FibDisp

Simulation of Ultrashort-Pulse Propagation in Gas-Filled Hollow-Core Fibers
Theory, Numerical Model, Installation, and User Guide

Davide Faccialà

Contents

1 Purpose and Scope

`FibDisp` simulates the propagation of an ultrashort laser pulse in a single-mode gas-filled hollow-core fiber or capillary. The propagation is calculated with a symmetric split-step Fourier method applied to a generalized nonlinear Schrödinger equation containing:

- second-order dispersion, described by β_2;
- third-order dispersion, described by β_3;
- linear attenuation, described by α;
- Kerr self-phase modulation, described by the nonlinear coefficient γ;
- self-steepening, also called the optical-shock correction;
- input quadratic and cubic spectral phase, specified by GDD and TOD.

The program also provides:

- Gaussian, hyperbolic-secant, and super-Gaussian input pulses;
- calculated gas/waveguide propagation coefficients and a Manual mode;
- selectable propagation terms for diagnostic calculations;
- two numerical solvers for the self-steepening nonlinear step;
- temporal and spectral evolution maps along the fiber;
- an optional fixed post-fiber GDD display in the main Results tab;
- spectral-phase polynomial fitting with spectral-power weighting;
- pure-GDD compressor analysis and numerical GDD optimization;
- parameter sweeps over physical parameters and direct propagation coefficients;
- compressed and uncompressed pulse-power maps during parameter sweeps;
- snapshots of the settings used for the current and most recent sweep;
- numerical export in NPZ and CSV formats.

The graphical interface contains five tabs:

1. **Settings**;
2. **Start & Results**;
3. **Phase Analysis**;
4. **GDD Compressor Design**;
5. **Parameter Sweep**.

The model is intended for studying Kerr-driven spectral broadening, linear dispersion, attenuation, self-steepening, spectral phase, and the degree to which an externally applied quadratic spectral phase can compress the pulse.

2 Notation, Coordinates, and Units

The principal quantities used throughout this document are defined here before they enter the propagation equations.

Symbol	**Definition**
z	Propagation coordinate along the fiber, in metres.
L	Total fiber length, in metres. The GUI displays it in centimetres.
t	Laboratory time, in seconds.
T	Retarded time, $T = t - \beta_1 z$, in seconds.
$U(z,T)$	Dimensionless complex pulse envelope used by the numerical model.
$A(z,T)$	Power-normalized complex envelope, $A = \sqrt{P_0}\,U$, with units $\sqrt{\mathrm{W}}$.
$P(z,T)$	Instantaneous pulse power, $P = P_0\lvert U\rvert^2$, in watts.
P_0	Peak-power normalization associated with the transform-limited input shape, in watts.
E_p	Pulse energy, $E_\mathrm{p} = \int P(T)\,\mathrm{d}T$, in joules.
λ_0	Carrier wavelength, in metres; displayed in nanometres.
$\nu_0 = f_0$	Carrier ordinary frequency, $\nu_0 = c/\lambda_0$, in hertz.
ω_0	Carrier angular frequency, $\omega_0 = 2\pi\nu_0$, in rad/s.
f_FFT	Frequency coordinate generated by the centred NumPy FFT, in hertz.
ν	Physical optical frequency, $\nu = \nu_0 - f_\mathrm{FFT}$, in hertz.
ω	Physical optical angular frequency, $\omega = 2\pi\nu$, in rad/s.
Ω	Physical angular-frequency detuning, $\Omega = \omega - \omega_0 = -2\pi f_\mathrm{FFT}$.
$\widetilde{U}$	Fourier-domain envelope corresponding to $U(T)$.
$\beta(\omega)$	Modal propagation constant, in rad/m.
β_j	jth derivative of β with respect to ω at ω_0: $\beta_j = (\partial^j\beta/\partial\omega^j)_{\omega_0}$.
β_1	Inverse group velocity, in s/m.
β_2	Group-velocity dispersion coefficient, in s^2/m; displayed as fs^2/m.
β_3	Third-order dispersion coefficient, in s^3/m; displayed as fs^3/m.
α	Power attenuation coefficient, in m^{-1}. A field amplitude therefore contains the factor $\exp(-\alpha z/2)$ when α is constant.
γ	Kerr nonlinear coefficient, in $\mathrm{W}^{-1}\mathrm{m}^{-1}$.
R	Capillary radius, in metres; displayed in micrometres.
A_eff	Effective modal area, in m^2.
n_gas	Linear refractive index of the filling gas.
n_2	Nonlinear Kerr refractive index, in m^2/W for a specified pressure.
p	Gas-pressure value entered in the GUI. Built-in coefficients use this value in the pressure-scaling relations of the gas model.

The speed of light in vacuum is

$$c = 299792458\ \mathrm{m/s}. \tag{1}$$

2.1 Fourier convention

The centred numerical transform corresponds to the continuous convention

$$\widetilde{U}(f_\mathrm{FFT}) = \int_{-\infty}^{+\infty} U(T)e^{-i2\pi f_\mathrm{FFT}T}\,\mathrm{d}T. \tag{2}$$

Consequently,

$$\frac{\partial}{\partial T} \longleftrightarrow i2\pi f_{\mathrm{FFT}}, \tag{3}$$

while the physical optical detuning used in plots and phase formulas is

$$\boxed{\Omega = -2\pi f_{\mathrm{FFT}}.} \tag{4}$$

The physical optical-frequency axis shown by the GUI is

$$\boxed{\nu = \nu_0 - f_{\mathrm{FFT}}.} \tag{5}$$

When the spectral axis is displayed as wavelength,

$$\lambda = \frac{c}{\nu}, \tag{6}$$

and only points with positive physical frequency are used.

3 Physical Background

3.1 Kerr refractive index and self-phase modulation

For an instantaneous Kerr response, the refractive index is written

$$n = n_0 + n_2 I, \tag{7}$$

where n_0 is the linear refractive index, n_2 is the Kerr nonlinear index, and I is optical intensity in $\mathrm{W/m^2}$. For a guided mode with effective area A_{eff}, intensity and power are related approximately by

$$I(T) \simeq \frac{P(T)}{A_{\mathrm{eff}}}. \tag{8}$$

The nonlinear propagation constant can be expressed with the nonlinear coefficient

$$\boxed{\gamma = \frac{n_2 \omega_0}{c A_{\mathrm{eff}}}.} \tag{9}$$

If depletion and loss are neglected over a short distance z, the Kerr phase is approximately

$$\phi_{\mathrm{NL}}(T, z) \simeq \gamma P(T) z. \tag{10}$$

The instantaneous optical-frequency shift generated by this temporal phase is

$$\boxed{\Delta\omega(T) = -\frac{\partial \phi_{\mathrm{NL}}}{\partial T}.} \tag{11}$$

The same relation in ordinary frequency is

$$\boxed{\Delta\nu(T) = -\frac{1}{2\pi}\frac{\partial \phi_{\mathrm{NL}}}{\partial T}.} \tag{12}$$

The leading and trailing edges of a smooth pulse therefore acquire opposite frequency shifts, producing self-phase-modulation spectral broadening.

A commonly used measure of accumulated nonlinear phase is the B-integral,

$$B = \int_0^L \gamma P_{\mathrm{peak}}(z)\,\mathrm{d}z, \tag{13}$$

where $P_{\mathrm{peak}}(z)$ is the maximum instantaneous power at position z. The B-integral is dimensionless and is expressed in radians of nonlinear phase.

3.2 Linear dispersion

The modal propagation constant is expanded around ω_0 as

$$\beta(\omega_0 + \Omega) = \beta_0 + \beta_1\Omega + \frac{\beta_2}{2}\Omega^2 + \frac{\beta_3}{6}\Omega^3 + \cdots, \tag{14}$$

where

$$\beta_0 = \beta(\omega_0), \qquad \beta_1 = \left.\frac{\partial\beta}{\partial\omega}\right|_{\omega_0}, \qquad \beta_2 = \left.\frac{\partial^2\beta}{\partial\omega^2}\right|_{\omega_0}, \qquad \beta_3 = \left.\frac{\partial^3\beta}{\partial\omega^3}\right|_{\omega_0}. \tag{15}$$

The carrier phase associated with β_0 is not part of the envelope dynamics. The first-order term is removed by using retarded time $T = t - \beta_1 z$. FibDisp therefore propagates the residual second- and third-order dispersion through the scalar coefficients β_2 and β_3 evaluated at the carrier.

3.3 Self-steepening

For a sufficiently broadband pulse, the frequency dependence of the nonlinear coupling cannot be treated as exactly constant. To first order,

$$\gamma(\omega) \simeq \gamma_0 \left(1 + \frac{\Omega}{\omega_0}\right), \tag{16}$$

where $\gamma_0 = \gamma(\omega_0)$. The nonlinear spectral term is therefore

$$i\gamma_0 \left(1 + \frac{\Omega}{\omega_0}\right) \widetilde{|A|^2 A}. \tag{17}$$

Using the Fourier convention defined above, this becomes in retarded time

$$\boxed{i\gamma_0|A|^2A - \frac{\gamma_0}{\omega_0}\frac{\partial}{\partial T}\left(|A|^2A\right).} \tag{18}$$

The first term is SPM. The derivative term is self-steepening. For positive n_2, the nonlinear group-delay correction is larger in the high-power part of the pulse, producing an intensity-dependent temporal delay and an optical shock tendency on the trailing side of the pulse.

3.4 Why spectral broadening can enable compression

Let the fiber output spectrum be

$$\widetilde{U}_{\text{out}}(\Omega) = \left|\widetilde{U}_{\text{out}}(\Omega)\right| e^{i\phi_{\text{out}}(\Omega)}, \tag{19}$$

where $\phi_{\text{out}}(\Omega)$ is the output spectral phase. Spectral broadening increases the bandwidth and can therefore support a shorter transform-limited pulse. The actual output is usually longer because ϕ_{out} is not flat.

An external phase-only compressor with transfer function

$$H(\Omega) = e^{i\phi_{\text{comp}}(\Omega)} \tag{20}$$

produces

$$U_{\text{comp}}(T) = \mathcal{F}^{-1}\left\{\widetilde{U}_{\text{out}}(\Omega)H(\Omega)\right\}. \tag{21}$$

A pure-GDD compressor uses

$$\boxed{\phi_{\text{comp}}(\Omega) = \frac{G}{2}\Omega^2,} \tag{22}$$

where G is the applied GDD in fs^2 when Ω is expressed in rad/fs. A value G that has the opposite sign to the dominant quadratic part of the output phase can reduce the pulse duration.

4 Input Pulse Model

The user specifies the transform-limited intensity FWHM τ_{FWHM}, the pulse energy E_{p}, and one of three temporal shapes. A shape-dependent characteristic time T_0 is calculated so that the requested τ_{FWHM} is reproduced.

4.1 Gaussian pulse

The transform-limited power is

$$P_{\text{TL}}(T) = P_0 \exp\left[-\left(\frac{T}{T_0}\right)^2\right], \tag{23}$$

and the field envelope is

$$U_{\text{TL}}(T) = \exp\left[-\frac{T^2}{2T_0^2}\right]. \tag{24}$$

FibDisp uses

$$\boxed{T_0 = \frac{\tau_{\text{FWHM}}}{1.665},} \tag{25}$$

where 1.665 is the numerical FWHM factor used by the code. The energy normalization is

$$\boxed{P_0 = \frac{E_\mathrm{p}}{\sqrt{\pi} T_0}.} \tag{26}$$

4.2 Hyperbolic-secant pulse

The transform-limited power is

$$P_\mathrm{TL}(T) = P_0 \operatorname{sech}^2\left(\frac{T}{T_0}\right), \tag{27}$$

with field

$$U_\mathrm{TL}(T) = \operatorname{sech}\left(\frac{T}{T_0}\right). \tag{28}$$

The intensity FWHM gives

$$\boxed{T_0 = \frac{\tau_\mathrm{FWHM}}{2 \operatorname{acosh}(\sqrt{2})},} \tag{29}$$

and, because $\int_{-\infty}^{+\infty} \operatorname{sech}^2(T/T_0)\,\mathrm{d}T = 2T_0$,

$$\boxed{P_0 = \frac{E_\mathrm{p}}{2T_0}.} \tag{30}$$

4.3 Super-Gaussian pulse

The transform-limited power is

$$P_\mathrm{TL}(T) = P_0 \exp\left[-\left(\frac{T}{T_0}\right)^4\right], \tag{31}$$

with field

$$U_\mathrm{TL}(T) = \exp\left[-\frac{1}{2}\left(\frac{T}{T_0}\right)^4\right]. \tag{32}$$

The characteristic time is

$$\boxed{T_0 = \frac{\tau_\mathrm{FWHM}}{2(\ln 2)^{1/4}}.} \tag{33}$$

Using

$$\int_{-\infty}^{+\infty} \exp[-(T/T_0)^4]\,\mathrm{d}T = 2T_0\Gamma(5/4), \tag{34}$$

where Γ is the gamma function, the peak-power normalization is

$$\boxed{P_0 = \frac{E_\mathrm{p}}{2T_0\Gamma(5/4)}.} \tag{35}$$

4.4 Input GDD and TOD

After constructing the transform-limited field, FibDisp applies the requested input spectral phase

$$\boxed{\phi_\mathrm{in}(\Omega) = \frac{\mathrm{GDD}_\mathrm{in}}{2}\Omega^2 + \frac{\mathrm{TOD}_\mathrm{in}}{6}\Omega^3.} \tag{36}$$

Here GDD_in is in fs^2, TOD_in is in fs^3, and Ω is expressed in rad/fs for the numerical phase evaluation. The chirped input spectrum is

$$\widetilde{U}_\mathrm{in}(\Omega) = \widetilde{U}_\mathrm{TL}(\Omega)\exp[i\phi_\mathrm{in}(\Omega)]. \tag{37}$$

Because this transfer function has unit magnitude, input GDD and TOD change the phase and temporal shape without changing the pulse energy or spectral power.

5 Gas and Hollow-Capillary Model

5.1 Built-in gases

The built-in gas choices are Helium, Neon, Argon, Krypton, Xenon, and Nitrogen. For these gases the program calculates β_2, β_3, a frequency-dependent capillary loss $\alpha(\nu)$, and the nonlinear coefficient γ from the selected gas, pressure, wavelength, and capillary radius.

The effective modal area is approximated by

$$\boxed{A_\mathrm{eff} = 0.4766\,\pi R^2.} \tag{38}$$

5.2 Gas refractive index and pressure scaling

For each gas a wavelength-dependent reference refractive-index model is used. The pressure dependence is applied through a Lorentz–Lorenz form. If $n_\mathrm{ref}(\lambda)$ is the reference index and

$$\chi(\lambda, p) = \frac{n_\mathrm{ref}^2(\lambda) - 1}{n_\mathrm{ref}^2(\lambda) + 2}\,p, \tag{39}$$

then the gas index used by the code is

$$\boxed{n_\mathrm{gas}(\lambda, p) = \sqrt{\frac{1+2\chi}{1-\chi}}.} \tag{40}$$

The pressure variable p is the numerical pressure value entered in the GUI. The nonlinear index is also scaled linearly with this pressure value in the built-in model.

5.3 Large-core capillary approximation

The fundamental capillary-mode transverse eigenvalue is

$$u_{11} \simeq 2.405. \tag{41}$$

The exact ideal-capillary propagation constant can be written

$$\beta(\omega) = \sqrt{n_{\mathrm{gas}}^2(\omega)\frac{\omega^2}{c^2} - \frac{u_{11}^2}{R^2}}. \tag{42}$$

FibDisp evaluates dispersion through the corresponding large-core expansion of the effective index,

$$\boxed{n_{\mathrm{eff}}(\lambda) \simeq n_{\mathrm{gas}}(\lambda)\left[1 - \frac{1}{2}\left(\frac{u_{11}\lambda}{2\pi R n_{\mathrm{gas}}(\lambda)}\right)^2\right].} \tag{43}$$

The wavelength derivatives of n_{eff} are evaluated numerically in a local wavelength interval around λ_0. The resulting carrier coefficients are

$$\boxed{\beta_2 = \frac{\lambda_0^3}{2\pi c^2}\left.\frac{\mathrm{d}^2 n_{\mathrm{eff}}}{\mathrm{d}\lambda^2}\right|_{\lambda_0},} \tag{44}$$

and

$$\boxed{\beta_3 = -\frac{\lambda_0^4}{4\pi^2 c^3}\left[3\frac{\mathrm{d}^2 n_{\mathrm{eff}}}{\mathrm{d}\lambda^2} + \lambda_0\frac{\mathrm{d}^3 n_{\mathrm{eff}}}{\mathrm{d}\lambda^3}\right]_{\lambda_0}.} \tag{45}$$

Thus the displayed β_2 and β_3 already contain both gas and waveguide contributions. During propagation they are used as scalar coefficients in the Taylor expansion around ω_0; the full $\beta(\omega)$ is not propagated directly.

5.4 Nonlinear coefficient

For the built-in gases the nonlinear coefficient is

$$\boxed{\gamma = \frac{n_{2,1}\, p\, \omega_0}{c A_{\mathrm{eff}}},} \tag{46}$$

where $n_{2,1}$ is the gas nonlinear-index coefficient stored by the model per unit pressure and p is the pressure value entered in the GUI. Increasing pressure therefore increases the Kerr coupling approximately linearly while it also changes the gas contribution to dispersion.

5.5 Capillary attenuation

For built-in gases the attenuation is frequency dependent. Let

$$q = \frac{n_v}{n_{\mathrm{gas}}(\lambda_0, p)}, \tag{47}$$

where $n_v = 1.45356$ is the wall-index value used by the capillary-loss model. At physical optical frequency ν, the attenuation used in the spectral operator is

$$\boxed{\alpha(\nu) = \left(\frac{u_{11}c}{2\pi\nu}\right)^2 \frac{1}{R^3} \frac{q^2+1}{\sqrt{q^2-1}}.} \tag{48}$$

The Settings tab displays $\alpha(\nu_0)$, the value at the carrier, while the propagation uses the frequency-dependent array $\alpha(\nu)$.

5.6 Manual propagation coefficients

Manual mode bypasses the built-in coefficient calculation. The user directly specifies

$$\beta_2\ [\mathrm{fs}^2/\mathrm{m}], \qquad \beta_3\ [\mathrm{fs}^3/\mathrm{m}], \qquad \alpha\ [\mathrm{m}^{-1}], \qquad \gamma\ [\mathrm{W}^{-1}\mathrm{m}^{-1}]. \tag{49}$$

In Manual mode α is constant across the simulated spectrum. The direct coefficients are otherwise used in the same propagation equation as the built-in gas coefficients.

6 Propagation Equation

6.1 Linear spectral operator

With retarded time $T = t - \beta_1 z$ and physical detuning $\Omega = \omega - \omega_0$, the linear part of the implemented model is

$$\boxed{\left.\frac{\partial \widetilde{U}}{\partial z}\right|_{\mathrm{L}} = D(\Omega)\widetilde{U},} \tag{50}$$

with

$$\boxed{D(\Omega) = -\frac{\alpha(\omega_0+\Omega)}{2} + i\frac{\beta_2}{2}\Omega^2 + i\frac{\beta_3}{6}\Omega^3.} \tag{51}$$

For Manual mode, α is constant. For a built-in gas, $\alpha(\omega_0+\Omega)$ is the capillary-loss function described above.

6.2 Nonlinear envelope equation

The numerical envelope is normalized according to

$$A(z,T) = \sqrt{P_0}\,U(z,T), \qquad P(z,T) = P_0|U(z,T)|^2. \tag{52}$$

The nonlinear part of the propagation equation is

$$\boxed{\left.\frac{\partial U}{\partial z}\right|_{\mathrm{NL}} = i\gamma P_0|U|^2U - \frac{\gamma P_0}{\omega_0}\frac{\partial}{\partial T}\left(|U|^2U\right).} \tag{53}$$

Combining linear and nonlinear contributions gives the exact form used by the split-step implementation:

$$\boxed{\frac{\partial U}{\partial z} = \mathcal{F}^{-1}\left\{D(\Omega)\widetilde{U}\right\} + i\gamma P_0|U|^2U - \frac{\gamma P_0}{\omega_0}\frac{\partial}{\partial T}\left(|U|^2U\right).} \tag{54}$$

If α is constant, Eq. (54) can equivalently be written in time-domain differential form as

$$\begin{aligned}\frac{\partial U}{\partial z} = &-\frac{\alpha}{2}U - i\frac{\beta_2}{2}\frac{\partial^2 U}{\partial T^2} + \frac{\beta_3}{6}\frac{\partial^3 U}{\partial T^3} \\ &+ i\gamma P_0|U|^2U - \frac{\gamma P_0}{\omega_0}\frac{\partial}{\partial T}\left(|U|^2U\right).\end{aligned} \tag{55}$$

The physical roles are:

Contribution	**Physical effect**
$-\alpha U/2$	Linear field attenuation when α is constant.
$-i(\beta_2/2)\partial^2U/\partial T^2$	Group-velocity dispersion.
$(\beta_3/6)\partial^3U/\partial T^3$	Third-order dispersion.
$i\gamma P_0|U|^2U$	Kerr self-phase modulation.
$-(\gamma P_0/\omega_0)\partial_T(|U|^2U)$	Self-steepening / optical-shock term.

Each contribution can be enabled or disabled in the Start & Results tab. The self-steepening term is physically part of the broadband Kerr response; enabling it without SPM is mainly useful as a numerical diagnostic.

7 Numerical Self-Steepening Solvers

FibDisp offers two numerical treatments of the nonlinear substep when self-steepening is enabled.

7.1 Fast exponential solver

Define

$$Q(U) = \frac{\partial}{\partial T}\left(|U|^2U\right). \tag{56}$$

The nonlinear equation can formally be grouped as

$$\left.\frac{\partial U}{\partial z}\right|_{\mathrm{NL}} = \left[i\gamma P_0|U|^2 - \frac{\gamma P_0}{\omega_0}\frac{Q(U)}{U}\right]U. \tag{57}$$

The fast solver freezes the nonlinear coefficients over one global SSFM step Δz and applies multiplicative exponentials. The shock ratio is regularized as

$$\frac{Q(U)}{U+\epsilon}, \qquad \epsilon = 0.002, \tag{58}$$

where U is dimensionless and has a characteristic peak amplitude of order unity before strong reshaping. The temporal derivative is evaluated spectrally. This method is computationally efficient and is suitable when the nonlinear evolution is adequately resolved by the global propagation step.

7.2 Accurate adaptive RK4 solver

The accurate solver integrates the nonlinear right-hand side directly,

$$F(U) = i\gamma P_0|U|^2U - \frac{\gamma P_0}{\omega_0}\frac{\partial}{\partial T}\left(|U|^2U\right), \tag{59}$$

with classical fourth-order Runge–Kutta stages. The derivative $\partial_T(|U|^2U)$ is recomputed at every RK4 stage, so no division by $U + \epsilon$ is required.

Each global nonlinear interval Δz is automatically divided into N_{sub} internal steps. Let

$$I_U = \max_T |U(T)|^2, \qquad g = |\gamma P_0| I_U. \tag{60}$$

The substep count is selected so that the estimated nonlinear phase increment per internal step does not exceed approximately 0.20 rad and the estimated self-steepening Courant number does not exceed approximately 0.25:

$$\frac{g\Delta z}{N_{\text{sub}}} \lesssim 0.20, \tag{61}$$

$$\frac{3g\Delta z}{\omega_0 \Delta T\, N_{\text{sub}}} \lesssim 0.25, \tag{62}$$

where ΔT is the temporal grid spacing. The factor $3g/\omega_0$ is an estimate of the intensity-characteristic speed associated with the reduced self-steepening equation and is used only as a numerical step-size criterion.

The accurate mode is slower but is substantially more robust when the nonlinear coefficient, pulse peak power, or optical-shock strength is large. If more than 5000 internal RK4 steps would be required inside one global SSFM interval, FibDisp stops the calculation and asks for a finer propagation or temporal grid.

No time integrator can compensate for an unresolved temporal shock or a spectrum reaching the FFT boundaries. Strong self-steepening calculations should therefore also be checked for convergence with respect to N_Z, FFT size, and temporal window.

8 Split-Step Fourier Propagation

The full propagation uses a symmetric split-step Fourier method. Let $\widehat{D}$ denote the linear propagation operator and $\widehat{N}$ the nonlinear operator. Over one interval Δz,

$$U(z + \Delta z) \simeq e^{\widehat{D}\Delta z/2} e^{\widehat{N}\Delta z} e^{\widehat{D}\Delta z/2} U(z). \tag{63}$$

The numerical sequence is:

1. Fourier transform the temporal field;

2. multiply the spectrum by $\exp[D(\Omega)\Delta z/2]$;
3. inverse transform to retarded time;
4. apply the nonlinear substep with the selected self-steepening solver;
5. Fourier transform again;
6. apply the second factor $\exp[D(\Omega)\Delta z/2]$;
7. inverse transform to obtain the field at the next propagation point.

If N_Z is the number of points on the propagation axis, then there are $N_Z - 1$ intervals and

$$\boxed{\Delta z = \frac{L}{N_Z - 1}.} \tag{64}$$

The first point is $z = 0$ and the final point is $z = L$.

9 Temporal and Spectral Numerical Grids

The FFT contains

$$\boxed{N = 2^n} \tag{65}$$

samples, where n is the GUI value `Points time/freq 2^n`.

Let m denote the GUI value `ratio time/FWHM m`. In the implemented grid construction, m multiplies the shape-dependent characteristic time T_0. The periodic temporal interval is

$$\boxed{-mT_0 \leq T < mT_0,} \tag{66}$$

with total duration

$$T_{\text{win}} = 2mT_0. \tag{67}$$

The temporal spacing is

$$\boxed{\Delta T = \frac{2mT_0}{N},} \tag{68}$$

and the FFT-frequency spacing is

$$\boxed{\Delta f = \frac{1}{N\Delta T} = \frac{1}{T_{\text{win}}}.} \tag{69}$$

At fixed m, increasing n increases the temporal resolution and the available Nyquist bandwidth. At fixed n, increasing m enlarges the temporal window but also increases ΔT; strong broadening can therefore require increasing both m and n.

The `Saved z samples` value controls only how many propagation snapshots are retained for the temporal/spectral evolution map. It does not change N_Z, Δz, or the propagation physics.

10 Computational-Boundary Warnings

During propagation, FibDisp monitors the first and last samples of the temporal and spectral fields. A warning is generated when an edge amplitude exceeds 10^{-3} of the corresponding initial peak field amplitude. In symbols, a frequency-boundary warning is triggered when

$$\frac{|\widetilde{U}_{\mathrm{edge}}|}{\max|\widetilde{U}_{\mathrm{in}}|} > 10^{-3}, \tag{70}$$

and the analogous test is applied in time.

A boundary warning indicates that periodic FFT wrap-around may influence the solution. The usual remedies are:

- increase n when the spectrum approaches the frequency boundaries;
- increase m when the temporal pulse approaches the time boundaries;
- increase both when a larger temporal window is needed without losing temporal resolution.

Parameter sweeps perform the same check for every sweep point and report the boundary type and approximate propagation position at which the condition was first detected.

11 Derived Quantities

11.1 Output energy and transmission

The calculated spectral power density is denoted $S(\nu)$ and has units W/Hz. The output energy is

$$\boxed{E_{\mathrm{out}} = \int S_{\mathrm{out}}(\nu)\,\mathrm{d}\nu.} \tag{71}$$

The transmission percentage is

$$\boxed{\eta = 100\frac{E_{\mathrm{out}}}{E_{\mathrm{in}}},} \tag{72}$$

where E_{in} is the input pulse energy.

11.2 Spectral broadening factor

For a spectral density $S(f)$, define the power-weighted mean frequency

$$\bar{f} = \frac{\int f S(f)\,\mathrm{d}f}{\int S(f)\,\mathrm{d}f} \tag{73}$$

and standard deviation

$$\sigma_f = \sqrt{\frac{\int (f-\bar{f})^2 S(f)\,\mathrm{d}f}{\int S(f)\,\mathrm{d}f}}. \tag{74}$$

FibDisp reports the dimensionless broadening factor

$$\boxed{B = \frac{\sigma_{f,\text{out}}}{\sigma_{f,\text{in}}}.} \tag{75}$$

The sign reversal between f_{FFT} and physical detuning does not change this standard-deviation ratio.

11.3 Transform-limited output pulse

The transform-limited field associated with the output spectral amplitude is

$$\boxed{U_{\text{TL}}(T) = \mathcal{F}^{-1}\left\{|\widetilde{U}_{\text{out}}(\Omega)|\right\}.} \tag{76}$$

Its temporal FWHM is the duration obtainable if the complete output spectral phase could be removed while preserving the spectral amplitude.

11.4 Temporal instantaneous-frequency shift

Let

$$\phi_T(T) = \arg U(T) \tag{77}$$

be the unwrapped temporal envelope phase. The displayed chirp is the ordinary frequency shift

$$\boxed{\Delta\nu(T) = -\frac{1}{2\pi}\frac{\mathrm{d}\phi_T}{\mathrm{d}T}.} \tag{78}$$

FibDisp stores this quantity in hertz and displays it in terahertz.

11.5 Spectral group delay

Let $\phi(\omega)$ be the unwrapped spectral phase. The group delay is

$$\boxed{GD(\omega) = \frac{\mathrm{d}\phi}{\mathrm{d}\omega}.} \tag{79}$$

It is stored in seconds and displayed in femtoseconds. With the internal FFT coordinate, the same derivative is evaluated as

$$GD = -\frac{1}{2\pi}\frac{\mathrm{d}\phi}{\mathrm{d}f_{\text{FFT}}}. \tag{80}$$

12 Tab 1: Settings

The Settings tab defines the baseline parameters used for a propagation or a parameter sweep.

12.1 Gas and fiber parameters

GUI field	**Meaning**
Select gas	Built-in gas model or Manual direct-coefficient mode.
Pressure (atm)	Pressure value p used by the built-in gas model.
Fiber length (cm)	Total propagation distance L.
Radius (μm)	Capillary radius R.

12.2 Pulse parameters

GUI field	**Meaning**
Pulse shape	Gaussian, Sech, or Super-Gaussian transform-limited intensity profile.
Energy (mJ)	Input pulse energy E_p.
TL FWHM (fs)	Transform-limited intensity FWHM τ_FWHM.
Wavelength (nm)	Carrier wavelength λ_0.
GDD (fs^2)	Input quadratic spectral-phase coefficient GDD_in.
TOD (fs^3)	Input cubic spectral-phase coefficient TOD_in.

12.3 Simulation parameters

GUI field	**Meaning**
Points prop. axis z	Number N_Z of propagation-axis samples.
Points time/freq 2^n	FFT sample count $N = 2^n$.
ratio time/FWHM m	Numerical half-window factor; the implemented half-width is mT_0.

12.4 Propagation coefficients

For a built-in gas, the tab displays β_2, β_3, $\alpha(\nu_0)$, and γ. The values update when gas, pressure, wavelength, or radius is changed. The coefficient fields are read-only in this mode.

In Manual mode the four fields become editable. The value entered for α is then used as a frequency-independent loss coefficient.

12.5 Confirm (preview input)

The preview button generates the input pulse using the selected shape, energy, FWHM, wavelength, GDD, TOD, and numerical grid. It displays temporal power and phase together with spectral power density and phase before fiber propagation.

13 Tab 2: Start & Results

This tab performs the propagation and displays the principal output diagnostics.

13.1 Scalar results

The reported scalar values are:

- output pulse energy E_{out};
- transmission η;
- spectral broadening factor B;
- transform-limited output FWHM.

13.2 Selectable propagation terms

The following terms can be independently enabled or disabled:

- SPM;
- self-steepening;
- GVD (β_2);
- TOD (β_3);
- loss (α).

The self-steepening solver can be selected as either **Fast (exponential)** or **Accurate RK4 (adaptive)**. The solver choice has no effect when self-steepening is disabled.

13.3 Raw and fixed-GDD result display

The **Result display** control has two modes:

Raw output displays the complex field directly at the fiber exit.

Fixed GDD applies a user-selected external GDD G to the final output spectrum and reconstructs the corresponding temporal field.

The fixed GDD is phase-only post-processing:

$$\widetilde{U}_G(\Omega) = \widetilde{U}_{\text{out}}(\Omega) \exp\left(i\frac{G}{2}\Omega^2\right). \tag{81}$$

It does not modify the fiber propagation, the saved z evolution, the raw simulation result used by Phase Analysis, or the GDD Compressor Design tab.

For a fixed-GDD temporal display, the automatic x-range is obtained from the first and last samples above 1% of the compensated pulse peak, with additional padding equal to half of that significant span on each side when the numerical window permits. This rule affects only the displayed axis limits; it does not modify the temporal field or the computational time grid.

13.4 Result plots

The 2×3 layout contains:

1. pulse power and temporal phase;
2. spectral power density and spectral phase;
3. transform-limited pulse power;
4. instantaneous-frequency shift (chirp);
5. spectral group delay;
6. propagation evolution map.

The spectral axis can be displayed in THz or nm. The Matplotlib navigation toolbar provides zoom, pan, home, and save functions and remains outside the resizable plot area so that it stays accessible when the window height is reduced.

13.5 Spectral or temporal evolution along the fiber

The lower-right panel can display either

$$S(\nu, z) \tag{82}$$

or

$$P(T, z), \tag{83}$$

where S is spectral power density and P is temporal power. Both maps use the same saved propagation positions. The number of stored positions is controlled by `Saved z samples`.

Both maps always represent the field *inside the fiber before any external fixed-GDD compensation.* An external compressor is applied only after the final propagation position and therefore has no physical meaning at intermediate z.

13.6 Saving a single simulation

A simulation can be saved as NPZ or CSV. The NPZ archive retains the numerical arrays, including complex temporal and spectral fields. CSV export writes the main table and companion files for the propagation maps.

The saved quantities include:

- temporal grid and FFT-frequency grid;
- input and output complex fields;
- input and output temporal power;
- input and output spectral power density;
- temporal and spectral phase;

- instantaneous-frequency shift;
- spectral group delay;
- transform-limited pulse;
- spectral evolution $S(\nu, z)$;
- temporal evolution $P(T, z)$;
- physical and numerical settings associated with the run.

14 Tab 3: Phase Analysis

The Phase Analysis tab analyzes the **raw fiber output**. The optional fixed GDD selected only for the Start & Results display is not included in this analysis.

14.1 Polynomial convention

The fit variable is physical angular-frequency detuning in rad/fs. The fitted phase is written

$$\boxed{\phi_{\text{fit}}(\Omega) = p_0 + p_1\Omega + \frac{p_2}{2!}\Omega^2 + \frac{p_3}{3!}\Omega^3 + \cdots .} \tag{84}$$

The coefficients therefore have direct dispersion units:

$$p_0 \quad [\text{rad}], \tag{85}$$

$$p_1 \quad [\text{fs}], \tag{86}$$

$$p_2 \quad [\text{fs}^2], \tag{87}$$

$$p_3 \quad [\text{fs}^3]. \tag{88}$$

In particular, p_1 is a fitted group-delay offset, p_2 is the fitted GDD, and p_3 is the fitted TOD.

14.2 Spectral-power-weighted least squares

For spectral samples indexed by i, let S_i be the output spectral power density, ϕ_i the unwrapped output phase, and Ω_i the physical detuning. The fit minimizes

$$\boxed{\chi^2 = \sum_i S_i \left[\phi_i - \phi_{\text{fit}}(\Omega_i)\right]^2 .} \tag{89}$$

The weights are normalized internally, so only their relative values matter. Strong spectral regions therefore dominate over weak wings with poorly conditioned phase.

14.3 Fit range and display axes

The fit interval can be entered in THz or nm. Changing the displayed unit converts the limits so that the same physical spectral interval is preserved. The fitted quantity is always computed using Ω in rad/fs.

The spectral-phase plot automatically chooses its y-limits from the phase data that lie inside the currently displayed x-range. When a polynomial fit is present, the visible part of the fitted curve is included in the same y-range calculation. A small margin is added above and below. This display scaling does not alter phase unwrapping, fit limits, weights, or fitted coefficients.

The fitted GDD is a global, spectral-power-weighted quadratic component over the selected interval. After strong nonlinear broadening it need not equal the local curvature

$$\left.\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2}\right|_{\omega_0}. \tag{90}$$

15 Tab 4: GDD Compressor Design

This tab applies a pure quadratic spectral phase to the raw fiber output and reconstructs the temporal field.

15.1 Pure-GDD transfer function

For applied GDD G,

$$\boxed{H_G(\Omega) = \exp\left(i\frac{G}{2}\Omega^2\right)}, \tag{91}$$

and

$$\widetilde{U}_G(\Omega) = \widetilde{U}_{\text{out}}(\Omega)H_G(\Omega). \tag{92}$$

Since $|H_G| = 1$, the spectral power density is unchanged. Only spectral phase and the reconstructed temporal field change.

15.2 Manual GDD

The user can enter any value of G in fs^2 and inspect the compensated pulse, the raw fiber output, the transform-limited pulse, the spectrum, and the compensated spectral phase.

15.3 Temporal quality metrics

Four metrics are available for characterizing or optimizing the compressed pulse.

15.3.1 Main FWHM

The main FWHM is the temporal distance between the nearest half-maximum crossings on either side of the global pulse maximum. It describes the main peak but can ignore separated satellites.

15.3.2 RMS duration

For temporal power $P(T)$, define the power-weighted centroid

$$\bar{T} = \frac{\int T P(T)\,\mathrm{d}T}{\int P(T)\,\mathrm{d}T}. \tag{93}$$

The RMS duration is

$$\boxed{\sigma_T = \sqrt{\frac{\int (T-\bar{T})^2 P(T)\,\mathrm{d}T}{\int P(T)\,\mathrm{d}T}}.} \tag{94}$$

This metric is sensitive to the complete temporal profile, including weak pedestals and distant satellites.

15.3.3 95% energy width

The 95% energy width is the shortest contiguous interval $[T_1, T_2]$ that contains 95% of the total temporal energy:

$$\boxed{\Delta T_{95} = T_2 - T_1,} \tag{95}$$

with

$$\frac{\int_{T_1}^{T_2} P(T)\,\mathrm{d}T}{\int_{-\infty}^{+\infty} P(T)\,\mathrm{d}T} \geq 0.95, \tag{96}$$

and $T_2 - T_1$ chosen as small as possible.

15.3.4 95%-bounded FWHM

This is the default optimization metric. First, the shortest interval containing 95% of the pulse energy is found. Starting from each boundary of that interval and moving toward the global peak, the first crossing of $P_{\max}/2$ is located. The distance between those two crossings is the 95%-bounded FWHM.

The metric remains sensitive to sufficiently strong pre-pulses and post-pulses inside the 95%-energy interval while retaining an FWHM-like interpretation.

15.4 Optimized GDD

The GDD objective can contain several local minima. FibDisp therefore combines:

1. a spectral-phase estimate used to form the seed $G \simeq -p_2$;
2. a broad global GDD scan;
3. identification of promising minima;
4. progressively finer local grids around several candidate basins;
5. retention of the best objective value encountered;
6. a small final local refinement when useful.

The optimized GDD is the value that minimizes the selected temporal quality metric. It is an operational compressor setting and is not required to equal the negative of the fitted GDD when higher-order or structured spectral phase is important.

16 Tab 5: Parameter Sweep

The Parameter Sweep tab uses the current Settings values, propagation-term switches, self-steepening solver, and Phase Analysis configuration as the baseline for a family of calculations.

16.1 Sweep parameters

The selectable sweep parameter is one of:

- pressure p;
- fiber length L;
- capillary radius R;
- pulse energy E_p;
- transform-limited input FWHM τ_FWHM;
- carrier wavelength λ_0;
- input GDD;
- input TOD;
- direct β_2;
- direct β_3;
- direct α;
- direct γ;
- compensating output GDD.

When β_2, β_3, α, or γ is selected as the sweep parameter, only that coefficient is replaced at each sweep point. All other baseline physical parameters and direct coefficients remain fixed.

For a built-in gas:

- a β_2, β_3, or γ override leaves the calculated frequency-dependent capillary loss $\alpha(\nu)$ unchanged;
- an α override replaces the calculated loss with the selected constant attenuation value over the whole spectrum.

16.2 Sweep values and spacing

A sweep is defined by a minimum x_{min}, maximum x_{max}, and number of values N_{sweep}. Both endpoints are included.

Let

$$u_j = \frac{j}{N_{\mathrm{sweep}} - 1}, \qquad j = 0, 1, \ldots, N_{\mathrm{sweep}} - 1. \tag{97}$$

For linear spacing,

$$x(u) = x_{\mathrm{min}} + (x_{\mathrm{max}} - x_{\mathrm{min}})u. \tag{98}$$

For exponential spacing with base $b > 0$, $b \neq 1$,

$$\boxed{s_{\mathrm{exp}}(u) = \frac{b^u - 1}{b - 1},} \tag{99}$$

and

$$x(u) = x_{\mathrm{min}} + (x_{\mathrm{max}} - x_{\mathrm{min}})s_{\mathrm{exp}}(u). \tag{100}$$

For logarithmic spacing,

$$\boxed{s_{\mathrm{log}}(u) = \frac{\ln[1 + (b-1)u]}{\ln b},} \tag{101}$$

with the same mapping to $[x_{\mathrm{min}}, x_{\mathrm{max}}]$. Both normalized mappings approach linear spacing as $b \to 1$ and can be used for signed parameter ranges because the logarithm or exponential acts on the normalized coordinate u, not directly on the physical parameter.

The Preview field shows the exact list of values before the sweep starts.

16.3 Compressor criterion

The sweep optimizer has its own compressor-criterion selector. The available metrics are:

- 95%-bounded FWHM;
- RMS duration;
- 95% energy width;
- Main FWHM.

The selected criterion is used to determine the optimized GDD and optimized compressed duration at each relevant sweep point.

16.4 Quantities calculated at each point

The sweep result contains, where applicable:

- phase-fit GDD;
- optimized GDD;
- actual raw output FWHM;
- transform-limited output duration;
- optimized compressed duration;
- boundary-warning information;
- raw temporal and spectral output fields needed for post-fiber compression views.

The sweep table reports the optimized compressed duration as a stable numerical reference. The selectable compressed plots described below may instead display a fixed or swept GDD without changing the stored raw fiber outputs.

16.5 Phase-fit range used during a sweep

If the Phase Analysis tab contains a valid fit range, the sweep uses that physical range. If no valid range is available, an automatic significant spectral interval is selected independently for each sweep point. The phase fit remains spectral-power weighted.

16.6 Plot choices

The Plot selector provides:

- Phase-fit GDD;
- Optimized GDD;
- Actual output FWHM;
- TL duration;
- Compressed duration;
- Spectral power map;
- Pulse power map (uncompressed);
- Pulse power map (compressed).

The spectral map displays $S(\nu, x)$ and the temporal maps display $P(T, x)$, where x denotes the selected sweep parameter. If different sweep points use different physical frequency or time grids, the maps are interpolated onto a common grid for visualization. The original per-point grids and arrays are retained separately in the numerical result.

16.7 Compression mode for compressed plots

The **Compressed plots** selector controls both **Compressed duration** and **Pulse power map (compressed)**.

For every sweep other than a compensating-GDD sweep, the available modes are:

- **Optimized GDD**: use the independently optimized GDD for each sweep point;
- **Fixed GDD**: apply the same user-entered GDD to every sweep point.

If the sweep parameter itself is **Compensating GDD**, a third mode is available:

- **Swept GDD**: apply the GDD value represented by each point on the sweep axis.

Thus a compensating-GDD sweep offers Swept GDD, Optimized GDD, and Fixed GDD; all other sweeps offer Optimized GDD and Fixed GDD. Changing this selector is a post-processing operation on the stored complex output spectrum and does not rerun the fiber propagation.

16.8 Special propagation strategies

For a fiber-length sweep, the field is propagated once to the largest requested length. Whenever a requested intermediate length is reached, the complex field is analyzed at that physical position. If necessary, the active SSFM interval is shortened so the requested length is reached exactly.

For a compensating-GDD sweep, the fiber parameters do not change. The fiber is therefore propagated once, the optimized GDD is determined once for that raw output, and every swept GDD is then applied to the same complex spectrum.

16.9 Viewing sweep settings

Two read-only windows help identify the full parameter context of a sweep:

View current settings shows the live GUI values that would be used if a sweep were started at that moment.

View last run sweep settings shows a frozen snapshot of the settings used by the most recently completed sweep. Editing the Settings tab later does not modify this snapshot.

The snapshot includes gas/fiber parameters, pulse parameters, direct coefficients, propagation-term switches, self-steepening solver, numerical grid, sweep definition, phase-fit configuration, and compressor settings.

16.10 Saving sweep data

Sweep results can be exported as NPZ or CSV. The data include the sweep values, phase-fit and optimized GDD, durations, boundary warnings, common interpolated maps, native per-point spectral and temporal axes, raw pulse-power arrays, complex output spectra for compressor reconstruction, and baseline settings associated with the sweep.

17 Interpreting Fitted and Optimized GDD

After nonlinear propagation, the output spectral phase can contain quadratic, cubic, higher-order, and oscillatory contributions:

$$\phi(\Omega) = \phi_0 + \phi_1\Omega + \frac{\phi_2}{2}\Omega^2 + \frac{\phi_3}{6}\Omega^3 + \phi_{\mathrm{HO}}(\Omega), \tag{102}$$

where ϕ_{HO} denotes contributions beyond the displayed low-order terms.

Three different quadratic-phase concepts should be distinguished:

1. the local spectral-phase curvature
$$\left.\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2}\right|_{\omega_0};$$
2. the global spectral-power-weighted fitted GDD p_2 over a chosen spectral interval;
3. the optimized compressor GDD G_{opt}, defined as the value that minimizes a selected temporal quality metric.

For a nearly quadratic phase,

$$G_{\mathrm{opt}} \simeq -p_2. \tag{103}$$

After strong broadening, the two can differ because a pure quadratic phase cannot remove TOD, higher-order spectral phase, or rapid phase structure.

18 Installation and Running from Python

FibDisp consists of two Python source files that must be kept in the same directory:

`fibdisp_core.py` propagation physics, pulse generation, phase analysis, compressor calculations, and parameter sweeps;

`fibdisp_gui.py` Tkinter graphical interface, plotting, controls, warnings, and data export.

The principal Python dependencies are NumPy, SciPy, Matplotlib, and Tkinter. PyInstaller is needed only when creating a standalone executable.

A standalone executable is *not* required to use FibDisp. Running the Python source directly is often the simplest choice for development and scientific work.

18.1 Windows: Python virtual environment

Install a recent 64-bit Python distribution with Tk support. Python 3.12 is a suitable choice. Open PowerShell in the directory containing `fibdisp_gui.py` and `fibdisp_core.py`.

Create a local virtual environment:

```
python -m venv .venv
```

If PowerShell prevents activation of local scripts, allow scripts only for the current PowerShell process:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Activate the environment:

```
.\.venv\Scripts\Activate.ps1
```

Install the numerical packages:

```
python -m pip install --upgrade pip
python -m pip install numpy scipy matplotlib
```

Run FibDisp directly:

```
python fibdisp_gui.py
```

The packages are installed inside the local `.venv` directory. They do not need to be installed system-wide.

18.2 Linux: Python virtual environment

On Debian or Ubuntu, install Python, the virtual-environment module, and Tk if needed:

```
sudo apt update
sudo apt install python3 python3-venv python3-tk
```

Create and activate the environment:

```
python3 -m venv .venv
source .venv/bin/activate
```

Install dependencies and run:

```
python -m pip install --upgrade pip
python -m pip install numpy scipy matplotlib
python fibdisp_gui.py
```

For another Linux distribution, install the equivalent Python and Tk packages with the system package manager.

18.3 macOS: Python virtual environment

Use a Python installation with Tk support, then open Terminal in the FibDisp source directory:

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install numpy scipy matplotlib
python fibdisp_gui.py
```

If several Python installations are present, verify the selected interpreter with

```
python3 --version
```

before creating the environment.

18.4 Anaconda or Miniconda

FibDisp can also be run without compiling inside an isolated Conda environment. From Anaconda Prompt, PowerShell, or a terminal with Conda initialized:

```
conda create -n fibdisp python=3.12 numpy scipy matplotlib tk
conda activate fibdisp
python fibdisp_gui.py
```

To use an already existing Conda environment, activate it and install any missing dependencies:

```
conda activate my_environment
conda install numpy scipy matplotlib tk
python fibdisp_gui.py
```

The source directory only needs to contain `fibdisp_gui.py` and `fibdisp_core.py`; Python imports the core module from the same folder.

19 Building Standalone Executables

A standalone application bundles the Python interpreter and the required Python packages. End users can then run FibDisp without installing Python separately. The recommended PyInstaller format is `-onedir`, which creates one application directory containing the executable and its runtime libraries.

Native executables should be built on the target operating system: build the Windows application on Windows, the Linux application on Linux, and the macOS application on macOS.

19.1 Install build dependencies

Inside a virtual environment or Conda environment containing FibDisp, install PyInstaller:

```
python -m pip install pyinstaller
```

If NumPy, SciPy, and Matplotlib are not already installed, install them as well:

```
python -m pip install numpy scipy matplotlib
```

19.2 Generic PyInstaller command

Run this command from the directory containing the two FibDisp source files:

```
python -m PyInstaller --noconfirm --clean --windowed --onedir --name FibDisp fibdisp_gui.
    py
```

The `-windowed` option suppresses a separate terminal window for the GUI application. The `-onedir` option keeps all bundled runtime files inside one application directory.

19.3 Windows output

The Windows application is normally created at

```
dist\FibDisp\FibDisp.exe
```

Distribute the complete `dist\FibDisp` directory, not only `FibDisp.exe`, because the executable uses the bundled libraries stored beside it.

If a supplied PowerShell build script is used, run it from its own directory:

```
powershell -ExecutionPolicy Bypass -File .\build_windows.ps1
```

A local Python virtual environment created for building can be deleted after a successful build if no further builds are required.

19.4 Linux output

The Linux executable is normally created at

```
dist/FibDisp/FibDisp
```

Distribute the complete `dist/FibDisp` directory. For broad Linux compatibility, build on a distribution at least as old as the oldest distribution that must run the application, because the resulting binary is linked to the build system's C runtime.

19.5 macOS output

A windowed macOS build normally produces

```
dist/FibDisp.app
```

Apple Silicon and Intel binaries should be built and tested for their intended architectures. An unsigned local application can trigger Gatekeeper. Public macOS distribution should use appropriate code signing and notarization.

19.6 Build directories

PyInstaller commonly creates:

`.venv` optional local Python environment used for building;

`build` temporary PyInstaller build files;

`dist` final distributable application;

`FibDisp.spec` PyInstaller build specification.

The `build` directory can be removed after a successful build. The `.venv` directory can also be removed if it is not needed for later rebuilding. The `dist` output is the application to keep or distribute.

20 Recommended Simulation Workflow

A practical sequence is:

1. choose gas, pressure, fiber radius, fiber length, pulse shape, pulse energy, transform-limited FWHM, carrier wavelength, and input phase in **Settings**;

2. inspect the displayed β_2, β_3, α, and γ;
3. use **Confirm (preview input)** to verify the input field and numerical window;
4. select the propagation terms and self-steepening solver in **Start & Results**;
5. select the number of stored z samples needed for propagation maps;
6. run the propagation and check boundary warnings;
7. inspect raw pulse power, spectrum, chirp, group delay, transform limit, and the temporal or spectral evolution map;
8. use the fixed-GDD Result display when a quick post-fiber compensation view is useful;
9. use **Phase Analysis** to quantify the weighted spectral phase;
10. use **GDD Compressor Design** to compare phase-fit GDD with the optimized pure-GDD compressor;
11. choose a temporal optimization metric appropriate to the pulse structure, especially when satellites are present;
12. use **Parameter Sweep** to study trends and maps while keeping track of the full baseline through the settings-view buttons;
13. repeat selected calculations with finer numerical grids to verify convergence before treating small differences as physical.

21 Numerical Convergence Checks

A numerical result should be checked for convergence with respect to the propagation and FFT grids.

21.1 Propagation-step convergence

Increase N_Z, thereby reducing Δz, and compare important observables such as:

- output spectrum and temporal profile;
- broadening factor;
- transform-limited FWHM;
- phase-fit GDD;
- optimized GDD;
- optimized compressed duration.

Strong self-steepening can require substantially finer z resolution in the fast exponential mode. The accurate RK4 mode reduces this sensitivity by subdividing the nonlinear step internally, but convergence of the complete split-step solution should still be checked.

21.2 FFT-resolution convergence

Increase n so that $N = 2^n$ increases. Verify that spectral and temporal fine structure, fitted phase, and compressor results do not change significantly. Frequency-boundary warnings are a direct indication that the available bandwidth is insufficient.

21.3 Temporal-window convergence

Increase m when the pulse, temporal satellites, or dispersive tails approach the periodic time boundaries. Since increasing m at fixed N increases ΔT, increase n as well when the temporal resolution must be preserved.

21.4 Self-steepening convergence

For calculations dominated by optical-shock dynamics:

- compare Fast and Accurate RK4 solvers when practical;
- increase N_Z;
- increase FFT resolution;
- verify that the spectrum remains well separated from the FFT edges;
- verify that the sharpest temporal feature is sampled by several time points rather than one or two grid samples.

22 Model Limitations

The implemented propagation model includes Kerr nonlinearity, self-steepening, loss, and dispersion through β_3. It does not explicitly include:

- photoionization and plasma generation;
- ionization loss;
- a time-resolved molecular Raman response;
- dispersion coefficients beyond β_3 in the propagation Taylor expansion;
- propagation with the complete frequency-dependent modal $\beta(\omega)$;
- multimode transverse dynamics;
- longitudinally varying capillary radius or pressure profiles;
- real-compressor amplitude response or higher-order compressor phase beyond the applied pure quadratic phase.

The large-core capillary approximation is used to calculate the modal effective index for the built-in gas coefficients. Far from the carrier, particularly when a spectrum spans a very large fraction of an octave, a local $\beta_2 + \beta_3$ Taylor expansion can become less accurate than propagation based on the full modal dispersion relation.

For regimes where ionization, plasma dynamics, Raman response, multimode propagation, or higher-order dispersion are important, the result should be interpreted as the prediction of the model defined by Eq. (54), not as a complete description of every experimental process.

23 Summary of Sign and Phase Conventions

The principal conventions are:

- the centred FFT coordinate is f_{FFT};
- physical optical frequency is
$$\nu = \nu_0 - f_{\mathrm{FFT}};$$
- physical detuning is
$$\Omega = -2\pi f_{\mathrm{FFT}};$$
- input or compressor quadratic phase is written
$$\phi_{\mathrm{GDD}}(\Omega) = \frac{\mathrm{GDD}}{2}\Omega^2;$$
- cubic phase is written
$$\phi_{\mathrm{TOD}}(\Omega) = \frac{\mathrm{TOD}}{6}\Omega^3;$$
- temporal instantaneous-frequency shift is
$$\Delta\nu(T) = -\frac{1}{2\pi}\frac{\mathrm{d}\phi_T}{\mathrm{d}T};$$
- spectral group delay is
$$GD = \frac{\mathrm{d}\phi}{\mathrm{d}\omega};$$
- a predominantly positive output quadratic spectral phase is compensated by a negative applied GDD, and vice versa.

24 References

References

[1] M. Nisoli, S. De Silvestri, and O. Svelto, "Generation of high energy 10 fs pulses by a new pulse compression technique," *Applied Physics Letters* **68**, 2793–2795 (1996). DOI: 10.1063/1.116609.

[2] M. Nisoli, S. De Silvestri, O. Svelto, R. Szipöcs, K. Ferencz, C. Spielmann, S. Sartania, and F. Krausz, "Compression of high-energy laser pulses below 5 fs," *Optics Letters* **22**, 522–524 (1997).

[3] C. Vozzi, M. Nisoli, G. Sansone, S. Stagira, and S. De Silvestri, "Optimal spectral broadening in hollow-fiber compressor systems," *Applied Physics B* **80**, 285–289 (2005). DOI: 10.1007/s00340-004-1721-1.

[4] B. Schenkel, J. Biegert, U. Keller, C. Vozzi, M. Nisoli, G. Sansone, S. Stagira, S. De Silvestri, and O. Svelto, "Generation of 3.8-fs pulses from adaptive compression of a cascaded hollow fiber supercontinuum," *Optics Letters* **28**, 1987–1989 (2003). DOI: 10.1364/OL.28.001987.