Original software publication

GNLStools.py: A generalized nonlinear Schrödinger Python module implementing different models of input pulse quantum noise

Oliver Melchert *, Ayhan Demircan

Leibniz Universität Hannover, Institute of Quantum Optics (IQO), 30167 Hannover, Germany
Cluster of Excellence PhoenixD (Photonics, Optics, and Engineering – Innovation Across Disciplines), Hannover, Germany

ARTICLE INFO

ABSTRACT

We provide Python tools enabling numerical simulation and analysis of the propagation dynamics of ultrashort laser pulses in nonlinear waveguides. The modeling approach is based on the widely used generalized nonlinear Schrödinger equation for the pulse envelope. The presented software implements the effects of linear dispersion, pulse self-steepening, and the Raman effect. The focus lies on the implementation of input pulse shot noise, i.e. classical background fields that mimic quantum noise, which are often not thoroughly presented in the scientific literature. We discuss and implement commonly adopted quantum noise models based on pure spectral phase noise, as well as Gaussian noise. Coherence properties of the resulting spectra can be calculated. We demonstrate the functionality of the software by reproducing results for a supercontinuum generation process in a photonic crystal fiber, documented in the scientific literature. The presented Python tools are open-source and released under the MIT license in a publicly available software repository.

Code metadata

Current code version	1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00165
Legal Code License	MIT License
Code versioning system used	none
Software code languages, tools, and services used	Python, GitHub
Compilation requirements, operating environments & dependencies	The provided software requires Python, numpy and scipy. The provided examples need Pythons `matplotlib` for figure generation.
If available Link to developer documentation/manual	Documentation provided within code
Support email for questions	melchert@iqo.uni-hannover.de

1. Introduction

The propagation of laser pulses in nonlinear waveguides supports the generation of supercontinuum spectra [1–3]. Starting from a spectrally narrow input pulse, the interplay of linear and nonlinear effects induces tremendous spectral broadening, yielding flat spectra that can extend from the violet to the infrared [4]. Such effects can be achieved, e.g., in photonic crystal

fibers (PCFs) [5,6], wherein supercontinuum spectra can be produced using $\sim$ 100 fs-duration pulses, peak powers $\sim$ 10 kW and propagation lengths on the order of 1 m [4]. The resulting broad, flat spectra with high spectral density find application, e.g., in optical frequency metrology [7], and optical technologies [2].

A flexible theoretical framework for studying the complex physical processes associated with the generation of supercontinuum spectra is provided by the generalized nonlinear Schrödinger equation (GNLS) [1]. In order to model the propagation dynamics of laser pulses it combines the effects of linear dispersion, pulse self-steepening [8,9], and the Raman effect [10]. This accounts for various processes that support the generation of widely

---

* Corresponding author at: Leibniz Universität Hannover, Institute of Quantum Optics (IQO), 30167 Hannover, Germany.
E-mail addresses: melchert@iqo.uni-hannover.de (Oliver Melchert), demircan@iqo.uni-hannover.de (Ayhan Demircan).

extended supercontinuum spectra, such as the modulation instability [11], soliton-fission [12,13], and self-frequency shift of Raman solitons [10]. It furthers forms the basis for modeling optical rouge waves [14–17], and the interaction of solitons and dispersive waves in presence of an optical event-horizon [18–21]. The GNLS describes the real-valued optical field in terms of a complex-valued envelope at a fixed reference frequency using a nonlinear wave equation of first order in the propagation coordinate. It has proven to yield reliable results even in the single cycle regime [22]. Overall, the GNLS provides a classical description of the propagation dynamics of input pulses. Basic phenomena described by the GNLS are susceptible to the presence of noise. When attempting to model noise on the quantum level, a fundamental issue arises since the rigorous description of quantum noise sources requires the introduction of quantum mechanical operators [23–26], which are incompatible with classical fields. When the number of photons in the input pulse is large, however, semiclassical approaches exist that account for quantum fluctuations by randomizing physical interactions using a classical background field of low power, entering the GNLS via the initial condition. Such input pulse noise is modeled by simply adding to the coherent input pulse an incoherent, stochastic noise field. We consider commonly adopted one photon per frequency mode [3,27–30], and half a photon per temporal mode [31–33] noise models, and further implement a background field based on a classical analog of the zero-point field of quantum field theory (QFT) [34]. Including noise by perturbing the initial conditions is in accord with the Wigner method, an efficient computational approach for modeling quantum noise in optical fibers [25,26,35]. Considering an ensemble of independent simulation runs perturbed by noise, coherence properties of the simulated spectra can be investigated. Accounting for shot-to-shot fluctuations in numerical simulations appears essential for the dynamics and enables a better comparison to actual experiments, employing multishot measurement techniques [27,28].

In this article we present software tools allowing to reliably simulate and analyze basic phenomena described by the GNLS in presence of various types of input pulse shot noise. We detail the scientific problem addressed by the provided software in Section 2, and discuss its implementation, with emphasis on input pulse noise models, in Section 3. Section 4 reports verification tests, reproducing results for a supercontinuum generation process in a photonic crystal fiber, well documented in the scientific literature, where it has been used for illustrating nonlinear-optics effects [3], and for benchmarking algorithms [36–40]. We comment on impact and conclude in Section 5.

2. Scientific problem solved by the software

The provided Python tools enable simulation and analysis of the dynamics of ultrashort laser pulses, governed by the generalized nonlinear Schrödinger equation (GNLS) [1,3]

$$\partial_z u = i \sum_{n \geq 2} \frac{\beta_n}{n!} (i\partial_t)^n u + i\gamma \left(1 + \frac{i\partial_t}{\omega_0}\right)$$
$$\times \left[u(z,t) \int R(t') |u(z, t-t')|^2 \, dt' \right], \tag{1}$$

for a complex-valued field envelope $u \equiv u(z,t)$ on a periodic time-domain of extend T with boundary condition $u(z, -T/2) = u(z, T/2)$ and propagation distance z. An initial condition $u(z = 0, t) = u_0(t)$ needs to be specified. In Eq. (1), β_n (in units of $ps^n/\mu m$) is the nth order dispersion coefficient, γ ($W^{-1}/\mu m$) is a scalar nonlinear coefficient, and ω_0 (rad/ps) a reference frequency. t is a retarded time, measured in a reference frame moving with the group velocity at ω_0. The Raman effect is included via the total response function

$$R(t) = (1 - f_R)\delta(t) + f_R\, h_R(t), \tag{2}$$

where the first term results in an instantaneous Kerr-type nonlinear response, and where the Raman response function $h_R(t)$ enters with fractional contribution f_R [41,42]. A generic two-parameter model is

$$h_R(t) = \frac{\tau_1^2 + \tau_2^2}{\tau_1 \tau_2^2} \, e^{-t/\tau_2} \, \sin(t/\tau_1)\, \Theta(t), \tag{3}$$

with the Heaviside step function $\Theta(t)$ ensuring causality [41]. Based on a single damped harmonic-oscillator approximation of molecular responses, this simple model reproduces the Raman gain spectrum measured for silica glass reasonably well. For silica fibers, numerical values for the three relevant model parameters are $f_R = 0.18$, $\tau_1 = 12.2$ fs, and $\tau_2 = 32$ fs [41]. More complex response functions, based on improved response models and valid for other media, exist [43–45]. In this framework, using a discrete sequence of angular frequency detunings $\Omega \in \frac{2\pi}{T}\mathbb{Z}$ relative to ω_0, the expressions

$$F[u(z,t)] \equiv \frac{1}{T} \int_{-T/2}^{T/2} u(z,t) e^{i\Omega t} \, dt = u_\Omega(z), \tag{4a}$$

$$F^{-1}[u_\Omega(z)] \equiv \sum_\Omega u_\Omega(z) e^{-i\Omega t} = u(z,t), \tag{4b}$$

specify forward [Eq. (4a)], and inverse [Eq. (4b)] Fourier transforms, relating the field envelope $u(z,t)$ to the spectral envelope $u_\Omega(z)$. Using Parseval's identity for the transforms (4), the energy in both domains is

$$E(z) = \int_{-T/2}^{T/2} |u(z,t)|^2 \, dt = T \sum_\Omega |u_\Omega(z)|^2, \tag{5}$$

with instantaneous power $|u(z,t)|^2$ (W = J/s) and power spectrum $|u_\Omega(z)|^2$ (W). While Eq. (1) represents the GNLS in its time-domain formulation, a consistent formulation in the frequency domain is also possible [46].

Input pulse noise is modeled by simply adding to the coherent input pulse $u_0(t)$ an incoherent, stochastic noise field $\Delta u(t)$ with properties

$$\langle \Delta u(t) \rangle = 0, \tag{6a}$$

$$\langle \Delta u(t) \Delta u^*(0) \rangle = \sigma^2 \delta(t), \tag{6b}$$

where $\langle \ldots \rangle$ denotes an ensemble average over independent instances of noise, and σ^2 is the noise variance. According to Eq. (6a), at each point t, this background field has zero mean. The δ-function in the autocorrelation (6b) indicates that the noise varies fast in comparison to any reasonable field $u_0(t)$, and that subsequent actions of the noise are uncorrelated. For instance, considering *half a photon* with energy $\hbar\omega_0$ per temporal mode [31–33], the noise variance is $\sigma^2 = \hbar\omega_0/2$ with reduced Planck constant $\hbar$. Noise models for treating more specific pump laser spectra exist [47,48].

Let us note that there also exist formulations of the propagation dynamics in terms of stochastic equations, derived from a quantum theory of fibers with quantum noise sources taken into account via stochastic processes [49–51]. Technically, this amounts to adding multiplicative noise which contributes to the nonlinearity of the propagation equation [25]. More specifically, a stochastic variant of the GNLS, which extends Eq. (1) by spontaneous Raman scattering, is discussed in Ref. [52]. Recently, pulse propagation in presence of quantum noise has been generalized to waveguides with arbitrary frequency-dependent nonlinear profiles [53]. Such approaches are powerful for modeling and design of quantum devices based on nonlinear waveguides, for which quantum limits become important [53]. When aiming at describing the propagation dynamics of intense classical fields,

a phenomenological inclusion of quantum-limited shot noise via the above method is common practice [28].

In the frequency domain, shot-to-shot fluctuations of the field can then be characterized by the spectrally resolved modulus of first order coherence for zero time-lag

$$|g_{12}(\Omega)| = \left| \frac{\langle u_{\Omega,m} u_{\Omega,k}^* \rangle_{m \neq k}}{\sqrt{\langle |u_{\Omega,m}|^2 \rangle \langle |u_{\Omega,k}|^2 \rangle}} \right|, \tag{7}$$

where the angular brackets specify an average over nonidentical pairs of fields, labeled m and k, obtained from an ensemble of M simulation runs with independent noise fields [3,27,28]. At a given angular frequency detuning Ω, $0 \leq |g_{12}(\Omega)| \leq 1$, where $|g_{12}(\Omega)| \approx 1$ indicates good stability in amplitude and phase. Different from this measure of interpulse coherence, the modified intrapulse coherence

$$\tilde{\Gamma}(\Omega_1, \Omega_2) = \frac{|\langle u_{\Omega_1,m} u_{\Omega_2,m}^* \rangle|}{\langle |u_{\Omega_1,m} u_{\Omega_2,m}^*| \rangle} \tag{8}$$

allows to assess the coherence between different spectral components within a pulse. With focus on f-to-$2f$ (f = frequency) setups, the intrapulse coherence was shown to play an important role in carrier-envelope phase measurement and stabilization of ultrashort pulses [54]. Eq. (8) modifies the intrapulse coherence of Ref. [54] by relaxing the $f/2f$ condition. Instead, two general distinct frequencies Ω_1, and Ω_2 are considered.

3. Software description

GNLStools is written using the Python programming language [55], and depends on the functionality of numpy, scipy [56], and matplotlib [57]. It can be cloned directly from GitHub (see the link to the repository in the Code metadata), where it is available under the MIT license.

3.1. Software functionalities

The current version of GNLStools features:

- A data structure for the GNLS (1), which can be easily tailored to a specific nonlinear waveguide.
- Functions that implement optional input pulse noise models considering both, pure spectral phase noise as well as Gaussian noise.
- Functions for calculating the coherence properties of simulated spectra.

The software can be used on its own, it includes a basic driver script implementing a fixed stepsize "fourth-order Runge Kutta in the interaction picture" (RK4IP) solver [36] (Section 4.1), or as extension module for py-fmas [40] (Section 4.2). In the latter case, a user can take advantage of variable stepsize z-propagation algorithms and more elaborate Raman-response functions. The presented software is not overly feature-rich (Fig. 1), but allows to reliably simulate the complex physical processes that enable generation of supercontinuum spectra in nonlinear waveguides [3].

3.2. Implementation details

Solving the GNLS in the time-domain formulation Eq. (1) presents several disadvantages. Higher-order dispersion is difficult to handle. Even for the lowest order t-derivatives, introduction of finite-difference operations comes with unwanted truncation errors. It is therefore advantageous to perform some of the operations in the frequency-domain, allowing to employ spectral derivatives $\partial_t^n u = \mathsf{F}^{-1}[(-i\Omega)^n u_{\Omega}]$. The operating principles
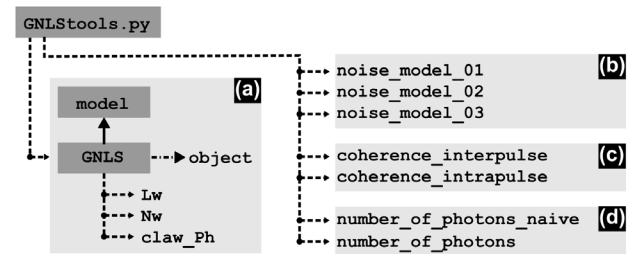

Fig. 1. Pictorial outline of the GNLStools module. (a) GNLS data structure, see Section 3.2.1. (b) Functions implementing the noise models, see Section 3.2.2. (c) Functions for computing coherence properties, see Section 3.2.3. (d) Functions for computing the number of photons within a field pulse (documentation provided within the code). Solid arrow indicates class instantiation ("creates" relationship), dashed arrows indicate functions and methods ("has-a" relationship), dash–dotted arrow indicates reference to an object ("is-a" relationship). The model in (a) integrates well with the propagation algorithms implemented in py-fmas [40], see Section 4.2.

of common algorithms for the solution of nonlinear Schrödinger-type equations, like the split-step Fourier method [58,59], the RK4IP method [36], and the variable step-size conservation quantity error method [37,38], exploit this. To facilitate implementation, the interval $-T/2 \ldots T/2$ is divided into N equal subintervals with temporal grid spacing $\Delta t = T/N$, yielding discrete grid points $t_m = m \Delta t$, and detuning grid points $\Omega_m = m \Delta\Omega$, $\Delta\Omega = 2\pi/T$, with $m = -N/2, \ldots, N/2 - 1$. Subsequently we write $u(z,t)|_{t=t_m} = u_m(z)$, and $u_{\Omega}(z)|_{\Omega=\Omega_m} = u_{\Omega_m}(z)$. Respecting the sign choice and normalization of the transforms (4), we use the numpy native discrete Fourier transform (DFT) routine ifft to implement Eq. (4a), and fft to implement Eq. (4b) [60].

3.2.1. The GNLS data structure

Instantiating an instance of the class GNLS requires a user to specify several input parameters. Below, they are listed as "parameter_name (data type): description":

- w (array, float): discrete angular frequencies $(\Omega_m)_{m=-N/2}^{N/2-1}$ in units of rad/fs;
- beta_n (array, float): ordered sequence of dispersion parameters $(\beta_n)_{n=2}^{N_{max}} = (\beta_2, \beta_3, \ldots, \beta_{N_{max}})$, with $N_{max} \geq 2$ and β_n in units of $\text{fs}^n/\mu\text{m}$;
- gamma (float): nonlinear parameter γ ($\text{W}^{-1}/\mu\text{m}$);
- w0 (float): reference angular frequency ω_0 (rad/fs);
- fR (float): Raman contribution f_R (default: 0.18);
- tau1 (float): time-scale τ_1 (fs) (default: 12.2 fs);
- tau2 (float): time-scale τ_2 (fs) (default: 32 fs).

While the angular frequency grid w is passed as positional argument, all other parameters are passed as keyword arguments. An example is discussed in Section 4.1. For using an instance of the GNLS class with a z-propagation algorithm, several instance methods are available. Below, they are listed in the format "method_name(arg1,arg2, . . .): description":

Lw(self): Using spectral derivatives, the frequency domain representation of the linear operator $L \equiv i \sum_{n \geq 2} \frac{\beta_n}{n!}(i\partial_t)^n$ on the right-hand side (rhs) of Eq. (1) can be written as

$$L_{\Omega} \equiv i \sum_{n=2}^{N_{max}} \frac{\beta_n}{n!} \Omega^n. \tag{9a}$$

For practical reasons, the sum has to be truncated at a finite integer number $N_{max} \geq 2$. This class method returns L_{Ω}, evaluated at the points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

Input parameters:

- The method relies on class attributes only.

Output parameters:

- Lw (array): L_Ω [Eq. (9a)] at the points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

The method is decorated as @property, so it can be conveniently used as gnls_instance.Lw.

Nw(self, uw): Using the convolution theorem for the transforms (4), the second term on the rhs of Eq. (1) can be written in a mixed representation as

$$N_\Omega(u) \equiv i\gamma(1 + \omega_0^{-1}\Omega)\, \mathsf{F}\left[(1-f_R)\mathcal{I}_1 + f_R\mathcal{I}_2\right], \qquad (9b)$$

where $\mathcal{I}_1 = |u|^2 u$, $\mathcal{I}_2 = \mathsf{F}^{-1}[\tilde{h}_R(\Omega)\mathsf{F}[|u|^2]]u$, and

$$\tilde{h}_R(\Omega) \equiv T\,\mathsf{F}[h_R(t)] = \frac{\tau_1^{-2} + \tau_2^{-2}}{\tau_1^{-2} - (\Omega + i\tau_2^{-1})^2}. \qquad (9c)$$

The method takes on input the spectral envelope u_Ω, retrieves $u = \mathsf{F}^{-1}[u_\Omega]$, and evaluates and returns Eq. (9b) at $(\Omega_m)_{m=-N/2}^{N/2-1}$.

Input parameters:

- uw (array): Spectral envelope u_{Ω_m} at the angular frequency grid points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

Output parameters:

- Nw (array): N_Ω [Eq. (9b)] at the points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

claw_Ph(self,i,zi,w,uw): Class method evaluating a conservation law of the GNLS (1), related to the classical analog of the photon number [41]. This method considers the energy [Eq. (5)] in the form $E = \hbar\sum_\Omega n_\Omega(\omega_0 + \Omega)$, where the dimensionless quantity $n_\Omega \equiv T|u_\Omega|^2/[\hbar(\omega_0+\Omega)]$ specifies the number of photons with energy $\hbar(\omega_0+\Omega)$. The total number of photons is then given by

$$C_{\mathrm{Ph}}(z) \equiv \sum_\Omega n_\Omega = \frac{2\pi}{\hbar\Delta\Omega}\sum_\Omega \frac{|u_\Omega(z)|^2}{\omega_0 + \Omega}. \qquad (9d)$$

In a rigorous quantum mechanical treatment, the photon number is instead defined using photon creation and annihilation operators [24]. Let us note that while the GNLS (1) conserves the total number of photons Eq. (9d), it does not conserve the pulse energy [Eq. (5)] [41].

Input parameters:

- i (int): Integer label of the current z-propagation step.
- zi (float): Current propagation distance.
- w (array): Angular frequency grid $(\Omega_m)_{m=-N/2}^{N/2-1}$.
- uw (array): Spectral envelope u_{Ω_m} at the angular frequency grid points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

Output parameters:

- Cph (float): Total number of photons C_{Ph} at the current point along z [Eq. (9d)].

The method has the structure of an user–action function for use with py-fmas [40] (Section 4.2).

3.2.2. Implemented noise models

We provide optional functions for generating time-domain representations of input pulse noise, consistent with Eqs. (6). Below, they are listed in the format "function_name(arg1,arg2, . . .): description":

noise_model_01(t,w0,s0): Function generating an instance of noise by directly sampling in the time-domain. The underlying noise model assumes normally distributed amplitudes.

Input parameters:

- t (array): Time-grid $(t_m)_{m=-N/2}^{N/2-1}$.
- w0 (float): Pulse center frequency ω_0.
- s0 (int): Integer seed s_0 for pseudo random number generator.

Output parameters:

- du_t (array): Instance $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of time-domain noise.

An instance of this type of noise is obtained by directly sampling a sequence $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of complex-valued noise-amplitudes in the time domain as

$$\Delta u_m = \sqrt{\frac{\hbar\omega_0}{4\Delta t}}\,(X + iY), \qquad (10a)$$

where the independent random variables X and Y yield independent identically distributed (iid) standard normal random numbers [i.e. $X, Y \sim \mathcal{N}(0, 1)$]. Performing an ensemble average, this type of noise exhibits the properties

$$\langle\Delta u_m\rangle = 0, \qquad (10b)$$

$$\langle\Delta u_m\Delta u_0^*\rangle = \frac{\hbar\omega_0}{2\Delta t}\,\delta_{m0}, \qquad (10c)$$

i.e. the average field is zero and the autocorrelation (10c) is equivalent to Eq. (6b) on a discrete grid with grid spacing Δt [31]. An instance of this noise has average energy $\langle\sum_m |u_m|^2\Delta t\rangle = N\hbar\omega_0/2$, corresponding, on average, to *half a photon* with energy $\hbar\omega_0$ per temporal mode.

noise_model_02(t,w0,s0): Function generating an instance of time-domain noise by sampling its Fourier representation. The underlying noise model assumes pure phase noise.

Input parameters:

- t (array): Time-grid $(t_m)_{m=-N/2}^{N/2-1}$.
- w0 (float): Pulse center frequency ω_0.
- s0 (int): Integer seed s_0 for pseudo random number generator.

Output parameters:

- du_t (array): Instance $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of time-domain noise.

An instance of this type of noise is obtained using a Fourier method by first sampling a sequence $(\Delta u_{\Omega_m})_{m=-N/2}^{N/2-1}$ of random complex-valued spectral amplitudes

$$\Delta u_{\Omega_m} \equiv \sqrt{\frac{\hbar(\omega_0+\Omega_m)}{T}}\,e^{-i\Phi}, \qquad (11)$$

where the random variable Φ yields iid phase-angles uniformly distributed in the range $0 \ldots 2\pi$ [i.e. $\Phi \sim U(0, 2\pi)$]; an inverse Fourier transform, consistent with Eq. (4b), is then used to obtain the sequence $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of time-domain noise amplitudes.

For this type of noise, the magnitude of the spectral amplitudes (11) is definite and the energy per mode Ω is $T|u_\Omega|^2 = \hbar(\omega_0 + \Omega)$. With the number of photons n_Ω as per Eq. (9d), adding noise as $u_\Omega \to u_\Omega + \Delta u_\Omega$ results in $n_\Omega \to n_\Omega + 1$. The photon occupation number is thus increased by the minimal definite amount of *one (entire) photon* per mode. The phase

of this photon is, however, entirely indefinite and the noise in different modes has no particular phase relationship. An instance of the noise has total energy $T \sum_{\Omega} |u_{\Omega}|^2 = N\hbar\omega_0$.

`noise_model_03(t,w0,s0)`: Function generating an instance of time-domain noise by sampling its Fourier representation. The underlying noise model assumes normally distributed spectral amplitudes.

Input parameters:

- `t` (array): Time-grid $(t_m)_{m=-N/2}^{N/2-1}$.
- `w0` (float): Pulse center frequency ω_0.
- `s0` (int): Integer seed s_0 for pseudo random number generator.

Output parameters:

- `du_t` (array): Instance $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of time-domain noise.

An instance of this type of noise is obtained using a Fourier method by first sampling a sequence $(\Delta u_{\Omega_m})_{m=-N/2}^{N/2-1}$ of random complex-valued spectral amplitudes

$$\Delta u_{\Omega_m} \equiv \sqrt{\frac{\hbar(\omega_0 + \Omega_m)}{T}} \sqrt{I} e^{-i\Phi}, \qquad (12)$$

where the random variable I obeys an exponential distribution with rate parameter 2 [i.e. $I \sim \mathrm{Exp}(2)$ with expectation value $\langle I \rangle = 1/2$], and Φ is uniformly distributed in $0 \ldots 2\pi$ [i.e. $\Phi \sim U(0, 2\pi)$]; an inverse Fourier transform, consistent with Eq. (4b), is then used to obtain the sequence $(\Delta u_m)_{m=-N/2}^{N/2-1}$ of time-domain noise amplitudes.

Averaging over the noise in a mode Ω yields energy $\langle T|\Delta u_{\Omega}|^2 \rangle = \hbar(\omega_0 + \Omega)\langle I \rangle = \hbar(\omega_0 + \Omega)/2$, corresponding, on average, to *half a photon* per mode. In contrast to the pure phase noise model implemented by `noise_model_02`, this type of noise exhibits instance to instance fluctuations of the energy in each mode. Thereby, the random variable I takes the role of an occupation number with support $I \in [0, \infty)$. On average, an instance of this noise has energy $\langle T \sum_{\Omega} |u_{\Omega}|^2 \rangle = N\hbar\omega_0/2$. Alternatively, using the Box–Müller transform [61], we may rewrite $\sqrt{I} e^{-i\Phi}$, with $I \sim \mathrm{Exp}(2)$, $\Phi \sim U(0, 2\pi)$, in Eq. (12) as $(X + iY)/2$, with $X, Y \sim \mathcal{N}(0, 1)$ two iid standard normal random variables. This noise model is based on a classical analog of the zero-point field of QFT, which exactly reproduces the statistics of the electromagnetic vacuum [34].

Fourier methods that obtain time-domain noise by sampling frequency dependent spectral noise amplitudes generally result in correlations $\langle \Delta u_m \Delta u_0^* \rangle \sim |t_m|^{-1}$. This is demonstrated in Fig. 2, where the ensemble averaged field and autocorrelation for the three noise models are shown. As evident from the inset in Fig. 2(b), both Fourier methods result in correlations that persist over many grid spacings Δt.

Let us note that the above noise models differ with respect to details of implementation and interpretation. Nevertheless, with some degree of approximation, they can easily be related to each other. For instance, under the assumption that the required bandwidth of the computational domain is small compared to the pulse center frequency, i.e. $N\Delta\Omega \ll \omega_0$, we may approximate Eq. (11) by $\Delta u_{\Omega_m} \approx \sqrt{\hbar\omega_0/T} \exp(-\Phi)$. All terms in the inverse Fourier transform [Eq. (4b)] of these noise spectral amplitudes are then identically distributed. Approximating these sums using the central limit theorem yields Eq. (10a) times a factor of $\sqrt{2}$ (this is a consequence of one model assuming one entire photon per mode, while the other assumes half-a photon per mode). Let us note that this approximation is accompanied by difficulties:
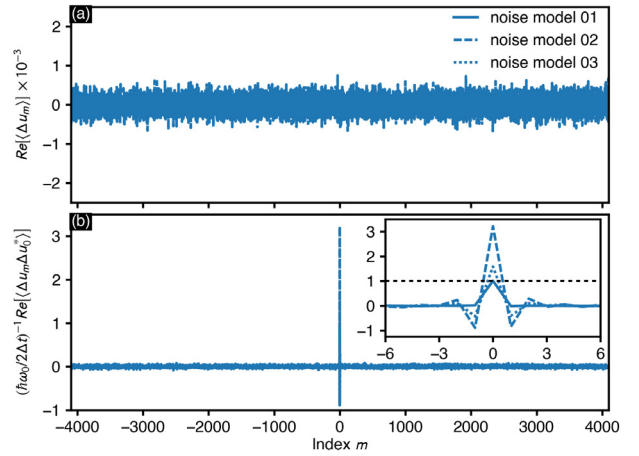

Fig. 2. Ensemble averaged noise moments. (a) Real part of the time-domain noise amplitudes as function of the grid point index $m = t_m/\Delta t$. (b) Scaled autocorrelation. The inset shows a close up view of the index range $-6 \ldots 6$. Noise is sampled on a grid with temporal extent $T = 4$ (ps), $N = 2^{13}$ grid points, and for $\omega_0 = 2.2559$ rad/fs. Averages are performed over 10^4 independent instances of noise.

it corresponds to *one (entire) photon* per mode if the energy (5) is written as $E = \hbar\omega_0 \sum_{\Omega} n'_{\Omega}$, with $n'_{\Omega} \equiv T|u_{\Omega}|^2/(\hbar\omega_0)$. Thus, irrespective of Ω, all photons are assumed to contribute the same energy $\hbar\omega_0$. Put into the context of pulse propagation models, this is a feature of standard nonlinear Schrödinger-type equations for which energy conservation and photon number conservation are trivially linked. This, however, is at odds with the GNLS (1) which conserves the number of photons but not the energy [41]. Under the same assumption and approximation, Eq. (12) can easily be related to Eq. (10a): applying the inverse Fourier transform, we may rewrite the individual terms using the Box–Müller transform [61] to directly obtain Eq. (10a). Moreover, in terms of a Fourier method and in the above approximation, practically any iid spectral amplitudes will yield time-domain noise with properties Eqs. (6) [62]. As a technical detail, let us note that when using a Fourier method to set up noise, the noise spectral amplitudes also depend on the normalization of the transform pair (4). For instance, in Ref. [30], a GNLS on the infinite t-domain with a different transform pair and, consequently, spectral noise amplitudes with different normalization, was considered.

3.2.3. Functions for calculating coherence properties

We provide optional functions for calculating the coherence properties of spectra obtained from pulse propagation simulations in terms of the GNLS (1). Below, they are listed in the format "function_name(arg1,arg2, . . .): description":

`coherence_interpulse(w,uw_list)`: Function computing the interpulse coherence Eq. (7).

Input parameters:

- `w` (array): Angular-frequency grid $(\Omega_m)_{m=-N/2}^{N/2-1}$.
- `uw_list` (array): List [`uw1`, `uw2`, `...`], comprising independent spectra `uw1`, `uw2`, `...`, obtained for the same propagation distance z but for different instances of input pulse noise.

Output parameters:

- `g12` (array): $g_{12}(\Omega)$ [Eq. (7)] at the points $(\Omega_m)_{m=-N/2}^{N/2-1}$.

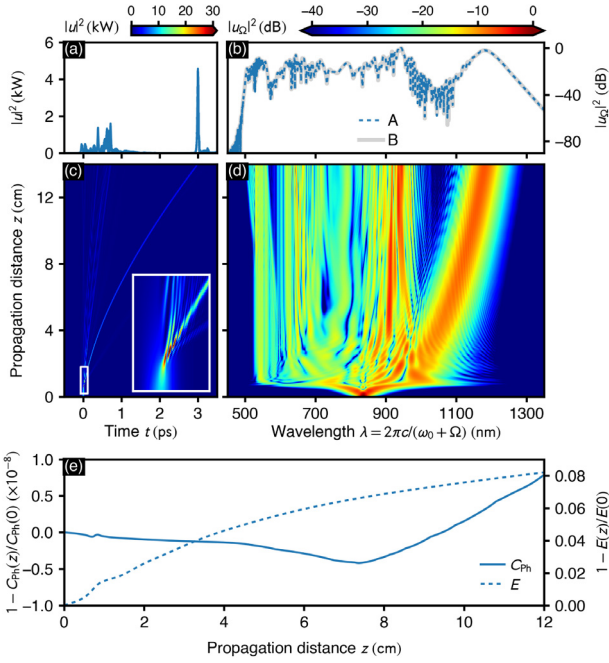`coherence_intrapulse(w,uw_list,w1,w2)`: Function computing the intrapulse coherence Eq. (8).

Fig. 3. Supercontinuum generation in a photonic crystal fiber. (a) Instantaneous power, and, (b) spectrum after 14 cm of propagation. In (b), result of the presented implementation (labeled A) is compared to pyNLO [63] (labeled B). (c) Propagation of the instantaneous power, and, (d) spectrum as function of propagation distance. White boxes in (c) show a close-up view of the soliton fission process. (d) Normalized deviation of photon number (C_{Ph}) and energy (E) along the fiber.

Input parameters:

- w (array): Angular-frequency grid $(\Omega_m)_{m=-N/2}^{N/2-1}$.
- uw_list (array): List [uw1, uw2, ...], comprising independent spectra uw1, uw2, ..., obtained for the same propagation distance z but for different instances of input pulse noise.
- w1 (float): Reference angular frequency Ω_1.
- w2 (float): Reference angular frequency Ω_2.

Output parameters:

- G (float): $\tilde{\Gamma}^{CEP}(\Omega_1, \Omega_2)$ [Eq. (8)].

4. Sample results

As a verification test of the presented Python tools, we consider a supercontinuum generation process in a photonic crystal fiber, detailed in Ref. [3]. Specifically, we perform simulations in terms of the GNLS (1), using the sequence of dispersion coefficients $\beta_2 = -1.183 \times 10^{-2}$ fs^2/μm, $\beta_3 = 8.10383 \times 10^{-2}$ fs^3/μm, $\beta_4 = -9.5205 \times 10^{-2}$ fs^4/μm, $\beta_5 = 0.20737$ fs^5/μm, $\beta_6 = -0.53943$ fs^6/μm, $\beta_7 = 1.3486$ fs^7/μm, $\beta_8 = -2.5495$ fs^8/μm, $\beta_9 = 3.0524$ fs^9/μm, $\beta_{10} = -1.7140$ fs^{10}/μm, and $\gamma = 0.11 \times 10^{-6}$ W^{-1}/μm. For the Raman response we use the standard values for silica fibers $f_R = 0.18$, $\tau_1 = 12.2$ fs, and $\tau_2 = 32$ fs [41]. As initial condition we take a hyperbolic-secant pulse $u_0(t) = \sqrt{P_0}\,\mathrm{sech}(t/t_0)$, with duration $t_0 = 28.4$ fs, peak power $P_0 = 10$ kW, and pump wavelength $\lambda_0 = 835$ nm corresponding to $\omega_0 = 2\pi c/\lambda_0 \approx 2.2559$ rad/fs (with speed of light $c = 0.29979$ μm/fs). The number of photons [Eq. (9d)] in this pulse

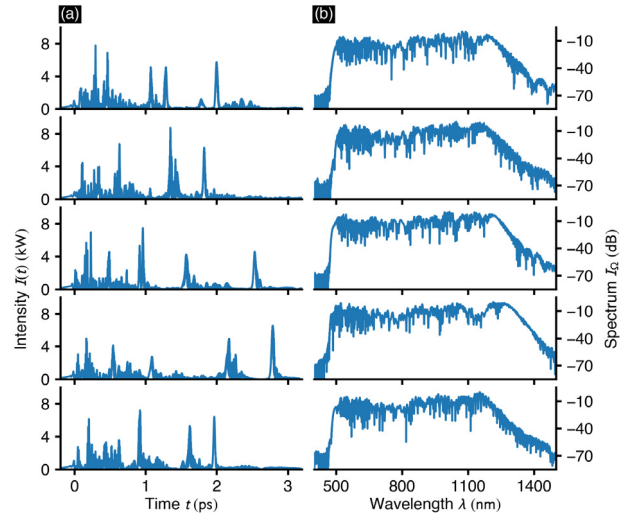**Fig. 4.** Shot-to-shot fluctuations of (a) instantaneous power, and, (b) corresponding spectra, for input pulses with $t_0 = 85.0$ fs duration input pulses and 10 cm of propagation (cf. Fig. 18 of Ref. [3]).

amounts to $C_{Ph} \approx 2.4 \times 10^9$. The z-propagation dynamics of the above propagation scenario without noise background, followed over 14 cm, is shown in Figs. 3(a–d).

4.1. A minimal working example

To demonstrate how to initialize and use the GNLS data structure for performing pulse propagation in terms of Eq. (1), we provide a minimal example in code-listing 1. The provided code initializes the GNLS data structure in lines 10–28, sets up the input pulse in line 30, retrieves an instance of input pulse noise in line 31, and propagates the initial field for 10 cm using the fourth-order Runge Kutta in the interaction picture (RK4IP) method [36]. This type of algorithmic approach is also referred to as integrating factor method [64], or linearly exact Runge Kutta method [65]. Pulse propagation is performed in lines 35–42 using a fixed step-size of 10 μm, and using a temporal domain of extent $T = 7$ ps with $N = 2^{13}$ grid points. The script terminates in just under 15 s (Apple M1 chip @ 3.2 GHz) and reproduces Figs. 3(a,b) [cf. Figs. 18(a,b) of Ref. [3], Fig. 2(a) of Ref. [36], Fig. 1(a) of Ref. [38], and Fig. 2(b,c) of Ref. [40]]. The script shown in listing 1 is located in project folder numExp01 provided with the code [66].

4.2. Using GNLS with py-fmas

Integration with py-fmas. Fig. 3 is produced by using the GNLS data structure in conjunction with py-fmas [40], a Python package for the numerical simulation of the z-propagation dynamics of ultrashort optical pulses in terms of the analytic signal of the optical field. GNLStools can be used as an elaboration module, providing an envelope based model which integrates well with the propagation algorithms provided by py-fmas. Specifically, pulse propagation in Figs. 3–5 is performed by the variable step-size conservation quantity error method [37,38,40], using Eq. (9d) to guide step-size selection. Fig. 3(e) shows the numerical error accumulated for the photon number [Eq. (9d)] and energy [Eq. (5)], showing that C_{Ph} is conserved up to order 10^{-8}, while E is not conserved [41]. Considering C_{Ph}, the feature at $z \approx 1$ cm indicates the onset of soliton-fission (close-up view in Fig. 3c),
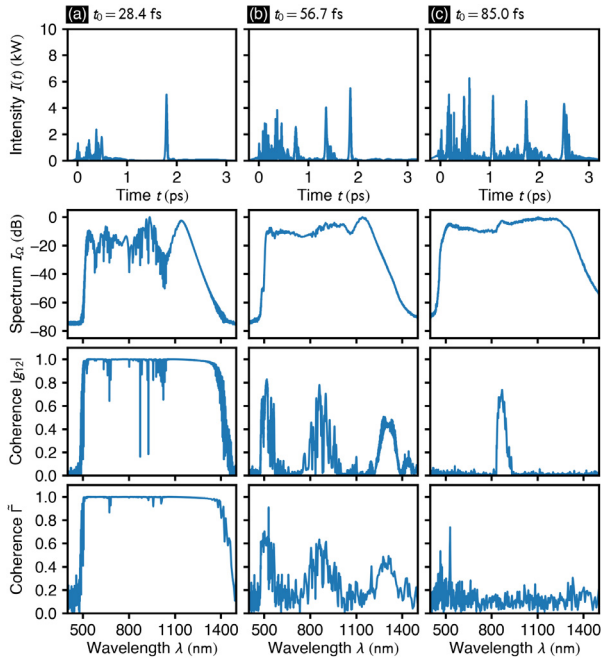
Fig. 5. Coherence properties of supercontinuum spectra after 10 cm of propagation (cf. Fig. 19 of Ref. [3]). (a) from top to bottom: exemplary single-shot output intensity, ensemble averaged spectrum, and degrees of coherence $|g_{12}|$ [Eq. (7)], and $\bar{\Gamma}$ [Eq. (8)], as function of wavelength $\lambda = 2\pi c/(\omega_0 + \Omega)$. For the calculation of the modified intrapulse coherence $\bar{\Gamma}(\Omega_1, \Omega_2)$, $\Omega_1 = \Omega$, and $\Omega_2 = -0.96$ rad/fs, corresponding to $\lambda_1 = \lambda$, and $\lambda_2 = 1.45$ nm. Duration of the input pulses is $t_0 = 28.4$ fs. (b) same for $t_0 = 56.7$ fs, and (c) same for $t_0 = 85.0$ fs.

and the change of the trend at $z \approx 7$ cm indicates the onset of the interaction of a soliton and a dispersive wave which mainly determines the interaction dynamics and generation of supercontinuum spectra. A small project workflow with a driver script that performs the simulation, and a postprocessing script that generates Fig. 3, are located in project folder `numExp02` provided with the code [66].

Shot-to-shot fluctuations. Fig. 4 demonstrates shot-to-shot variations in pulse intensity and spectrum after 10 cm of propagation, arising from the inclusion of input pulse noise for an input pulse with duration $t_0 = 85.1$ fs. The noise is generated via noise model 1, i.e. by direct sampling in the time-domain. A small project workflow with a driver script that performs the simulation, and a postprocessing script that generates Fig. 4, are located in project folder `numExp03_noise_model_01`, provided with the code [66].

Coherence properties. The coherence properties of input pulses of different duration, obtained by performing ensemble averages over 200 independent simulation runs with different noise seeds, are shown in Fig. 5. Fig. 5 reproduces Fig. 19 of Ref. [3], wherein a detailed discussion of the effects of input pulse noise and the interpretation of the coherence can be found. In addition, in Fig. 5 we also show the modified intrapulse coherence [Eq. (8)] discussed in Ref. [54]. A postprocessing script that generates Fig. 5 is located in folder `numExp03_noise_model_01` provided with the code [66]. From our experience, the slight correlations introduced by the Fourier method based noise models, see Section 3.2.2 and

Fig. 2, do not affect the coherence properties of the obtained supercontinuum spectra.

5. Impact and conclusions

The presented `GNLStools` comprise a data structure implementing the generalized nonlinear Schrödinger equation, two commonly adopted models of quantum noise and a further noise model based on a classical analog of the zero-point field of QFT, and functions for assessing the coherence properties of simulated spectra. It provides all features required for studying basic phenomena supported by the GNLS in the presence of input pulse shot noise. The provided software can be used on its own, as demonstrated in Section 4.1, or as elaboration module for use with the propagation algorithms provided by the `py-fmas` package [40], as demonstrated in Section 4.2.

With the provided software we hope to shed some more light on the implementation of quantum noise models for use in pulse propagation studies, an issue that is often not thoroughly presented in the scientific literature. We hope to directly benefit students and researchers alike, which are new to the field of nonlinear optics and seek a tutorial-type introduction on how to perform pulse propagation simulations including quantum noise, and assess the coherence properties of the resulting spectra. The minimal example in Section 4.1 can very well serve as classroom code or as starting point for seminar projects in computation oriented courses, aimed at demonstrating algorithmic approaches that go beyond simple split-step Fourier methods commonly used for solving nonlinear Schrödinger-type equations [1,58,59].

Extending Eq. (1) to a stochastic GNLS which includes spontaneous Raman scattering can be accomplished using the procedure described in Refs. [25,52]. Using a simple split-step Fourier method, which uses the GNLS data structure detailed in Section 3.2.1, and accounts for the additional multiplicative noise during the update associated with the nonlinear part of the propagation equation, seems the most basic way to include the necessary changes with the provided software.

Finally, we would like to reference original research applying the presented GNLS tools. In Ref. [21] we used the presented software to demonstrate an efficient all-optical switching scheme, based on controlling the features of soliton fission induced supercontinuum spectra using a time-delayed dispersive wave.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data/Code is available under the link to the repository provided in the Code metadata.

Acknowledgments

Listing 1: Python code for solving the GNLS (1) using the fourth-order Runge Kutta in the interaction picture method [36].

```python
import numpy as np
import matplotlib.pyplot as plt
from GNLStools import GNLS, noise_model_01

# -- SET COMPUTATIONAL GRID
z, dz = np.linspace(0, 0.1e6, 10000, retstep=True)
t = np.linspace(-3500, 3500, 2**13, endpoint=False)
w = np.fft.fftfreq(t.size, d=t[1]-t[0])*2*np.pi
# -- INSTANTIATE GENERALIZED NONLINEAR SCHROEDINGER EQUATION
gnls = GNLS(
    w,                  # (rad/fs)
    beta_n = [
        -1.1830e-2,     # (fs^2/micron) beta_2
        8.1038e-2,      # (fs^3/micron) beta_3
        -0.95205e-1,    # (fs^4/micron) beta_4
        2.0737e-1,      # (fs^5/micron) beta_5
        -5.3943e-1,     # (fs^6/micron) beta_6
        1.3486,         # (fs^7/micron) beta_7
        -2.5495,        # (fs^8/micron) beta_8
        3.0524,         # (fs^9/micron) beta_9
        -1.7140,        # (fs^10/micron) beta_10
        ],
    gamma=0.11e-6,      # (1/W/micron)
    w0= 2.2559,         # (rad/fs)
    fR = 0.18,          # (-)
    tau1 = 12.2,        # (fs)
    tau2 = 32.0         # (fs)
    )
# -- SPECIFY INITIAL PULSE
ut = np.sqrt(1e4)/np.cosh(t/28.4)
dut = noise_model_01(t, 2.2559, 1)
uw = np.fft.ifft(ut + dut)

# -- RK4IP PULSE PROPAGATION
P = np.exp(gnls.Lw*dz/2)
for n in range(1,z.size):
    uw_I = P*uw
    k1 = P*gnls.Nw(uw)*dz
    k2 = gnls.Nw(uw_I + k1/2)*dz
    k3 = gnls.Nw(uw_I + k2/2)*dz
    k4 = gnls.Nw(P*uw_I + k3)*dz
    uw = P*(uw_I + k1/6 + k2/3 + k3/3) + k4/6

# -- PLOT RESULTS
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8, 3))
I = np.abs(np.fft.fft(uw))**2
ax1.plot(t, I*1e-3)
ax1.set_xlim(-200,3200); ax1.set_xlabel(r"Time $t$ (fs)")
ax1.set_ylim(0,6); ax1.set_ylabel(r"Intensity $|u|^2$ (kW)")
Iw = np.abs(uw)**2
ax2.plot(2*np.pi*0.29979/(w+2.2559), 10*np.log10(Iw/np.max(Iw)))
ax2.set_xlim(0.45,1.4); ax2.set_xlabel(r"Wavelength $\lambda$ (micron)")
ax2.set_ylim(-60,0); ax2.set_ylabel(r"Spectrum $|u_\lambda|^2$ (dB)")
fig.tight_layout(); plt.show()
```

References

[1] Agrawal GP. Nonlinear fiber optics. Academic Press; 2019.

[2] Mitschke F. Fiber optics: physics and technology. Springer; 2016.

[3] Dudley JM, Genty G, Coen S. Supercontinuum generation in photonic crystal fiber. Rev Modern Phys 2009;78:1135.

[4] Ranka JK, Windeler RS, Stentz AJ. Visible continuum generation in air–silica microstructure optical fibers with anomalous dispersion at 800 nm. Opt Lett 2000;25:25.

[5] Knight JC. Photonic crystal fibres. Nature 2003;424:847.

[6] Birks TA, Knight JC, Russell PSJ. Endlessly single-mode photonic crystal fiber. Opt Lett 1997;22:961.

[7] Jones DJ, Diddams SA, Ranka JK, Stentz A, Windeler RS, Hall JL, et al. Carrier-Envelope Phase Control of Femtosecond Mode-Locked Lasers and Direct Optical Frequency Synthesis. Science 2000;288:635.

[8] DeMartini F, Townes CH, Gustafson TK, Kelley PL. Self-Steepening of Light Pulses. Phys Rev 1967;164:312.

[9] de Oliveira JR, de Moura MA, Hickmann JM, Gomes ASL. Self-steepening of optical pulses in dispersive media. J Opt Soc Amer B 1992;9:2025.

[10] Gordon JP. Theory of the soliton self-frequency shift. Opt Lett 1986;11:662.

[11] Demircan A, Bandelow U. Supercontinuum generation by the modulation instability. Opt Commun 2005;244:181–5.

[12] Husakou AV, Herrmann J. Supercontinuum Generation of Higher-Order Solitons by Fission in Photonic Crystal Fibers. Phys Rev Lett 2001;87:203901.

[13] Demircan A, Bandelow U. Analysis of the interplay between soliton fission and modulation instability in supercontinuum generation. Appl Phys B 2007;86:31.

[14] Solli DR, Ropers C, Koonath P, Jalali B. Optical rouge waves. Nature 2007;450:1054–7.

[15] Solli DR, Ropers C, Jalali B. Active Control of Rogue Waves for Stimulated Supercontinuum Generation. Phys Rev Lett 2008;101:233902.

[16] Demircan A, Amiranashvili S, Brée C, Mahnke C, Mitschke F, Steinmeyer G. Rogue events in the group velocity horizon. Sci Rep 2012;2:850.

[17] Dudley JM, Dias F, Erkintalo M, Genty G. Instabilities, breathers and Rogue waves in optics. Nat Photonics 2014;8:755–64.

[18] Demircan A, Amiranashvili S, Steinmeyer G. Controlling Light by Light with an Optical Event Horizon. Phys Rev Lett 2011;106:163901.

[19] Demircan A, Amiranashvili S, Brée C, Morgner U, Steinmeyer G. Supercontinuum generation by multiple scatterings at a group velocity horizon. Opt Express 2014;22:3866–79.

[20] Skryabin DV, Gorbach AV. Colloquium: Looking at a soliton through the prism of optical supercontinuum. Rev Modern Phys 2010;82:1287–99.

[21] Melchert O, Brée C, Tajalli A, Pape A, Arkhipov R, Willms S, et al. All-optical supercontinuum switching. Commun Phys 2020;3:146.

[22] Brabec T, Krausz F. Nonlinear Optical Pulse Propagation in the Single-Cycle Regime. Phys Rev Lett 1997;78:3282.

[23] Haus HA, Lai Y. Quantum theory of soliton squeezing: A linearized approach. J Opt Soc Amer B 1990;7:386.

[24] Henry CH, Kazarinov RF. Quantum noise in photonics. Rev Modern Phys 1996;68:801.

[25] Drummond PD, Corney JF. Quantum noise in optical fibers. I. Stochastic equations. J Opt Soc Amer B 2001;18(2):139.

[26] Corney JF, Drummond PD. Quantum noise in optical fibers. II. Raman jitter in soliton communications. J Opt Soc Amer B 2001;18:153.

[27] Dudley JM, Coen S. Numerical Simulations and Coherence Properties of Supercontinuum Generation in Photonic Crystal and Tapered Optical Fibers. IEEE J Sel Top Quantum Electron 2002;8:651.

[28] Dudley JM, Coen S. Coherence properties of supercontinuum spectra generated in photonic crystal and tapered optical fibers. Opt Lett 2002;27:1180.

[29] Kobtsev SM, Smirnov SV. Modelling of high-power supercontinuum generation in highly nonlinear, dispersion shifted fibers at CW pump. Opt Express 2005;13:6912–8.

[30] Genier E, Bowen P, Sylvestre T, Dudley JM, Moselund P, Bang O. Amplitude noise and coherence degradation of femtosecond supercontinuum generation in all-normal-dispersion fibers. J Opt Soc Amer B 2019;36:A161–7.

[31] Paschotta R. Noise of mode-locked lasers (Part I): Numerical model. Appl Phys B 2004;79:153.

[32] Ruehl A, Martin MJ, Cossel KC, Chen L, McKay H, Thomas B, et al. Ultrabroadband coherent supercontinuum frequency comb. Phys Rev A 2011;84:011806.

[33] Zhou B, Bache M. Invited Article: Multiple-octave spanning high-energy mid-IR supercontinuum generation in bulk quadratic nonlinear crystals. APL Photonics 2016;1:050802.

[34] Ibison M, Haisch B. Quantum and classical statistics of the electromagnetic zero-point field. Phys Rev A 1996;54:2737–44.

[35] Werner MJ, Raymer MG, Beck M, Drummond PD. Ultrashort pulsed squeezing by optical parametric amplification. Phys Rev A 1995;52:4202.

[36] Hult J. A Fourth-Order Runge–Kutta in the Interaction Picture Method for Simulating Supercontinuum Generation in Optical Fibers. J Lightwave Tech 2007;25(12):3770.

[37] Heidt AM. Efficient Adaptive Step Size Method for the Simulation of Supercontinuum Generation in Optical Fibers. J Lightwave Tech 2009;27(18):3984.

[38] Rieznik AA, Heidt AM, Konig PG, Bettachini VA, Grosz DF. Optimum integration procedures for supercontinuum simulation. IEEE Photonics J 2012;4:552–60.

[39] Balac S, Fernandez A. SPIP: A computer program implementing the Interaction Picture method for simulation of light-wave propagation in optical fibre. Comput Phys Comm 2016;199:139.

[40] Melchert O, Demircan A. py-fmas: A python package for ultrashort optical pulse propagation in terms of forward models for the analytic signal. Comput Phys Comm 2022;273:108257.

[41] Blow KJ, Wood D. Theoretical Description of Transient Stimulated Raman Scattering in Optical Fibers. IEEE J Quantum Electron 1989;25:2665.

[42] Stolen RH, Gordon JP, Tomlinson WJ. Raman response function of silica-core fibers. J Opt Soc Amer B 1989;6:1159.

[43] Lin Q, Agrawal GP. Raman response function for silica fibers. Opt Lett 2006;31:3086.

[44] Hollenbeck D, Cantrell CD. Multiple-vibrational-mode model for fiber-optic Raman gain spectrum and response function. J Opt Soc Amer B 2002;19:2886.

[45] Agger C, Petersen C, Dupont S, Steffensen H, Lyngsø JK, Thomsen CL, et al. Supercontinuum generation in ZBLAN fibers—detailed comparison between measurement and simulation. J Opt Soc Amer B 2012;29:635.

[46] François PL. Nonlinear propagation of ultrashort pulses in optical fibers: Total field formulation in the frequency domain. J Opt Soc Amer B 1991;8:276–93.

[47] Frosz MH, Bang O, Bjarklev A. Soliton collision and Raman gain regimes in continuous-wave pumped supercontinuum generation. Opt Express 2006;14:9391.

[48] Frosz MH. Validation of input-noise model for simulations of supercontinuum generation and Rogue waves. Opt Express 2010;18:14778.

[49] Carter SJ, Drummond PD, Reid MD, Shelby RM. Squeezing of quantum solitons. Phys Rev Lett 1987;58:1841–4.

[50] Drummond PD, Carter SJ. Quantum-field theory of squeezing in solitons. J Opt Soc Amer B 1987;4(10):1565–73.

[51] Brainis E, Amans D, Massar S. Scalar and vector modulation instabilities induced by vacuum fluctuations in fibers: Numerical study. Phys Rev A 2005;71:023808.

[52] Corwin KL, Newbury NR, Dudley JM, Coen S, Diddams SA, Weber K, et al. Fundamental Noise Limitations to Supercontinuum Generation in Microstructure Fiber. Phys Rev Lett 2003;90:113904.

[53] Bonetti J, Grosz DF, Hernandez SM. Quantum Noise in Fibers with Arbitrary Nonlinear Profiles. Phys Rev Lett 2021;126:213602.

[54] Raabe N, Feng T, Witting T, Demircan A, Brée C, Steinmeyer G. Role of Intrapulse Coherence in Carrier-Envelope Phase Stabilization. Phys Rev Lett 2017;119:123901.

[55] Rossum G. Python reference manual. Tech. rep., Amsterdam, The Netherlands, The Netherlands: CWI (Centre for Mathematics and Computer Science); 1995.

[56] Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 2020;17:261.

[57] Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng 2007;9:90.

[58] Taha T, Ablowitz M. Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations. II. Numerical, Nonlinear Schrödinger Equation. J Comput Phys 1984;55:203.

[59] DeVries P. Application of the Split Operator Fourier Transform method to the solution of the nonlinear Schrödinger equation. AIP Conf Proc 1987;160:269.

[60] On the discrete computational grid we identify F with ifft, and F^{-1} with fft. Consequently, the spectral envelope in the provided implementation differs from the one defined by Eq. (4a) by an unimportant phase factor.

[61] Box GEP, Muller ME. A Note on the Generation of Random Normal Deviates. Ann Math Stat 1958;29(2):610–1.

[62] Goedecke GH. Stochastic Electrodynamics. I. On the Stochastic Zero-Point Field. Found Phys 1983;13:1101.

[63] Ycas G. pyNLO: Nonlinear optics modeling for python. 2013, GitHub Repository, GitHub, https://github.com/pyNLO/PyNLO.

[64] Trefethen L. Spectral methods in MATLAB. Philadelphia: SIAM; 2000.

[65] García-Archilla B. Some Practical Experience with the Time Integration of Dissipative Equations. J Comput Phys 1995;25:122.

[66] Melchert O. GNLStools – Data structures and functions for simulation and analysis of the propagation dynamics of laser pulses in nonlinear waveguides. 2022, GitHub Repository, GitHub, https://github.com/omelchert/GNLStools.git. [Online; Accessed 07 June 2022].