

# On the use and validation of the OptFiberLocalError class in PhoSSiL

John Zweck

February 1, 2011

## 1 Introduction

This document augments the online documentation of the classes `cfftw`, `OptSignal`, and `OptFiberLocalError` in PhoSSiL, the Photonics Systems Simulator Library.

The `OptFiberLocalError` class was originally written to solve various forms of the generalized nonlinear Schrödinger equation (NLSE) that are commonly used to model propagation of optical signals in optical fiber communications systems. More recently additional features have been added to the class to model laser systems. In particular, the class can be used to solve the Haus master equation (HME) which is based on a distributed laser propagation model.

We begin by explaining the conventions used for the Fourier transform, definitions of average power and energy, some basics of the split-step Fourier method and a discussion of the different step size selection criteria in the `OptFiberLocalError` class.

Then we discuss several tests we performed to validate the code. These are based on the classical sech-soliton solution of the NLSE and on a chirped sech-soliton solution of the HME. This discussion includes basic information on how PhoSSiL was used to perform these tests.

## 2 The Fourier Transform

In PhoSSiL, Fourier transforms are performed using the FFTW (Fastest Fourier Transform in the West) software.<sup>1</sup>

There are several conventions for the Fourier transform.<sup>2</sup> PhoSSiL uses the Physics convention whereas FFTW is based on the Math convention. The class `cfftw` provides an interface between FFTW and PhoSSiL. However users never need to invoke this class for themselves. Instead it is called by the constructor of the `OptSignal` class.

---

<sup>1</sup>See <http://www.fftw.org/>. Matlab also uses FFTW.

<sup>2</sup>See scanned handwritten lecture notes by Curtis Menyuk on the PhoSSiL website for an overview.

In FFTW the forward discrete Fourier transform  $\mathbf{u} = \text{DFT}_{\text{FFTW}}(\tilde{\mathbf{u}})$  of a vector  $\tilde{\mathbf{u}} = (\tilde{u}_0, \dots, \tilde{u}_{N-1})$  is the vector  $\mathbf{u} = (u_0, \dots, u_{N-1})$ , where

$$u_n = \sum_{m=0}^{N-1} \tilde{u}_m \exp(-2\pi i n m / N). \quad (1)$$

In FFTW the inverse Fourier transform is computed using

$$\tilde{\mathbf{u}} = \frac{1}{N} \text{IDFT}_{\text{FFTW}}(\mathbf{u}), \quad (2)$$

where  $\text{IDFT}_{\text{FFTW}}$  is the unnormalized inverse Fourier transform in FFTW.

In PhoSSiL Fourier transforms are defined using the physics convention as follows. Let  $u(t)$  be a function in the time domain that, for example, represents the complex electric field envelope of the light. Suppose that  $u(t)$  is periodic with period  $T$ . ( $T = \text{OptSignal}::\text{TimeWindow}$ .)<sup>3</sup> We define the Fourier transform of  $u(t)$  to be  $\tilde{u}(\omega)$ , where  $\omega$  is frequency in radians/second, by

$$\tilde{u}(\omega) = \frac{1}{T} \int_0^T u(t) \exp(i\omega t) dt. \quad (3)$$

So, by the Fourier inversion theorem,

$$u(t) = \frac{T}{2\pi} \int_0^{\Omega} \tilde{u}(\omega) \exp(-i\omega t) d\omega, \quad (4)$$

where  $\frac{\Omega}{2\pi} = \text{OptSignal}::\text{FreqWindow}$  in Hz. The reason we use these normalizations is that then  $|u(t)|^2$  and  $|\tilde{u}(\omega)|^2$  both have units of Watts, W.

Let  $\mathbf{u}$  and  $\tilde{\mathbf{u}}$  be  $N$ -vectors discretizing  $u(t)$  and  $\tilde{u}(\omega)$ , respectively ( $N = \text{OptSignal}::\text{qtPoints}$ ). Then  $T \frac{\Omega}{2\pi} = N$  and  $\Delta v = \frac{\Delta \omega}{2\pi} = \text{OptSignal}::\text{DeltaFreq}$  which has units of Hz. Also,  $\Delta t = \text{OptSignal}::\text{DeltaTime} = T/N$ ,  $\Delta v = \Omega/(2\pi N)$  and  $\Delta t \Delta v = 1/N$ . With this discretization the Fourier transform in PhoSSiL is given by

$$\tilde{\mathbf{u}} = \text{DFT}_{\text{PhoSSiL}}(\mathbf{u}) := \frac{1}{N} \text{IDFT}_{\text{FFTW}}(\mathbf{u}), \quad (5)$$

and the inverse Fourier transform is

$$\mathbf{u} = \text{IDFT}_{\text{PhoSSiL}}(\tilde{\mathbf{u}}) = \text{DFT}_{\text{FFTW}}(\tilde{\mathbf{u}}). \quad (6)$$

The methods `cfftw::FFT` and `cfftw::IFFT` implement  $\text{DFT}_{\text{PhoSSiL}}$  and  $\text{IDFT}_{\text{PhoSSiL}}$ , respectively. With these definitions

$$\text{DFT}_{\text{PhoSSiL}} \circ \text{IDFT}_{\text{PhoSSiL}} = \text{Id} \quad (7)$$

is the identity operator.

---

<sup>3</sup>In this document, classes and variables in PhoSSiL are indicated using typewriter font.

### 3 Energy, Average Power and Power Spectral Density

We use the following definitions in the class `OptSignal`.

$$\text{Total Energy (in J = W.s)} = \int_0^T |u(t)|^2 dt \approx \sum_n |u_n|^2 \Delta t \quad (8)$$

and

$$\text{Average Power (in W = J/s)} = \frac{1}{T} \int_0^T |u(t)|^2 dt \approx \frac{1}{N} \sum_n |u_n|^2 \quad (9)$$

so that Total Energy = T Average Power. Parseval's Theorem says that the total energy is also

$$\text{Total Energy} = \int_0^T |u(t)|^2 dt = \frac{T^2}{2\pi} \int_0^\Omega |\tilde{u}(\omega)|^2 d\omega \approx T \sum_m |\tilde{u}_m|^2. \quad (10)$$

Also by Parseval,

$$\text{Average Power} \approx \sum_m |\tilde{u}_m|^2. \quad (11)$$

These quantities are computed using the methods `OptSignal::GetTotalEnergy()` and `OptSignal::GetAveragePower()`.

We define power spectral density to be average power in a bandwidth divided by the bandwidth in Hz. So

$$\text{PSD}_m = \frac{1}{\Delta\nu^2} \int_{2\pi m \Delta\nu}^{2\pi(m+1)\Delta\nu} |\tilde{u}(\omega)|^2 \frac{d\omega}{2\pi} \approx \frac{|\tilde{u}_m|^2}{\Delta\nu} \quad (12)$$

in W/Hz. Then it is easy to check that

$$\text{Average Power} = \sum \text{PSD}_m \Delta\nu. \quad (13)$$

## 4 The NLSE, HME, and CNLSE

### 4.1 The NLSE

With the physics convention the scalar NLS equation is

$$iu_z - \frac{1}{2}\beta'' u_{tt} - \frac{i}{6}\beta''' u_{ttt} + \gamma |u|^2 u - \frac{i}{2}\alpha u = 0. \quad (14)$$

The names of the parameters in `OptFiberLocalError` are

- $\beta'' = \text{FstOrDispFiber}$  in  $\text{s}^2/\text{m}$
- $\beta''' = \text{SndOrDispFiber}$  in  $\text{s}^3/\text{m}$
- $\gamma = \text{gamma}$  in  $(\text{Wm})^{-1}$

- $\alpha = -\text{NepperAttenuationFiber}$  in  $\text{m}^{-1}$

The scalar NLS is invoked when `TypeSolver = SCALAR-NLS`.

In the input parameter file for the `PhoSSiL` class `OptFiberLocalError` the dispersion is to be specified in one of two ways. If the flag `SetFstSndOrDispFlag` is set to true then `FstOrDispFiber =  $\beta''$`  in  $\text{s}^2/\text{m}$  and `SndOrDispFiber =  $\beta'''$`  in  $\text{s}^3/\text{m}$  are input directly. Otherwise, we specify `DispersionFiber` in  $\text{ps}/(\text{nm}\cdot\text{km})$  and `DispSlopeFiber` in  $\text{ps}/(\text{nm}^2\cdot\text{km})$ . Of course the value of `DispersionFiber` depends on a reference wavelength. We can either specify `ReferenceFreq` or `ReferenceWavelength` by setting the other value to 0.0. The formulae used to get the  $\beta$ 's are given in the method `OptFiberLocalError::SetDispersion()`. (Also see Agrawal's Optical Fiber Communications book.) In the alternate `OptFiber` class we specify `DispersionFiber` and `DispSlopeFiber`.

In class `OptFiberLocalError`, for loss we either specify positive `AttenuationFiber` in  $\text{dB}/\text{m}$  or positive `NepperAttenuationFiber` in  $\text{m}^{-1}$  or `TotalLinearAttenuationFiber` greater than 1 (Set the parameters you are not specifying to 0 for `AttenuationFiber` and `NepperAttenuationFiber` and 1 for `TotalLinearAttenuationFiber` or omit them.)

Similarly, the nonlinear coefficient  $\gamma$  can either be input directly, via `NonlinearCoefficientGamma`, or it can be computed from the input parameters `NonLinIndexFiber` in  $\text{m}^2/\text{W}$  and `EffectAreaFiber` in  $\text{m}^2$  using the formula

$$\gamma = \frac{\text{NonLinIndexFiber} * 2 * \pi * \text{CenterFreq}}{\text{EffectAreaFiber} * \text{LightSpeed}}. \quad (15)$$

To specify which parameter (`NonlinearCoefficientGamma` or `NonLinIndexFiber`) is to be used, set the other one to 0.

To change parameters in the course of a simulation one can call the methods `SetFstOrDispFiber()`, `SetSndOrDispFiber()`, `SetAttenuationLinear()`, and `SetGamma()` to set `FstOrDispFiber`, `SndOrDispFiber`, `NepperAttenuationFiber`, and `gamma` directly.

There are also options to solve the vector Manakov equation with or without PMD (`TypeSolver = VECTOR_MANAKOV_PMD` or `TypeSolver = VECTOR_MANAKOV_NO_PMD`). See Curtis Menyuk's paper in the Journal of Engineering Mathematics. Note though that the case with PMD is not so well developed. Instead users are encouraged to use the `OptFiber` class which was specifically designed for PMD modeling, and also includes effects such as fiber loss, chromatic dispersion and the Kerr nonlinearity.

## 4.2 The HME

Similarly, the scalar HME is given by

$$\frac{\partial u}{\partial z} = \left[ -\frac{\ell}{2} + \frac{g}{2} \left( 1 + \frac{1}{\Omega_g^2} \frac{\partial^2}{\partial t^2} \right) - \frac{i}{2} \beta'' \frac{\partial^2}{\partial t^2} + (\delta + i\gamma) |u|^2 \right] u, \quad (16)$$

where

- $\ell$  is the power loss in  $\text{m}^{-1}$ ,  $\ell = \text{NepperAttenuationFiber}$ ;
- $g = g(z)$  is the saturable power gain in  $\text{m}^{-1}$ , given by  $g(z) = \frac{g_0}{1+W(z)/W_{\text{sat}}}$ , where  $W(z)$  is the energy at  $z$ ,  $W_{\text{sat}}$  is the saturation energy and  $g_0$  is the unsaturated gain in  $\text{m}^{-1}$ ;
- $\Omega_g$  is the parabolic gain coefficient `FiberAmplifierParabolicGainCoefficient` in  $\text{s}^{-1}$ ;
- $\delta$  is the `FastSaturableAbsorptionCoefficient`, in  $(\text{Wm})^{-1}$ ;
- $\gamma$  is the Kerr nonlinear coefficient `gamma` in  $(\text{Wm})^{-1}$ .

Simulations with the scalar HME are performed with `TypeSolver = SCALAR-NLS`.

The unsaturated gain of the fiber amplifier can be specified either by setting `FiberAmplifierUnsaturatedGainInDBPerLength` in dB/m or `FiberAmplifierUnsaturatedNepperGainPerLength` in  $\text{m}^{-1}$  or `FiberAmplifierUnsaturatedTotalLinearGain`.

### 4.3 CNLSE

PhoSSiL can also be directed to solve the Coupled Nonlinear Schrödinger Equation (CNLS) in the form

$$\frac{\partial u}{\partial z} = \left[ -\frac{\ell}{2} + i\frac{g}{2} \left( 1 + \frac{1}{\Omega_g^2} \frac{\partial^2}{\partial t^2} \right) + i\Delta\beta - \Delta\beta' \frac{\partial}{\partial t} - \frac{i}{2}\beta'' \frac{\partial^2}{\partial t^2} + (\delta + i\gamma)(|u|^2 + \frac{2}{3}|v|^2) \right] u + i\frac{\gamma}{3}v^2u^* \quad (17)$$

$$\frac{\partial v}{\partial z} = \left[ -\frac{\ell}{2} + i\frac{g}{2} \left( 1 + \frac{1}{\Omega_g^2} \frac{\partial^2}{\partial t^2} \right) - i\Delta\beta + \Delta\beta' \frac{\partial}{\partial t} - \frac{i}{2}\beta'' \frac{\partial^2}{\partial t^2} + (\delta + i\gamma)(|v|^2 + \frac{2}{3}|u|^2) \right] v + i\frac{\gamma}{3}u^2v^*. \quad (18)$$

For this set `TypeSolver = VECTOR_CNLS`. (See C.R. Menyuk, *J. Quantum Electron.* Vol QE-23, pp. 174-176, 1987.) Here  $\Delta\beta$  is the (phase) birefringence (`CNLSPhaseBirefringence` in units of  $\text{m}^{-1}$ ) and  $\Delta\beta' = \frac{\partial\Delta\beta}{\partial\omega}$  is the walk-off birefringence (`CNLSWalkOffBirefringence` in units of s/m). If we let  $\mathbf{U} = (u, v)^T$  be the Jones vector at  $(t, z)$ , then the vector form of the nonlinear terms in the CNLS can be written in the form (See Curtis Menyuk, *J. Eng. Math.* paper):

$$\frac{\partial \mathbf{U}}{\partial z} = i\gamma \left[ |\mathbf{U}|^2 \mathbf{U} - \frac{1}{3} (\mathbf{U}^\dagger \boldsymbol{\sigma}_2 \mathbf{U}) \boldsymbol{\sigma}_2 \mathbf{U} \right] \quad (19)$$

where  $\boldsymbol{\sigma}_2$  is one of the Pauli spin matrices

$$\boldsymbol{\sigma}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \boldsymbol{\sigma}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \boldsymbol{\sigma}_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \boldsymbol{\sigma}_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (20)$$

The Stokes parameters are defined by

$$[S_0, S_1, S_2, S_3] = \mathbf{U}^\dagger [\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_3, \boldsymbol{\sigma}_1, -\boldsymbol{\sigma}_2] \mathbf{U}. \quad (21)$$

Then the Stokes parameters then the vector form of the CNLS can be written as

$$\frac{\partial \mathbf{U}}{\partial z} = i\gamma \left[ S_0 \mathbf{U} + \frac{1}{3} S_3 \boldsymbol{\sigma}_2 \mathbf{U} \right]. \quad (22)$$

Applying the product rule to the definition of the Stokes parameters and substituting in the CNLS gives the evolution equations

$$\frac{\partial S_0}{\partial z} = 0 \quad (23)$$

$$\frac{\partial S_1}{\partial z} = \frac{2\gamma}{3} S_2 S_3 \quad (24)$$

$$\frac{\partial S_2}{\partial z} = -\frac{2\gamma}{3} S_1 S_3 \quad (25)$$

$$\frac{\partial S_3}{\partial z} = 0. \quad (26)$$

Therefore the effect of the nonlinearity is that, on the Poincaré sphere the Stokes vector  $(S_1, S_2, S_3)$  rotates around the circle of latitude determined by the constant value of  $S_3$ , and moves west at a constant angular speed that is proportional to  $S_3$ .

## 5 Numerical solution via the symmetric split-step scheme

### 5.1 Solving the NLSE using the symmetric split-step scheme

The split-step scheme for solving equations like the NLSE relies on the solution of the equation in two special cases. First, if there is no nonlinearity, then in the frequency domain we have the exact analytical solution

$$\tilde{u}(z + \Delta z, \omega) = \tilde{u}(z, \omega) \exp\left(\frac{1}{2}\alpha\Delta z + \frac{i}{2}\beta''\omega^2\Delta z + \frac{i}{6}\beta'''\omega^3\Delta z\right). \quad (27)$$

Second, in the presence of nonlinearity alone we in the time domain we have the exact analytical solution

$$u(z + \Delta z, t) = u(z, t) \exp(i\gamma|u|^2\Delta z). \quad (28)$$

We use the symmetric split-step Fourier scheme in PhoSSiL. This means that for the NLSE

above the solution at  $z + \Delta z$  is given in terms of the solution at  $z$  by

$$\tilde{u}_{D1}(\omega) = \tilde{u}(z, \omega) \exp([\frac{1}{2}\alpha + \frac{i}{2}\beta''\omega^2 + \frac{i}{6}\beta'''\omega^3]\frac{\Delta z}{2}) \quad (29)$$

$$u_{D1}(t) = \text{IDFT}(\tilde{u}_{D1}(\omega)) \quad (30)$$

$$u_{\text{NL}}(t) = u_{D1}(t) \exp(i\gamma|u_{D1}(t)|^2\Delta z) \quad (31)$$

$$\tilde{u}_{\text{NL}}(\omega) = \text{DFT}(u_{\text{NL}}(t)) \quad (32)$$

$$\tilde{u}(z + \Delta z, \omega) = \tilde{u}_{\text{NL}}(\omega) \exp([\frac{1}{2}\alpha + \frac{i}{2}\beta''\omega^2 + \frac{i}{6}\beta'''\omega^3]\frac{\Delta z}{2}) \quad (33)$$

$$u(z + \Delta z, t) = \text{IDFT}(\tilde{u}(z + \Delta z, \omega)). \quad (34)$$

With this scheme, the local error is  $O(\Delta z^3)$  and the global error is  $O(\Delta z^2)$ .

## 5.2 Solving the HME using a nested symmetric split-step scheme

For the HME we do dispersion, loss, and Kerr nonlinearity steps as for the NLSE. The saturable gain and fast saturable absorption are done using Heun's Method (aka modified Euler method or 2nd-order Runge-Kutta), which is locally  $O(\Delta z^3)$ . In the time domain we do a half-step of Kerr nonlinearity followed by a whole step of fast saturable absorption followed by a half-step of Kerr nonlinearity. This split-step method means that the Kerr nonlinearity part is done exactly. For the quadratic fast saturable absorption in the standard HME we also derived an exact analytical solution, which we use if `TypeFastSaturableAbsorption = QUADRATIC_ANALYTIC`. If `TypeFastSaturableAbsorption = QUADRATIC`, we use Heun's Method. Overall the time domain part has a local error of  $O(\Delta z^3)$ . Similarly, in the frequency domain we choose to do the chromatic dispersion and loss exactly in two half-steps with a full step of saturable parabolic gain in between, so that the frequency domain part is also  $O(\Delta z^3)$ . Since we also have two half steps in the frequency domain with a whole step in the time domain in between, the entire nested split-step method is locally  $O(\Delta z^3)$  and hence globally is  $O(\Delta z^2)$  (at least when the step size is constant).

## 5.3 Solving the CNLSE via a Jones space transformation of the nonlinear terms

The basic method for solving the CNLSE is as for the NLSE/HME above. However to solve (19) in the time-domain we proceed as follows.

Our goal is to get rid of the four-wave mixing type terms  $v^*u^2$  and  $u^*v^2$  in (18). To do so we let  $\mathbf{U} = A\mathbf{W}$  where  $A$  is a  $2 \times 2$  unitary matrix which diagonalizes  $\sigma_2$ :

$$\sigma_2 = A\sigma_3A^\dagger \quad A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}. \quad (35)$$

Note that the basis  $\{(1, 0)^T, (0, 1)^T\}$  of Jones space is transformed to the basis  $\{\frac{1}{\sqrt{2}}(1, i)^T, \frac{1}{\sqrt{2}}(1, -i)^T\}$ . The corresponding sets of vectors in Stokes space are  $\{(1, 0, 0)^T, (-1, 0, 0)^T\}$  and  $\{(0, 0, -1)^T, (0, 0, 1)^T\}$ ,

ie  $A$  transform coordinates from a basis of linearly polarized variables to a basis of circularly polarized variables. Then (19) is transformed to

$$\frac{\partial \mathbf{W}}{\partial z} = i\Lambda \mathbf{W}, \quad (36)$$

where

$$\Lambda = S_0 \sigma_0 + \frac{1}{3} S_3 \sigma_3. \quad (37)$$

As we saw above  $S_0$  and  $S_3$  are constant in  $z$ . Therefore  $\Lambda$  is also constant in  $z$ . This makes sense since in the circularly polarized basis, energy is split equally between the two basis vectors as the Stokes vector of the light is just rotating around a circle of latitude.

Therefore the exact solution of (36) is

$$\mathbf{W}(z + \Delta z) = \exp(i\Lambda \Delta z) \mathbf{W}(z) \quad (38)$$

and so the exact solution of (19) is

$$\mathbf{U}(z + \Delta z) = A \exp(i\Lambda \Delta z) A^\dagger \mathbf{U}(z). \quad (39)$$

Multiplying out the matrices on the right hand side gives

$$\mathbf{U}(z + \Delta z) = e^{i\alpha} \begin{pmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{pmatrix} \mathbf{U}(z), \quad (40)$$

where  $\alpha = \gamma S_0 \Delta z$  and  $\beta = \frac{1}{3} \gamma S_3 \Delta z$ , which is easy to implement. (Note that the components of  $\mathbf{U}$  are complex and so we can't express the rotation of  $\mathbf{U}$  through angle  $\beta$  as a complex multiplication by  $e^{i\beta}$ .)

If we write  $\mathbf{W} = (r, s)^T$ , then (36) becomes

$$\frac{\partial r}{\partial z} = i \frac{2\gamma}{3} (|r|^2 + 2|s|^2) r \quad (41)$$

$$\frac{\partial s}{\partial z} = i \frac{2\gamma}{3} (2|r|^2 + |s|^2) s, \quad (42)$$

where

$$r = \frac{1}{\sqrt{2}}(u - iv) \quad s = \frac{1}{\sqrt{2}}(u + iv). \quad (43)$$

Note that in the special cases  $u = 0$  or  $v = 0$ , this system of equations reduces to the nonlinear terms in the NLSE.



## 6 Step Size Selection Criteria in `OptFiberLocalError`

For more details on step size selection criteria, see "Optimization of the split-step Fourier method in modeling optical fiber communications systems", by O.V. Sinkin *et al.* in *J. Lightwave Technology*, **21** (1), pp. 61-68, 2003, available on the PhoSSiL web site.

In `OptFiberLocalError` we have four step size selection options. The input parameter `TypeStepSizes` can take the values `LOCAL_ERROR_3RD = 1`, `LOCAL_ERROR_2ND = 2`, `WALK_OFF = 3`, `CONSTANT = 4`.

No matter what method we use we always choose the last step in a particular fiber so that the total propagation length in that fiber is correct.

The rest of this section applies whenever `TypeSolver` is either `SCALAR-NLS` or `VECTOR-MANAKOV_NO_PMD`. The option `VECTOR-MANAKOV_NO_PMD` should be used when the PMD in the system is negligible but you want to use a simulation with two orthogonal polarization states. The third option for `TypeSolver` is `VECTOR-MANAKOV_PMD`, which has not been thoroughly tested yet. (For PMD modeling, use the class `OptFiber` instead.)

### 6.1 CONSTANT method

In this method the step sizes are constant. The number of steps per fiber is `NumStepsBetweenScatterings`. (This variable name was chosen for the case when there is PMD. Hence the funny name.)

### 6.2 WALK\_OFF method

The code uses a variant of the criterion for choosing  $\Delta z$  given in the paper of Sinkin *et al.* For each type of fiber (e.g., SMF, DCF, DSF) the user inputs a value for the `WalkOffParameter` and the code sets the step size in that fiber type to be

$$\Delta z = \frac{\text{WalkOffParameter}}{|D_1\lambda_1 - D_2\lambda_2|} \quad (44)$$

where  $D_1$  and  $D_2$  are the dispersions in ps/nm-km at wavelengths  $\lambda_1$  and  $\lambda_2$ . We choose  $\lambda_1, \lambda_2$  as follows:

- WDM Case:  $\lambda_1, \lambda_2$  are the central wavelengths of the edge channels
- Single Channel Case:  $\lambda_{1,2}$  correspond to frequencies  $\omega_{1,2} = \text{CenterFreq} \pm \text{RootMeanSquare spectral width of signal}$ .

### 6.3 LOCAL\_ERROR methods

First read Sinkin's paper for the basic algorithm. Here we just discuss issues that are not mentioned in the paper.

For this method you need to set `RelativeErrorGoal`, and `DeltaZMax`. The algorithm aims to keep the local error in the range  $0.5 * \text{RelativeErrorGoal} < \text{LocalError} < 2 * \text{RelativeErrorGoal}$ . To determine the initial step size in a given fiber for the very first time through that fiber we take an initial guess of `DeltaZ = 0.5 * DeltaZMax`. Note though that the algorithm may not actually take a step of that size. For subsequent passes through the fiber we use the step size of the first step that was taken in the fiber last time we went through it.

Note that when `TypeGainLoss = FIBER_AMPLIFIER` the value of `DeltaZMax` is selected so as to correctly compute the gain and noise along the fiber. See the online documentation on `OptFiberLocalError` for more details.

For `LOCAL_ERROR_3RD` for each step we keep the high-order solution given by Eq. (9) of Sinkin's paper. For `LOCAL_ERROR_2ND` we keep the fine step solution, given by Eq. (8) of Sinkin's paper.

## 7 Examples and Code Validation

We present the results of simulations that both validate the code and demonstrate some important properties of propagation in optical fiber systems.

### 7.1 Soliton Solution of NLSE

We consider the classical sech-pulse solution of the NLSE

$$iu_z - \frac{1}{2}\beta'' u_{tt} + \gamma |u|^2 u = 0, \quad (45)$$

with  $\beta'' = -0.1 \text{ ps}^2/\text{km}$ ,  $\gamma = 2.2 \text{ W}^{-1} \text{ km}^{-1}$  and an initial pulse

$$u(z=0, t) = \sqrt{\frac{-\beta''}{\gamma \tau^2}} \text{sech}\left(\frac{t}{\tau}\right) \quad (46)$$

where  $\text{FWHM} = 2 \log(1 + \sqrt{2}) \tau = 4.0 \text{ ps}$ . The peak power is  $P_0 = 8.8273789 \text{ mW}$ . The formula for the solution is

$$u(z, t) = \sqrt{\frac{-\beta''}{\gamma \tau^2}} \text{sech}\left(\frac{t}{\tau}\right) \exp\left(i \frac{-\beta'' z}{2\tau^2}\right). \quad (47)$$

Observe that the power function  $|u(t)|^2$  is constant but the phase will evolve as a function  $z$ . Also note that classical solitons only exist when  $\beta'' < 0$ .

In Fig. 1, the optical power  $|u(t)|^2$  of the soliton is shown at 0 km and 500 km. Agreement is excellent. In Fig. 2, the real and imaginary parts of  $u$  are shown at 10,000 km. The agreement between theory and simulation is excellent. For this simulation we used `qtPoints = 4096`, `StringLength = 4`, `BitRateChann = 40 GHz`, and constant Step sizes with 50,000 steps. It is very important that the peak power in the simulation and theory agree as accurately as possible. Likewise for the FWHM.

In Fig. 3 we propagate the initial pulse in Eq. (46) for 300 km including only dispersion in the propagation model. The split-step method uses just 1 step since there is no nonlinearity. In Fig. 4 we show the same pulse after 500 km of propagation through fiber which only has dispersion. The result is not correct since the pulse is wider than the `TimeWindow` and so has wrapped around on itself. To overcome this problem you would need to make `StringLength` and `qtPoints` both larger.

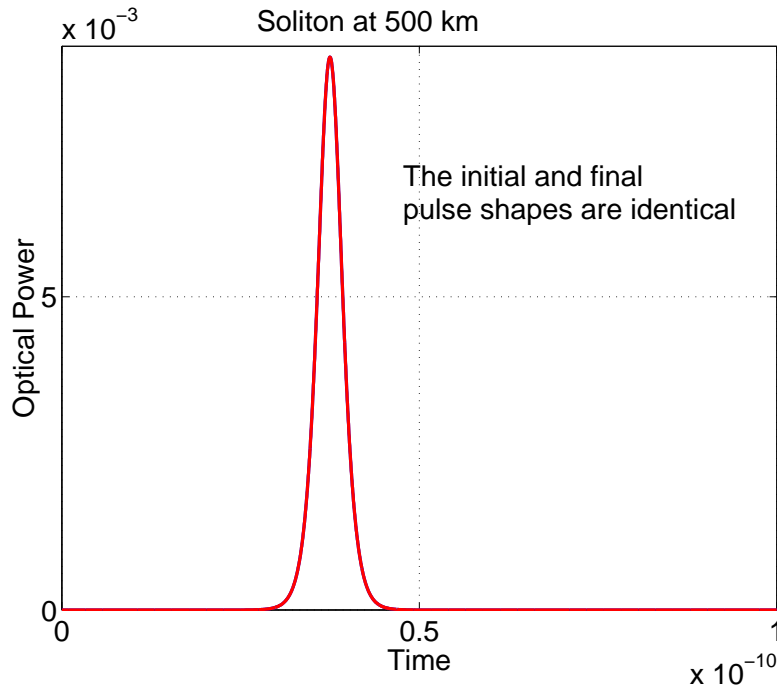


Figure 1: Optical power of classical soliton at 0 km and 500 km.

## 7.2 Pure Third Order Dispersion

Suppose that we have a fiber with pure third-order dispersion, i.e., the fiber has no nonlinearity ( $\gamma = 0$ ),  $\beta'' = 0$  and  $\beta''' = 1 \text{ ps}^3/\text{km}$  and that the initial pulse is a Gaussian with a FWHM of 4 ps. In Fig. 5 we show the output pulse after 100 km of propagation through such a fiber. Oscillations are present at times greater than the central time of the initial pulse, as we would expect.

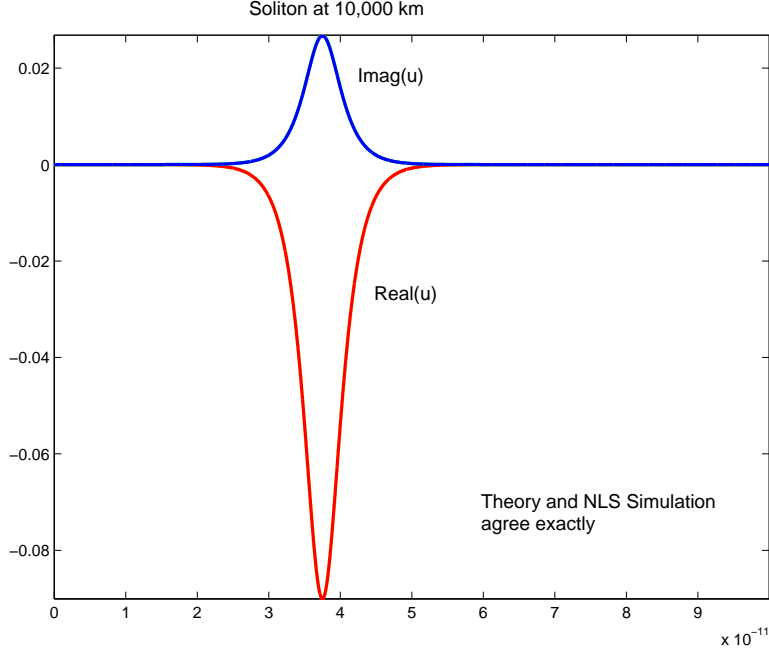


Figure 2: Real and imaginary parts of classical soliton at 10,000 km. Theory and simulation of the NLSE are in excellent agreement.

### 7.3 Frequency shift invariance

We simulate the propagation of a Gaussian pulse at  $\nu_{\text{ch}} = 193400$  GHz through 50 km of fiber with  $D = 20$  ps/nm-km. There is no dispersion slope or nonlinearity. We choose `qtPoints = 4096` and `StringLength = 32` to give `FrequencyWindow = 1280` GHz.

We do two simulations. For the first we choose `CenterFreq = 193400` GHz and for the second `CenterFreq = 193300` GHz. The outputs are shown in Fig 6. Both have the same shape but the second is moved by  $\Delta t = -801.5$  ps. This time shift is given by the formula

$$\Delta t = 2\pi\beta''z\Delta\nu, \quad (48)$$

where  $\Delta\nu = \nu_{\text{ch}} - \nu_0 = +100$  GHz and  $\beta'' = -25.513$  ps<sup>2</sup>/km. In particular, the signs of the time shifts are the same in the theory and with the NLS simulation. One way to derive the theoretical time shift is to use the solution of the linear dispersive equation with a Gaussian initial condition as one has in the Physics convention

$$u(z=0, t) = \exp(-t_*^2/2) \exp(-i\Delta\nu_* t_*) \quad (49)$$

where  $t_* = t/\tau_0$  and  $\Delta\nu_* = \Delta\nu\tau_0$  are unitless and  $\tau_0$  is a measure of the pulsewidth. The solution of the linear dispersive equation

$$iu_z - \frac{1}{2}\beta''u_{tt} = 0 \quad (50)$$

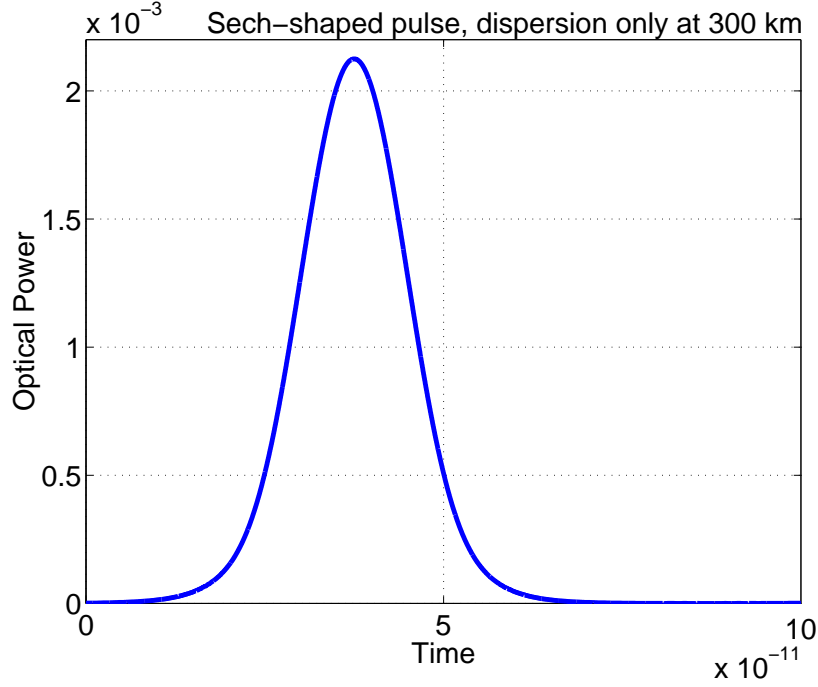


Figure 3: The result of propagating an initial sech-shaped pulse through dispersive fiber with  $\beta'' = -0.1 \text{ ps}^2/\text{km}$  for 300 km. Note that without the nonlinearity the pulse spreads in time.

is

$$u(z,t) = \frac{1}{\sqrt{1 - iB_*(z)}} \exp\left(-\frac{(t_* + i\Delta v_*)^2}{2(1 - iB_*(z))}\right) \exp(-\Delta v_*^2/2), \quad (51)$$

where  $B_*(z) = \int \beta''(w)dw/\tau_0^2$  is the normalized accumulated dispersion. From Eq. (51) it is easy to verify Eq. (48).

## 7.4 Tyco System

In Fig. 7 we plot the optical power versus time for propagation of a noise-free 32 bit PRBS through the “Tyco” system. This system is similar to a WDM system designed by Tyco in about 1998. Details of the system can be found, for example, in Ronald Holzloehner’s Ph.D. thesis on the PhoSSiL web site. The input pulses have a peak power of 1 mW. Plots of optical power for an isolated pulse versus time at the end of each span of the dispersion map are shown in Fig. 8. The initial pulse width ( $z = 0$ ) is about 30 ps, and after the pre-compensation it is about 200 ps. The final pulse width is about 22 ps. The corresponding electrical eye is shown in Fig. 9. Notice that the left side of the eye is steeper than the right side, as one would expect for a causal filter. (Causality implies that if the signal is zero for all times less than  $t_0$  then the filtered signal must be zero there too.) We also checked that the  $Q$ -factor and Eye Opening are invariant to shifts in the central frequency of the FreqWindow.

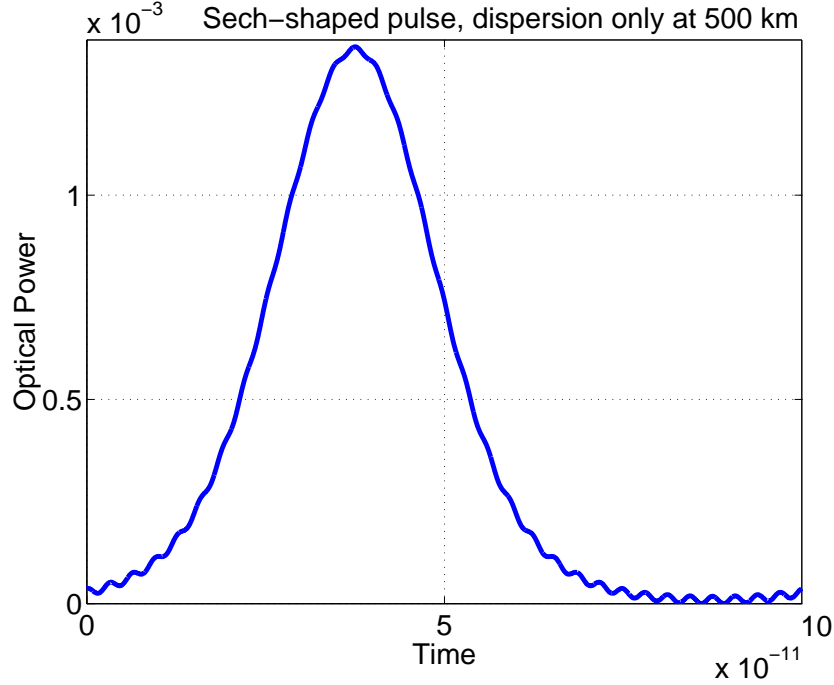


Figure 4: The result of propagating an initial sech-shaped pulse through dispersive fiber with  $\beta'' = -0.1 \text{ ps}^2/\text{km}$  for 500 km. The oscillations are numerical artifacts that occur because the time window is too narrow to contain the entire pulse.

## 7.5 Soliton solution of HME

The Haus master equation has a soliton solution of the form

$$u(z, t) = A [\text{sech}(t/\tau)]^{1+i\beta} \exp(i\psi z + i\theta) \quad (52)$$

where we recall that  $[\text{sech}x]^{1+i\beta} = \text{sech}x \exp(i\beta \ln \text{sech}x)$ . In Haus *et al.*, JOSA B, **8** (10), pp 2068-2076, 1991, the authors derive formulae for the pulse parameters  $A$ ,  $\tau$ ,  $\beta$  and  $\psi$  in terms of the material parameters of the laser, i.e in terms of the coefficients in the HME. In the online documentation of `OptFiberLocalError` there is a link to a Matlab function based on the JOSA B paper that computes the pulse parameters in terms of the laser parameters and checks whether the resulting solution is stable. Using the parameters  $A$ ,  $\tau$ ,  $\beta$  of the input pulse we can solve the HME numerically using `OptFiberLocalError` to obtain an output pulse and compare to the analytical formula above. In the following plots we show the agreement between the formula and the simulation for various values of the laser parameters.

If we choose laser parameters of  $g_0 = 3$ ,  $W_{\text{sat}} = 1$ ,  $\ell = 2$ ,  $\Omega_g = \sqrt{2}$ ,  $\delta = 1$ ,  $\gamma = 4$ ,  $\beta'' = 3$  then the pulse parameters are  $W = 0.5$ ,  $A = 0.0755794477$ ,  $\tau = 43.76559087$ ,  $\text{FWHM} = 2 \log(1 + \sqrt{2})\tau = 77.147633193$ ,  $\beta = 5.105983$ ,  $\psi = 0.02229926$ ,  $g = 2$  and the stability parameter is 5.56 which, since it is positive, implies the pulse is stable. We choose a time window of 1000 s discretized using 4096 points. We used `FiberAmplifierGainSaturationMaxGainPerStep = 1.01`

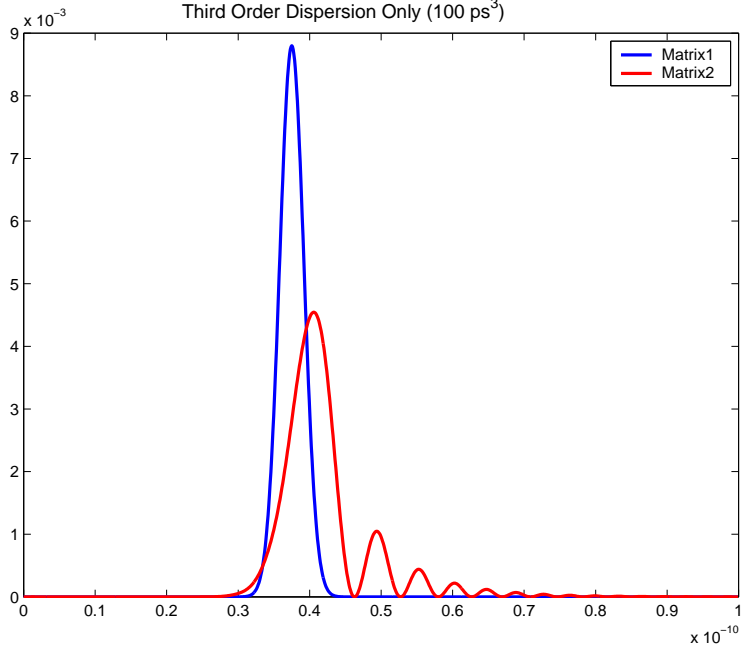


Figure 5: Pure third order dispersion of a Gaussian input pulse.

and  $\text{RelativeErrorGoal} = 10^{-6}$ . In Figs. 10 and 11, 12 we show the real and imaginary parts of the solution at  $z = 300$  and power and phase at  $z = 1000$ . After about 10 m the value of  $g$  stabilized to  $g = 1.9971$  corresponding to an energy of  $W = 0.502178$ . Agreement is excellent at  $z = 300$  m but there is a slight discrepancy at  $z = 1000$  m.

We checked that the results at  $z = 300$  m do not depend on the discretization by changing maximum step size, time window, frequency window and we found no difference in the results. However, we do need to specify the pulse parameters very accurately in order to get the level of agreement seen in these figures.

The dispersion length is  $L_D = \text{FWHM}/|\beta''| = 25.6$  m and the nonlinear length is  $L_{NL} = 1/(\gamma P_0) = 43.86$  m, where  $P_0$  is the peak power. So a propagation distance of  $z = 300$  m is about 6 nonlinear lengths and thus is a good test of the code.

We note that all effects are important in this simulation, and the results therefore validate of the code. For example, fast saturable absorption is important, since if we reduce  $\delta$  to 0.1, say, then the theory shows that the pulse will be unstable. Also, the value of  $\Omega_g$  is small enough for the filtering action to affect the pulse. This is because of the following observations. First, the FWHM in the frequency domain of the Gaussian filter that corresponds to propagation by distance  $z$  due to the parabolic gain shaped fiber amplifier is  $\text{FWHM} = \sqrt{\ln 2} \Omega_g / (\pi \sqrt{gz})$  Hz. For this FWHM to equal the frequency-FWHM of the pulse, which is 0.04 Hz, we need  $z = 44$  m. In other words, every 44 m the fiber amplifier does a Gaussian filtering with the same frequency-FWHM of the pulse.

Because our maximum step size is 0.007 m, the effective filter width for one step is about 3 Hz

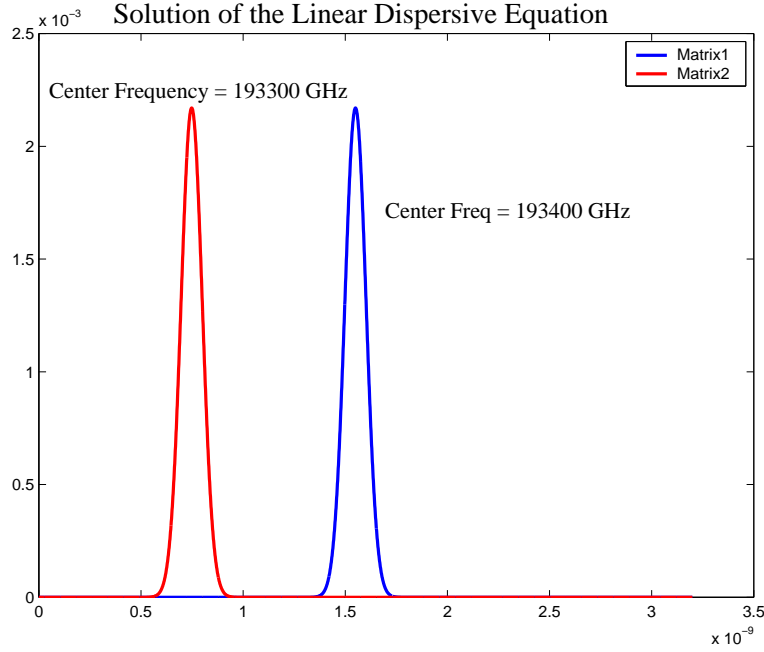


Figure 6: Result of a simulation that demonstrates the frequency-shift invariance of the NLSE.

which is 80 times the pulse width in frequency. Therefore there is no additional loss in each step and the gain is therefore constant. Note however, that if we reduce  $\Omega_g$  to 1, then the theory predicts that the pulse will be unstable.

We also studied the effect of using a larger saturation energy of  $W_{\text{sat}} = 10$  and  $g_0 = 2.1$ , so that  $g \approx 2$  will still hold. In this case the gain converges to  $g = 1.9969$  which would suggest that the energy should converge to 0.5023. However it converges to 0.516261, which is due to the fact that the gain takes about 60 m to decrease to its limiting value. (When  $W_{\text{sat}} = 1$  and  $g_0 = 3$  it only takes about 20 m.) As a consequence there is a slight disagreement between theory and simulation: at  $z = 300$  m both the power and phase disagree a little in the center of the pulse, see Fig. 13, 14. We expect that this pulse will converge at large  $z$  to a soliton, but with slightly different parameters.

When  $W_{\text{sat}} = 0.1$  and  $g_0 = 12$  the agreement is excellent once again and in this case the convergence distance is less than 20 m.



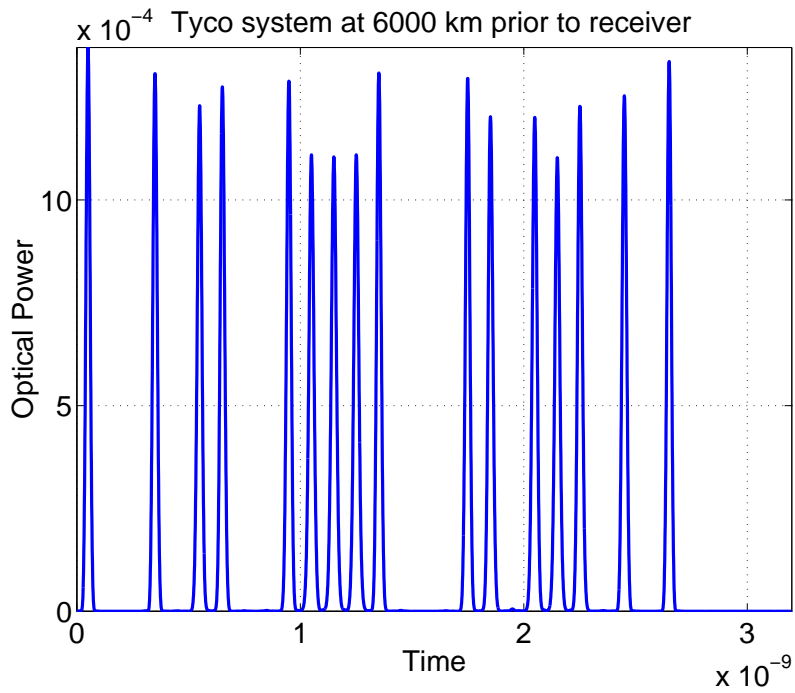


Figure 7: The output optical power for a 32-bit PRBS in the Tyco system.

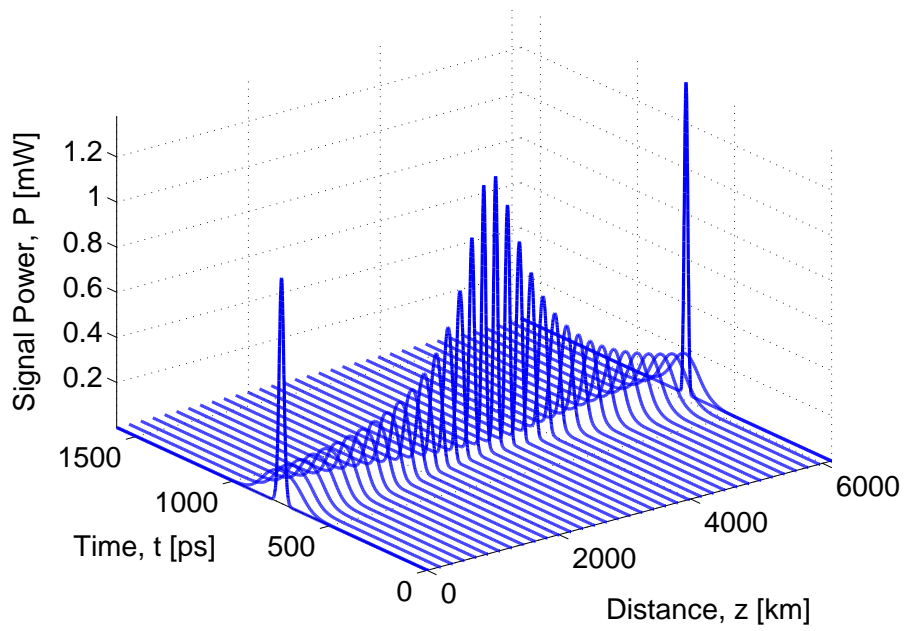


Figure 8: The optical power in the Tyco system as a function of distance and time for an isolated mark.

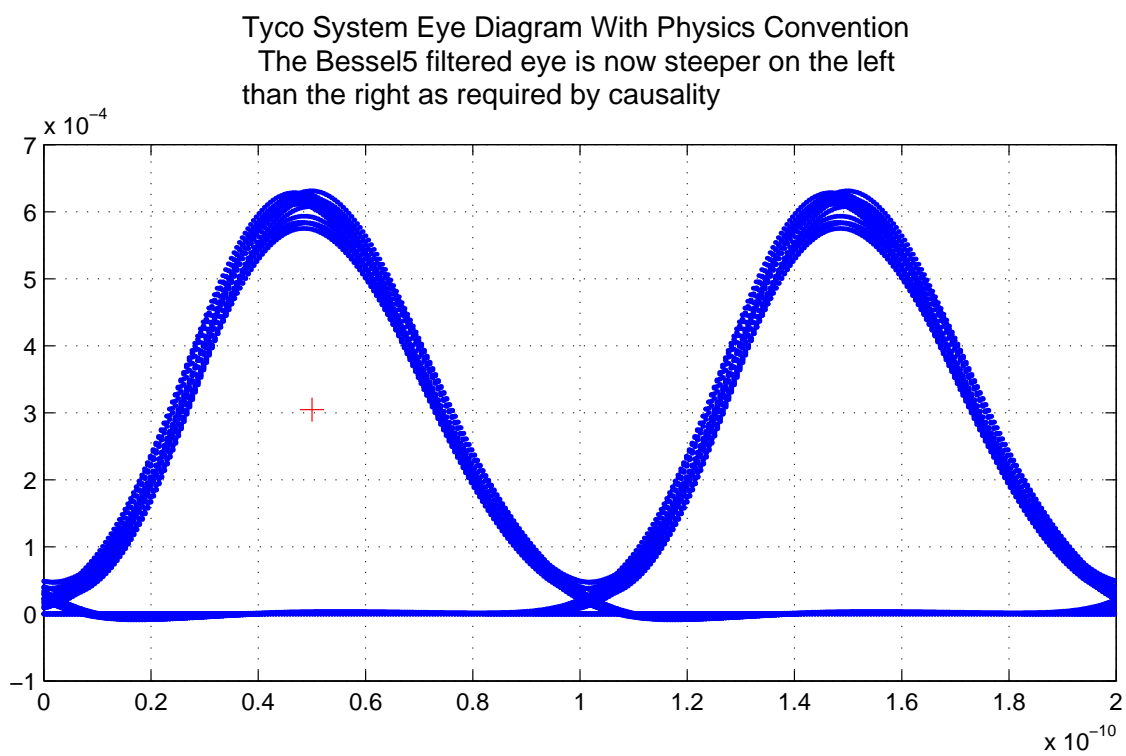


Figure 9: The electrical eye diagram for the noise-free Tyco system. The clock recovery time is indicated with a plus sign.

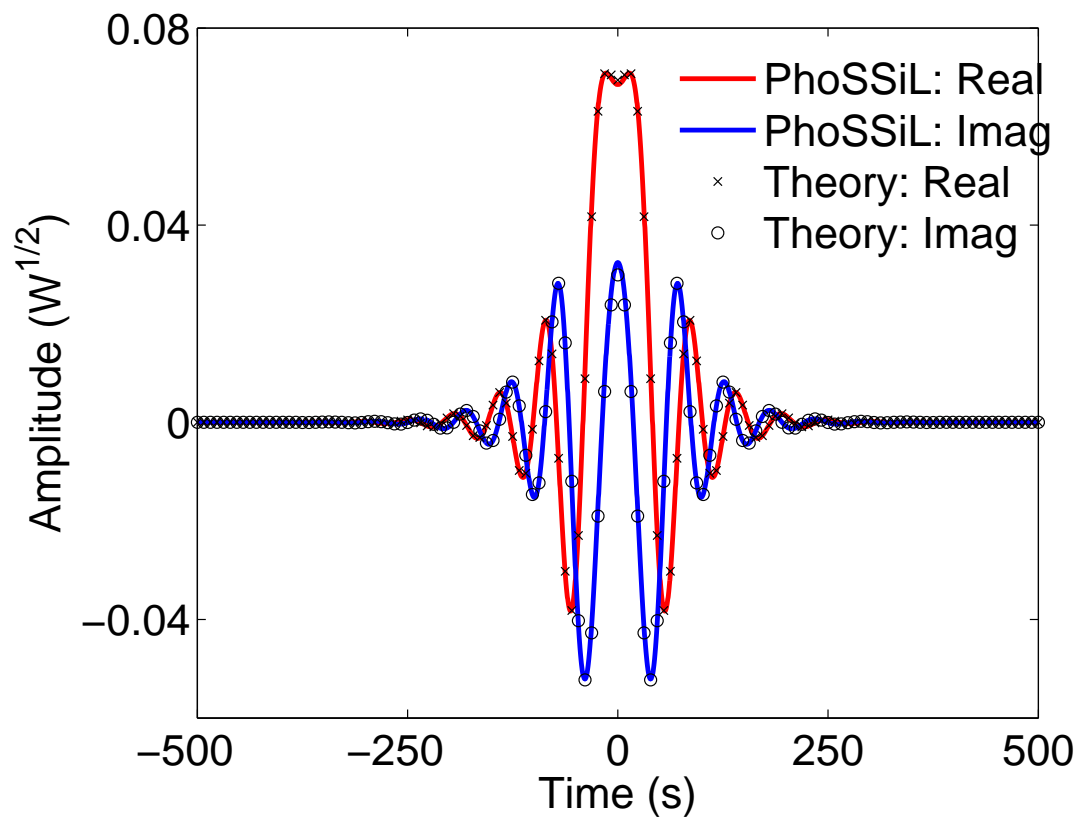


Figure 10: The real and imaginary parts of the solution of the HME at  $z = 300$  m.

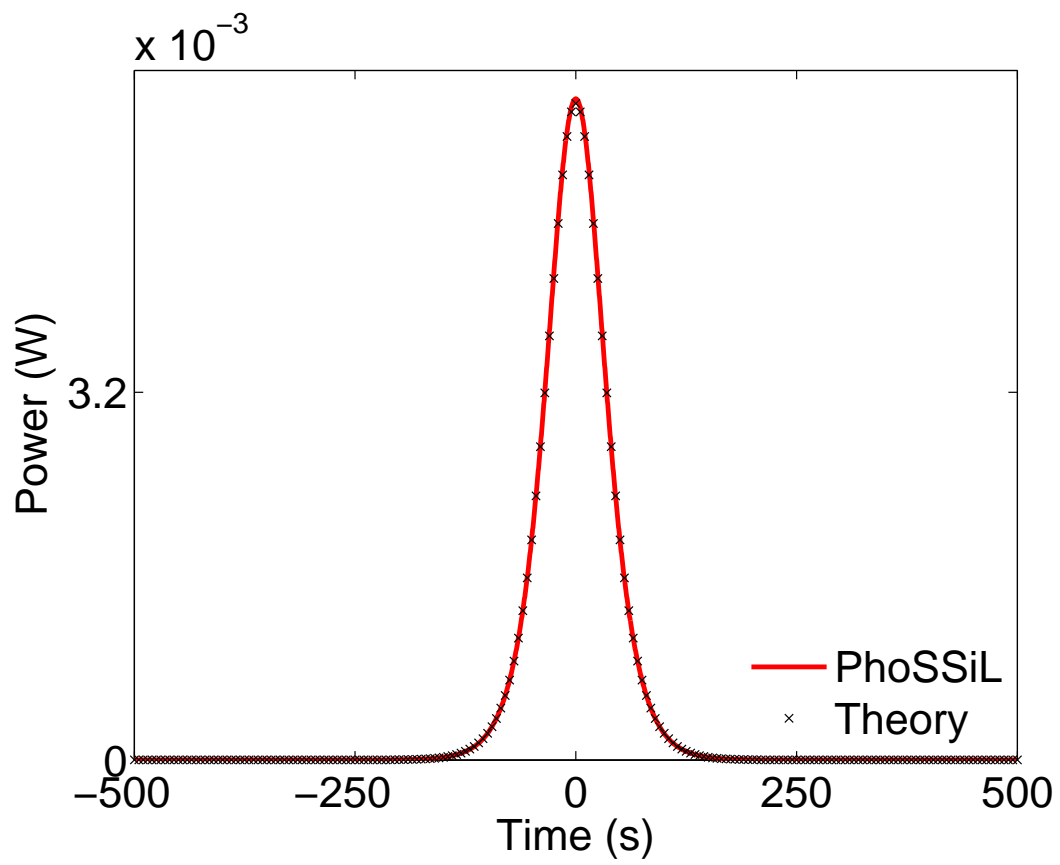


Figure 11: The power of the solution of the HME at  $z = 1000$  m

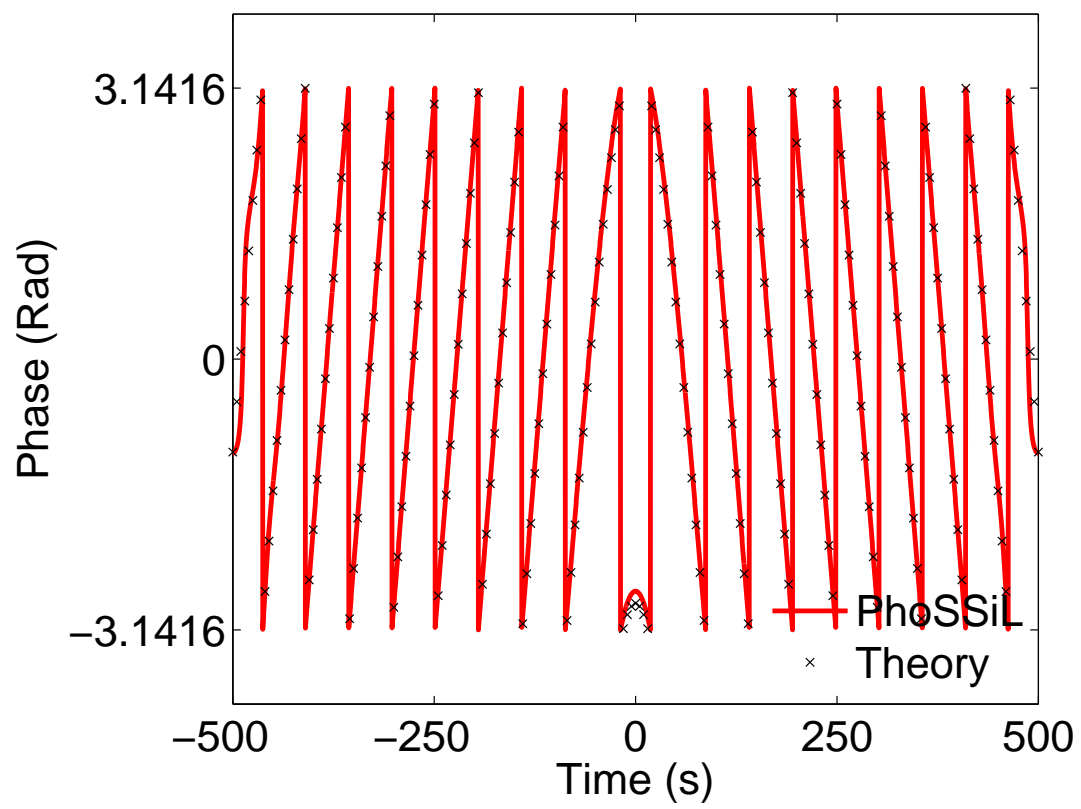


Figure 12: The phase of the solution of the HME at  $z = 1000$  m

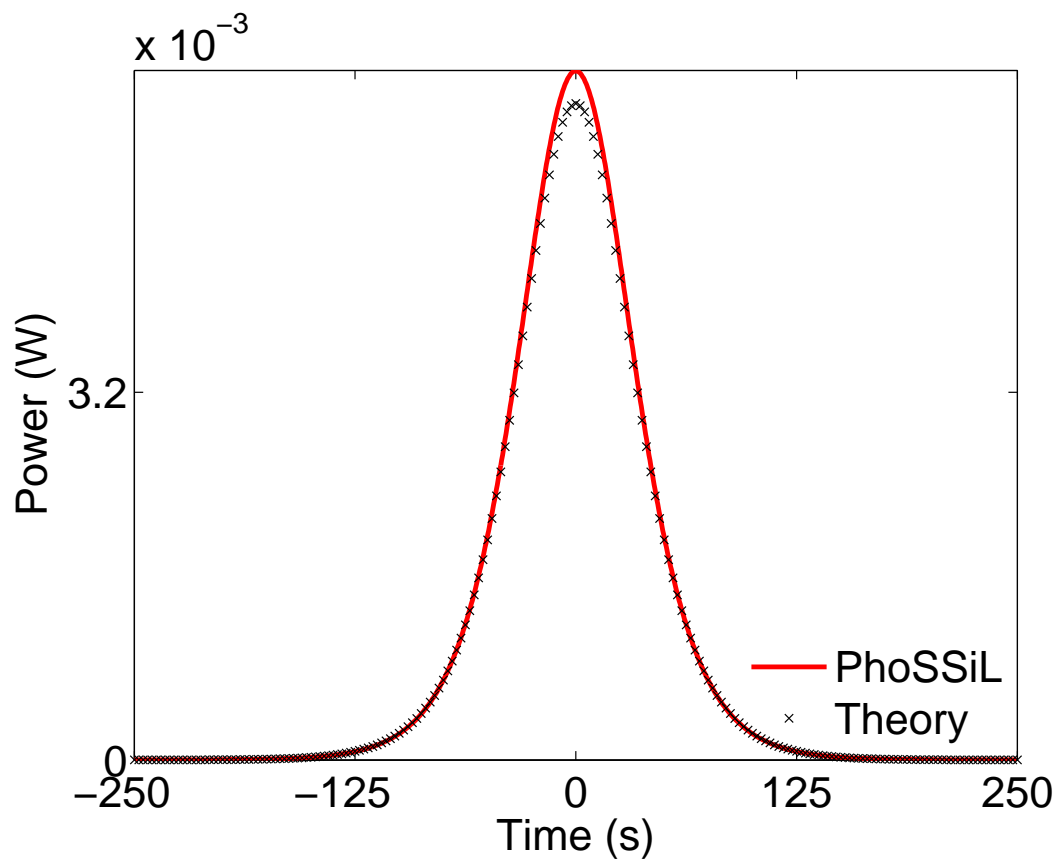


Figure 13: The power of the solution of the HME at  $z = 300$  m for  $W_{\text{sat}} = 10$  and  $g_0 = 2.1$ .

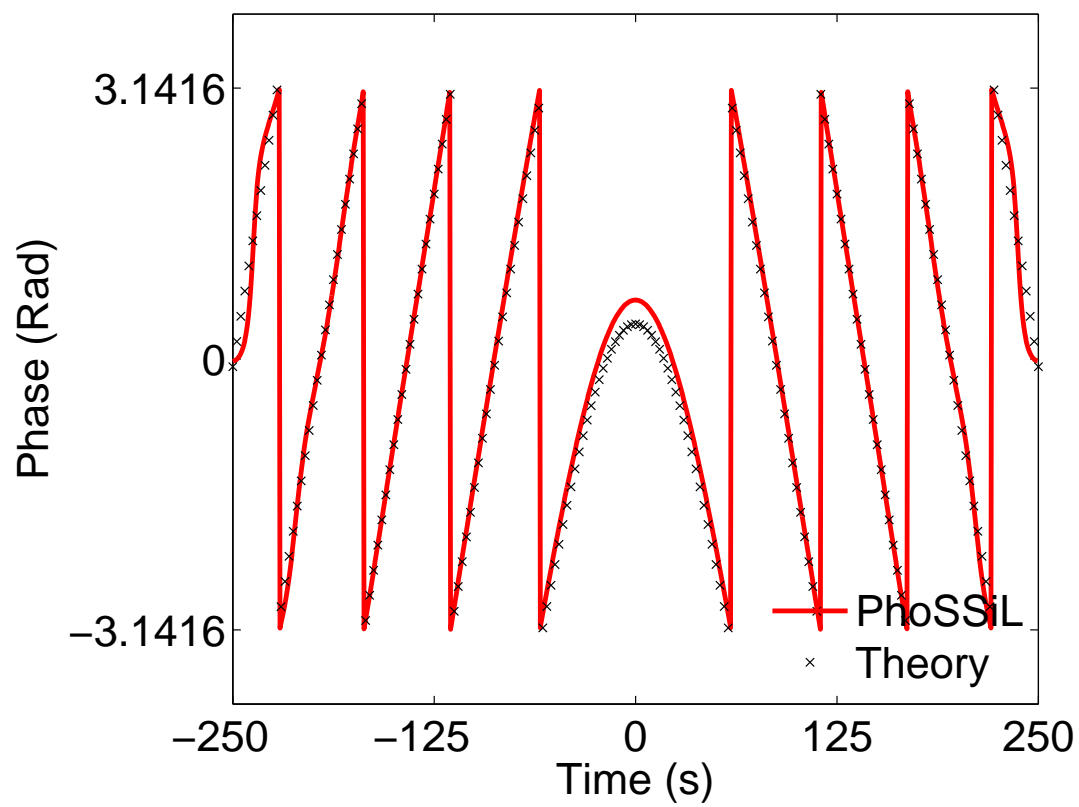


Figure 14: The phase of the solution of the HME at  $z = 300$  m for  $W_{\text{sat}} = 10$  and  $g_0 = 2.1$ .