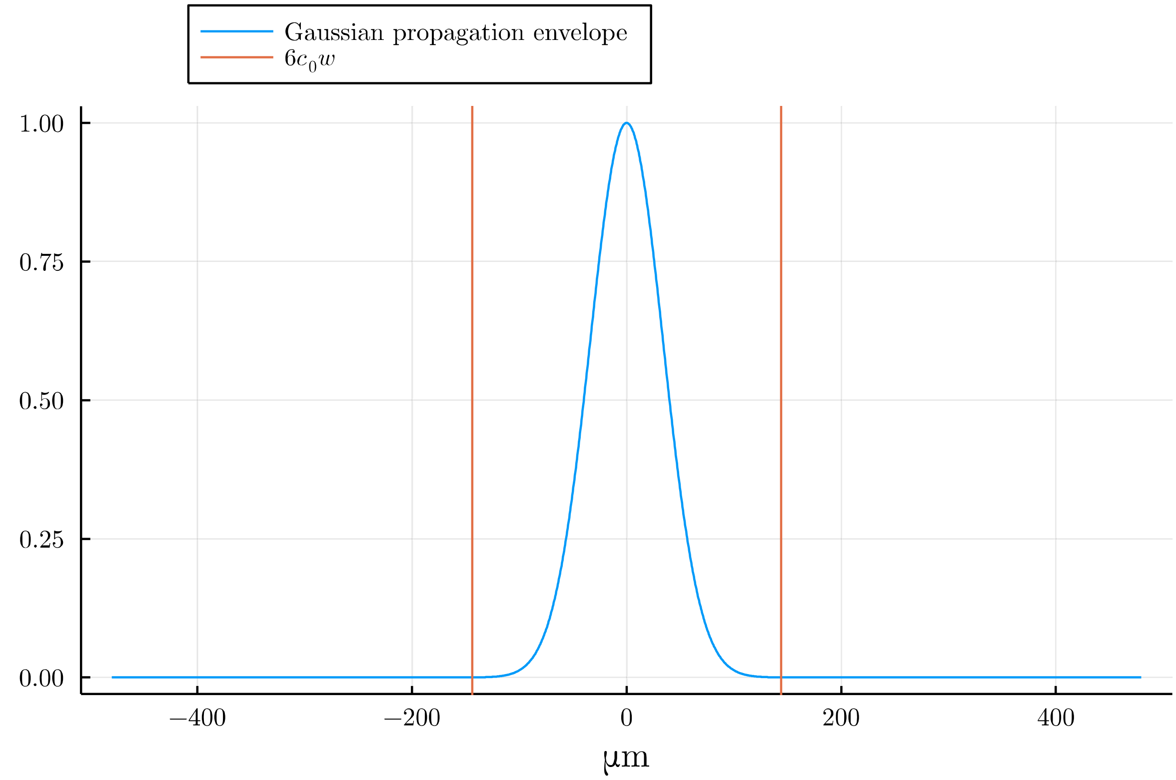
$$E_0 = a_0 m_e c_0 \omega |e|^{-1}$$
$$a_0 = E_0 |e| (m_e c_0 \omega)^{-1}$$

```
0.016537909736513888 PW
```

```
begin
    E₀ = uconvert(u"V/m", a₀*m_e*c_0*2*π*νmodel/abs(e))
    I₀ = abs(E₀^2/(2*377*u"Ω"))
    P₀ = uconvert(u"PW", π*d₀^2*I₀/8)
end
```

```
σμ = 9.611417206478485e-14
```

```
σμ = ustrip(u"s", inv(σmodel))
```




```
begin
    timew = 2*inv(σmodel)*√(log(2))
    g(z,t) = exp(-(t-z/c_0)^2/timew^2)
    plot(uconvert.(u"μm", c_0*range(-10*timew, stop = 10*timew, length =2000)), z->g(z,0u"s"), label = "Gaussian propagation envelope")
    vline!([-3c_0*timew, 3c_0*timew],
        label = L"6 c_0 w",
        legend = :outertop,
        dpi = 360,
        fontfamily = "computer modern")
end
```

Number of saves heuristic

```
100.06922855944562
```

```
ustrip(u"fs", 2*L/c_0)/200 # max saves 200
```

Epoch File Prototype & Save Location

```
md""" #### Epoch File Prototype & Save Location"""
```

```
epoch_input =
"begin:control\n\tnx = 10166\n\tny = 1144\n\n    \t# Size of domain\n    \tx_min = 0 * micron
```

```
epoch_input = """
```

```
    y_max = $(ustrip(y_max)) * micron


    # Final time of simulation
    t_end = $t_end

    stdout_frequency = 400
end:control

begin:boundaries
    bc_x_min = simple_laser
    bc_x_max = simple_outflow
    bc_y_min = open
    bc_y_max = open
end:boundaries

begin:constant
    dens   = $(ustrip(nₑ))
    densHe = $(ustrip(nHe))
    densN  = $(ustrip(nN))

    r = y
end:constant

begin:species
  name = electron
  charge = -1.0
  mass = 1.0
  temp = 0
  density = dens
  npart_per_cell = 9
end:species

begin:species
  name = helium
  charge = 2.0
  mass = $mHe_e
  temp = 0
  density = densHe
  npart_per_cell = 1
end:species
```

```
begin:species
  name = nitrogen
  charge = 7.0
  mass = $mN_e
  temp = 0
  density = densN
  npart_per_cell = 1
end:species

begin:laser
  boundary = x_min
  amp = $(ustrip(E₀))
  lambda = $(ustrip(u"m", λmodel))
  profile = gauss(r, 0, $(ustrip(u"m", w₀)))
  t_profile = gauss(time, $(ustrip(u"s", t₀)), $(ustrip(u"s", w_τ)))
end:laser

begin:output
  #timesteps between output dumps
  dt_snapshot = t_end/400

  # Properties on grid
  grid = always
  ex = always
  ey = always
  ez = always
  bx = always
  by = always
  bz = always

  # Properties at particle positions
  particle_grid = always
  px = always
  py = always

  number_density = always + species
  average_particle_energy = always + species
end:output

"""
```

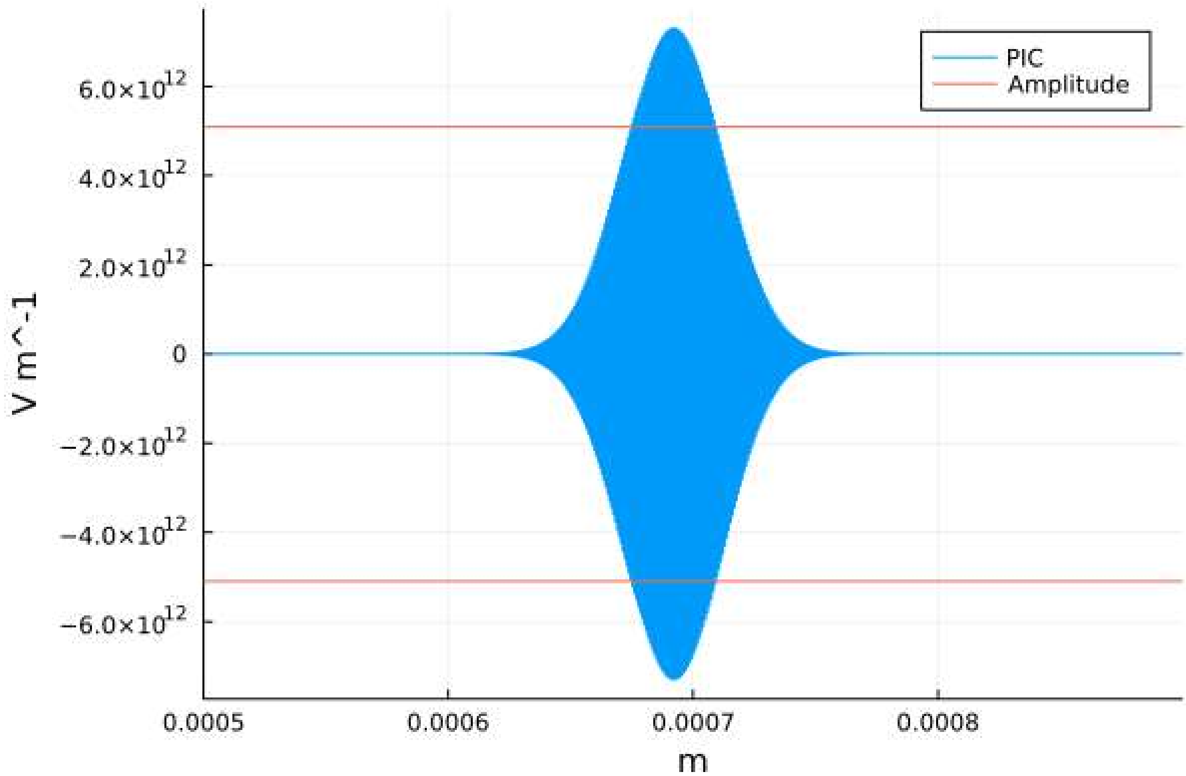
```
# begin:window
#   move_window = T
#   window_v_x = c
#   #when pulse is centered in window
#   window_start_time = (x_max - x_min)/c
#   bc_x_min_after_move = simple_outflow
#   bc_x_max_after_move = simple_outflow
# end:window
```

1535

```
write("input.deck", epoch_input)
```

Post Simulations Checks & Processing

```
md"""## Post Simulations Checks & Processing """
```



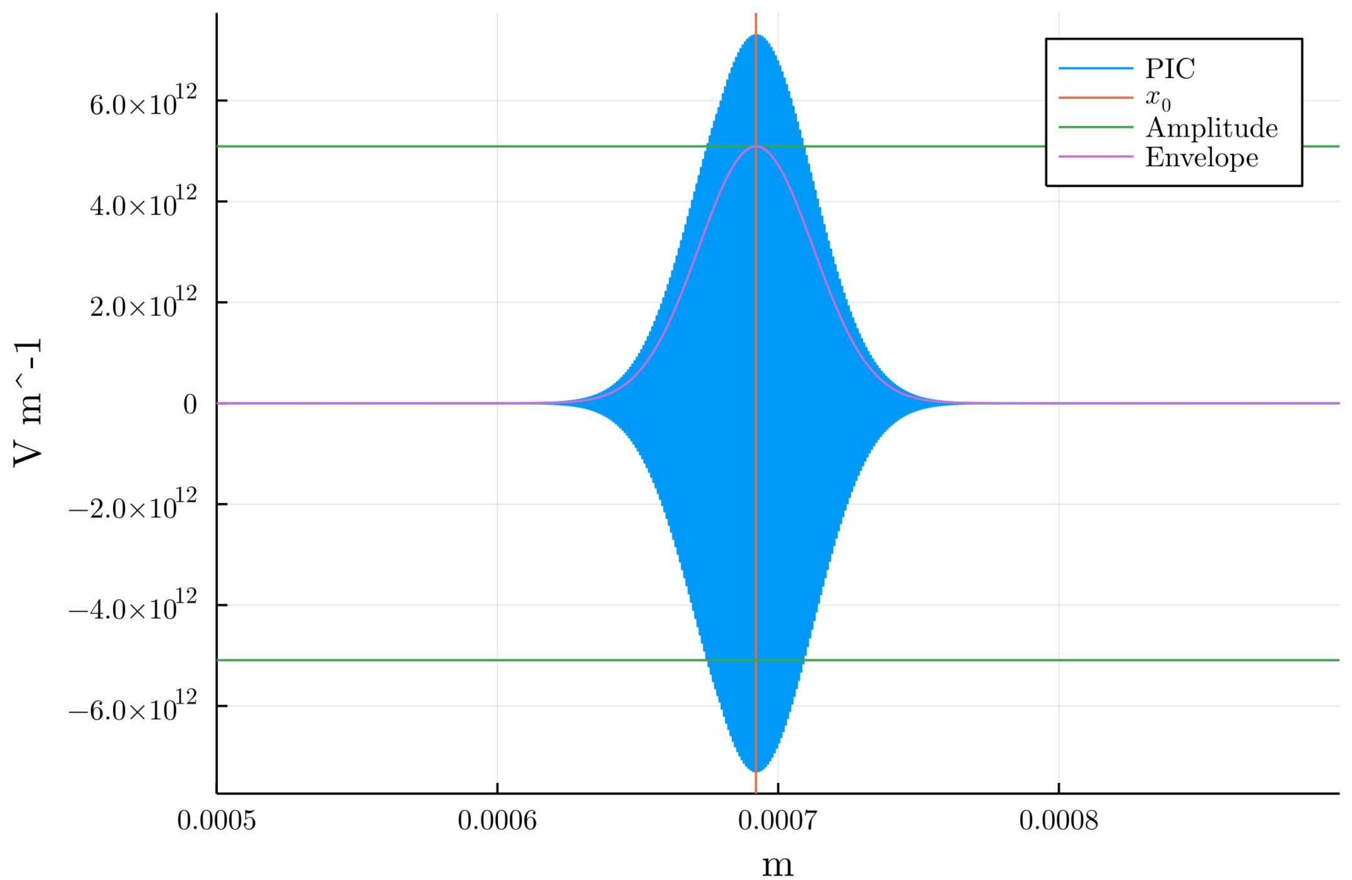

```
begin
    sim = read_simulation("/mnt/storage/epoch/PW_wakefield/1D/test_laser")
    file = sim[15]
    Ey = file[:ey]
    plot(collect(getdomain(Ey)), Ey, xlims = (0.0005,0.0009),label = "PIC")
    hline!([-E₀, E₀], label = ["Amplitude" "Amplitude"])
end
```

▸ (1.43384, 7.30011e12 V m^-1)

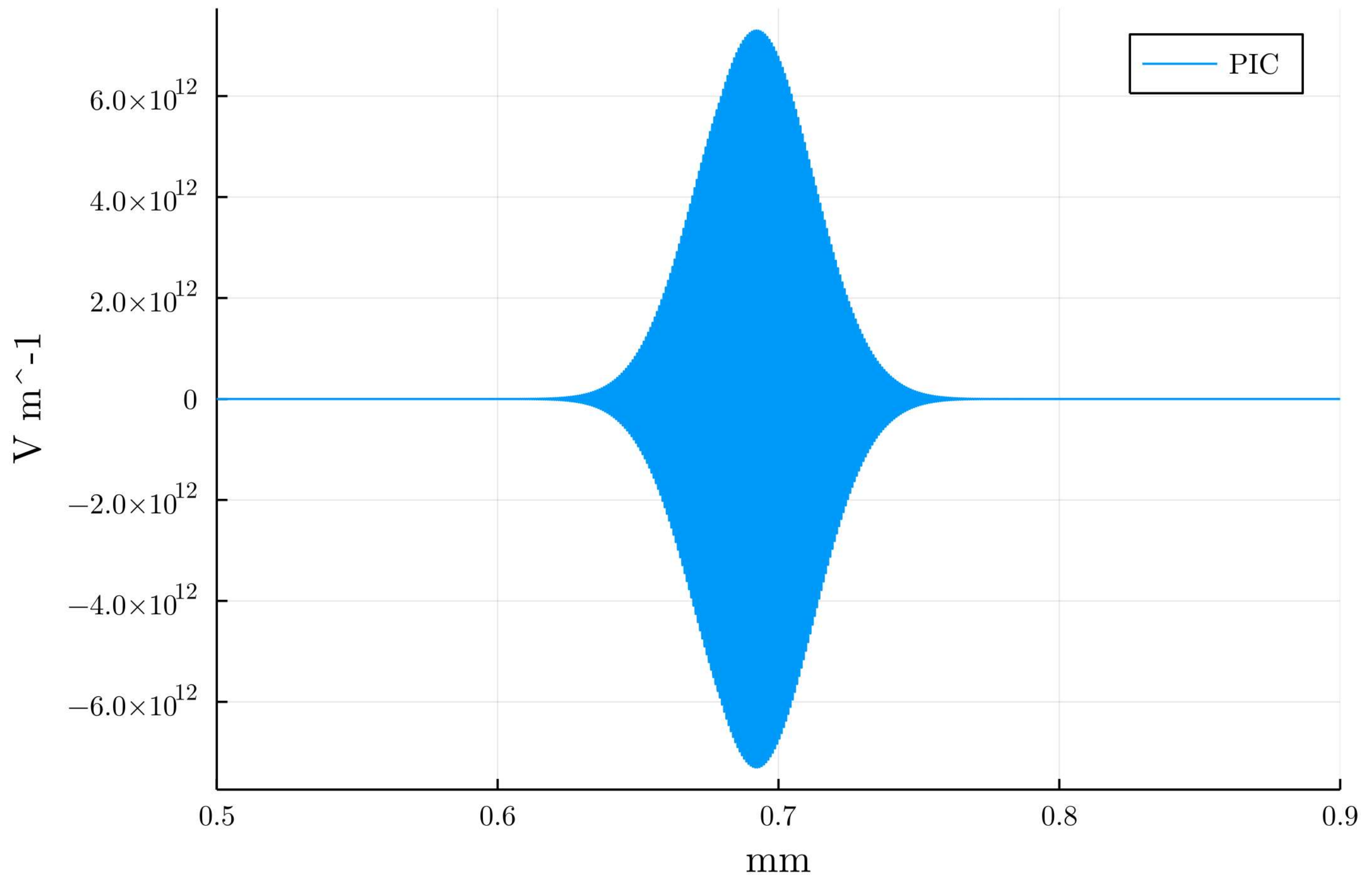
```
begin
    Emax, idx = findmax(abs.(Ey))
    Emax/E₀, Emax
end
```

0.000692088862430999 m

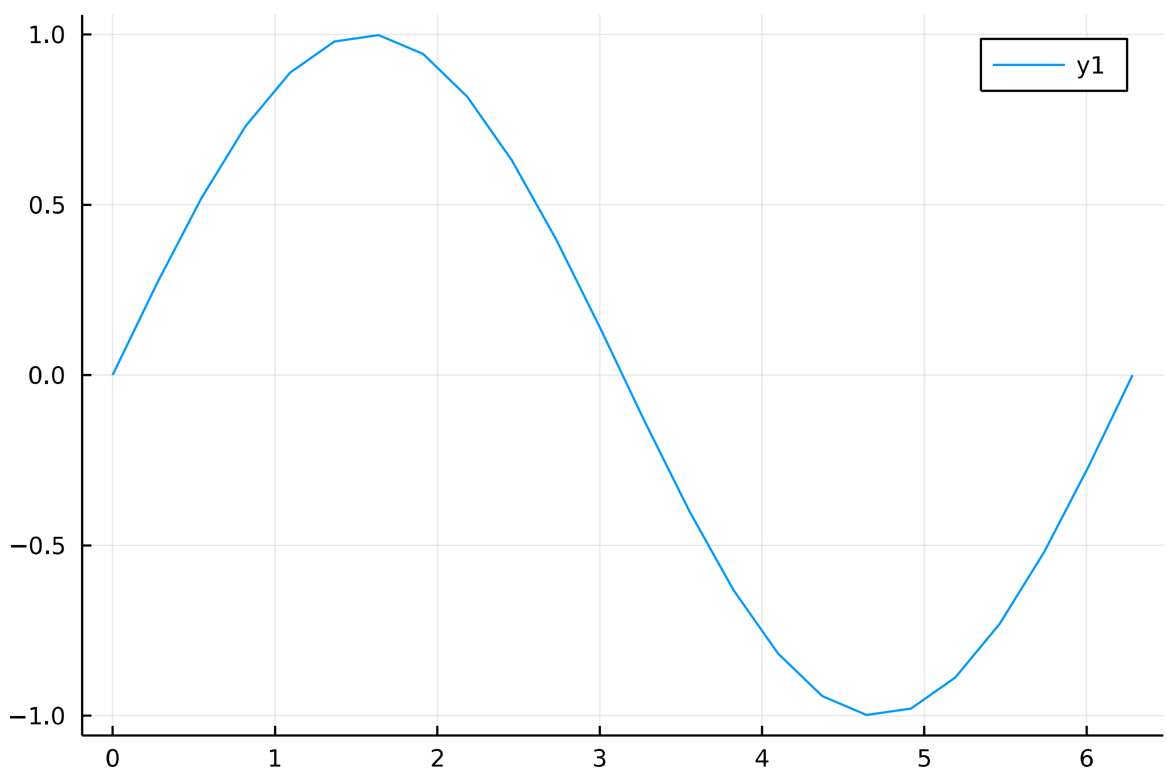
```
begin
    domain = getdomain(Ey).x
    x₀ = domain[idx]
end
```




```
begin
    plot(domain, Ey, xlims = (0.0005,0.0009),label = "PIC", dpi = 360, fontfamily =
"computer modern")
    vline!([domain[idx]], label = L"x_0")
    hline!([-E₀, E₀], label = ["Amplitude" "Amplitude"])
    plot!(domain, x->E₀*exp(-(x-x₀)^2/(c_0^2*inv(σmodel)^2)), label = "Envelope")
end
```




```
plot(uconvert.(u"mm",domain), Ey, xlims = (0.5,0.9),label = "PIC", dpi = 360,
fontfamily = "computer modern")
```




```
plot(range(0, 2π, length = 24), sin)
```

```
"/mnt/storage/vlad/Scripts&such/PIC Sims"
```

```julia
pwd()
```