



SPiP: A computer program implementing the Interaction Picture method for simulation of light-wave propagation in optical fibre[☆]



Stéphane Balac^{a,*}, Arnaud Fernandez^{b,c}

^a UMR FOTON, CNRS, Université de Rennes 1, ENSSAT, 22305 Lannion, France

^b Université de Toulouse, UPS, Toulouse, France

^c Laboratoire d'Analyse et d'Architecture des Systèmes LAAS, CNRS, Université de Toulouse, BP 54200, 31031 Toulouse, France

ARTICLE INFO

Article history:

Received 4 April 2015

Received in revised form

9 October 2015

Accepted 17 October 2015

Available online 31 October 2015

Keywords:

Generalized non-linear Schrödinger equation

Interaction Picture method

Embedded Runge–Kutta method

Adaptive step-size control

ABSTRACT

The computer program SPiP is aimed at solving the Generalized Non-Linear Schrödinger equation (GNLSE), involved in optics e.g. in the modelling of light-wave propagation in an optical fibre, by the Interaction Picture method, a new efficient alternative method to the Symmetric Split-Step method. In the SPiP program a dedicated costless adaptive step-size control based on the use of a 4th order embedded Runge–Kutta method is implemented in order to speed up the resolution.

Program summary

Program title: SPiP

Catalogue identifier: AEYQ_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEYQ_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Free software CeCILL license (see www.cecill.info/index.en.html)

No. of lines in distributed program, including test data, etc.: 81174

No. of bytes in distributed program, including test data, etc.: 6296651

Distribution format: tar.gz

Programming language: C.

Computer: Desktop computer.

Operating system: Linux, MS Windows.

RAM: 8 Giga bytes

Classification: 4.12, 18.

External routines: FFTW, a C subroutine library for computing the discrete Fourier transform, see [1] and <http://www.fftw.org>. Gnuplot, a portable command-line driven graphing utility, see [2] and <http://www.gnuplot.info>.

Nature of problem: The program solves the Generalized Non-Linear Schrödinger Equation (GNLSE) which occurs in the field of non-linear optics as a model of wave propagation in fibre optics.

Solution method: The GNLSE is solved by the Interaction Picture method coupled with an embedded Runge–Kutta scheme of order 4. The program includes a costless adaptive step-size control strategy taking advantage of the features of an embedded Runge–Kutta scheme designed for delivering a local error estimate at no extra-cost compared to the standard 4th order Runge–Kutta scheme.

Running time: Highly dependent on the fibre length and accuracy required for the results. Typically between half a minute and several dozens of minutes.

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Correspondence to: UFR de Mathématiques, Université de Rennes 1, Campus de Beaulieu, CS 74205 35042 Rennes, France.

E-mail addresses: stephane.balac@univ-rennes1.fr (S. Balac), afernand@laas.fr (A. Fernandez).

URLs: <http://foton.cnrs.fr> (S. Balac), <http://www.laas.fr> (A. Fernandez).

References:

- [1] M. Frigo and S.G. Johnson. The design and implementation of FFTW3. P IEEE, 2(93):216–231, (2005).
- [2] P Janert. Gnuplot in Action, Understanding Data with Graphs. Manning Publications (2009).

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In optics, the non-linear Schrödinger equation occurs for modelling light-wave propagation into a passive optical fibre. This particular form of the Schrödinger equation is obtained from the general set of Maxwell's equations taking advantage of a certain number of assumptions made possible from the very specific characteristics of (quasi-)monochromatic wave propagation in a medium such as a fibre [1,2]. One of the major assumption, referred as the *slowly varying envelope approximation*, concerns the expression of the electric field in the optical fibre. It assumes that the electric field \mathbf{E} is linearly polarized along a direction \mathbf{e}_x transverse to the direction of propagation \mathbf{e}_z defined by the fibre and can be represented as a function of time τ and position $\mathbf{r} = (x, y, z)$ in a reference frame $(O, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ as

$$\mathbf{E}(\mathbf{r}, \tau) = A(z, t) F(x, y) e^{-i(\omega_0 \tau - kz)} \mathbf{e}_x \quad (1)$$

where the complex valued function A represents the slowly varying optical pulse envelope, F is the electric wave transverse representation, k is the wave number, $\omega_0 = 2\pi\nu_0$ is the wave frequency, and t denotes a local time in a moving frame travelling along with the pulse at the group velocity $v_g = c/n_g$ where n_g is the fibre group index and c the speed of light in vacuum. The relationship between the absolute time τ and the local time t is: $t = \tau - z/v_g$. The expression of the electric wave transverse representation F can sometimes be computed explicitly using the classical method of separation of variables for partial differential equations (PDE). For instance, for circular constant transverse section fibres, it is expressed in terms of Bessel functions [1,2].

In the simplest cases of light-wave propagation in an optical fibre, the evolution of the slowly varying pulse envelope A is governed by the following PDE referred in the literature as the Non-Linear Schrödinger Equation (NLSE) [1,2]

$$\frac{\partial}{\partial z} A(z, t) = -\frac{\alpha}{2} A(z, t) - \frac{i}{2} \beta_2 \frac{\partial^2}{\partial t^2} A(z, t) + i\gamma A(z, t) |A(z, t)|^2. \quad (2)$$

Eq. (2) describes wave propagation in a single mode fibre taking into account phenomena such as the optical Kerr effect through the non-linear coefficient γ , linear attenuation through the linear attenuation coefficient α and linear dispersion through the chromatic dispersion coefficient β_2 .

In a more accurate model of light-wave propagation in an optical fibre, the evolution of the slowly varying pulse envelope A obeys the so-called Generalized Non-Linear Schrödinger Equation (GNLSE) [1,2]

$$\begin{aligned} \frac{\partial}{\partial z} A(z, t) = & -\frac{\alpha}{2} A(z, t) + \left(\sum_{n=2}^{n_{\max}} i^{n+1} \frac{\beta_n}{n!} \frac{\partial^n}{\partial t^n} A(z, t) \right) \\ & + i\gamma \left(1 + \frac{i}{\omega_0} \frac{\partial}{\partial t} \right) \left[A(z, t) \left((1 - f_R) |A(z, t)|^2 \right. \right. \\ & \left. \left. + f_R \int_{-\infty}^{+\infty} h_R(s) |A(z, t - s)|^2 ds \right) \right]. \end{aligned} \quad (3)$$

In Eq. (3) linear dispersion is now taken into account through the dispersion coefficients β_n , $n = 2, \dots, n_{\max}$ whereas non linear dispersion is taken into account through the simplified optical shock parameter $\tau_{\text{shock}} = 1/\omega_0$. Instantaneous Kerr effect manifests itself through the term $(1 - f_R) |A|^2$. The delayed Raman contribution in the time domain is taken into account through the convolution product between the instantaneous power $|A|^2$ and the Raman time response function h_R . The constant f_R represents the fractional contribution of the delayed Raman response to non-linear polarization.

Both evolution type PDE (2) and (3) have to be considered together with the initial condition at $z = 0$

$$\forall t \in \mathbb{R} \quad A(0, t) = a_0(t) \quad (4)$$

where a_0 is a given function and they are solved for all $t \in \mathbb{R}$ and for all $z \in [0, L]$ where L denotes the length of the fibre. The aim of the SPIP program is to solve the GNLSE (3) and the NLSE (2).

For a fixed value of z we denote by $A(z)$ the function $t \in \mathbb{R} \mapsto A(z, t) \in \mathbb{C}$. The same notation will be used for functions of 2 variables (z, t) throughout the paper. For convenience, when dealing with the GNLSE (3) we introduce the linear operator

$$\mathcal{D} : A(z) \mapsto -\frac{1}{2} \alpha A(z) + \sum_{n=2}^{n_{\max}} \beta_n \frac{i^{n+1}}{n!} \partial_t^n A(z) \quad (5)$$

and the non-linear operator:

$$\begin{aligned} \mathcal{N} : A(z) \mapsto & i\gamma \left(1 + \frac{i}{\omega_0} \frac{\partial}{\partial t} \right) \\ & \times \left[A(z) \left((1 - f_R) |A(z)|^2 + f_R (h_R * |A(z)|^2) \right) \right] \end{aligned} \quad (6)$$

where $*$ stands for the convolution product. With these notations, we are concerned by solving the following PDE problem

$$\begin{cases} \frac{\partial}{\partial z} A(z) = \mathcal{D} A(z) + \mathcal{N}(A(z)) & \forall z \in [0, L] \\ A(0) = a_0. \end{cases} \quad (7)$$

The NLSE (2) corresponds to $n_{\max} = 2$ in the definition (5) of \mathcal{D} and to the non-linear operator

$$\mathcal{N} : A(z) \mapsto i\gamma A(z) |A(z)|^2 \quad (8)$$

which is formally retrieved from (6) in the special case when we set $f_R = 0$ and $\frac{i}{\omega_0} = 0$.

The SPIP program solves the PDE problem (7) with \mathcal{N} as given by (6) or (8) by means of the Interaction Picture (IP) method. The main idea of the IP method is a change of unknown in order to transform the GNLSE for the unknown A into a new equation where only remains an explicit reference to the partial derivation with respect to the space variable z and where the time variable t appears as a parameter. This new equation can be solved numerically using the usual methods for ordinary differential equations (ODE) such as the fourth order Runge–Kutta (RK4) method. Then, by using the inverse transform we obtain approximate values for the unknown A at the grid points of a subdivision of the fibre length interval $[0, L]$. This numerical approach is referred to as the

RK4-IP method. The RK4-IP method has been developed by the Bose–Einstein condensate theory group of R. Ballagh from the Jack Dodd Centre at the University of Otago (New Zealand) in the 90s for solving the Gross–Pitaevskii equation which is ubiquitous in Bose condensation. It was described in the Ph.D. thesis of B.M. Caradoc-Davies [3] and M.J. Davis [4]. Latter the IP method has been applied for solving the GNLSE in [5]. A comprehensive mathematical study of the Interaction Picture method for solving the GNLSE has been conducted in [6]. Recently an efficient embedded RK method based on Dormand and Prince RK4(3)-T formula [7] and specifically designed for the IP method has been proposed in [8] to provide a costless adaptive step-size control in the RK4-IP method. It is this method, termed the ERK4(3)-IP method, that is implemented in the SPIP program.

To be comprehensive we may bring to the attention of the reader that a numerical comparison of the IP method and the Symmetric Split-Step method, the most popular method for solving the GNLSE, has been carried out on benchmark problems in optics in [5] and a theoretical comparison of the 2 methods has been achieved in [6]. These investigations demonstrate the supremacy of the IP method over the Symmetric Split-Step method for solving the GNLSE.

The paper is organized as follows. In Section 2 we present how the Interaction Picture method is used to solve the GNLSE. In Section 3 we focus on the way the local error is estimated in order to implement the adaptive step-size control. In Section 4 we show the algorithm implemented in the SPIP program to solve the GNLSE and we give information on the way to run the program. Finally, Section 5 is devoted to numerical experiments carried out with the SPIP program.

2. The interaction picture method

2.1. Overview of the interaction picture method

The interval $[0, L]$ corresponding to the fibre length is divided into K sub-intervals where the spatial grid points are denoted by z_k , $k = \{0, \dots, K\}$ such that $[0, L] = \bigcup_{k=0}^{K-1} [z_k, z_{k+1}]$ where $0 = z_0 < z_1 < \dots < z_{K-1} < z_K = L$. The step length between z_k and z_{k+1} is denoted h_k and we also set $z_{k+\frac{1}{2}} = z_k + \frac{h_k}{2}$.

Solving problem (7) is equivalent to solving the sequence of connected problems $(\mathcal{P}_k)_{k=0, \dots, K-1}$ where

$$(\mathcal{P}_0) \quad \begin{cases} \frac{\partial}{\partial z} A_0(z) = \mathcal{D} A_0(z) + \mathcal{N}(A_0(z)) & \forall z \in [z_0, z_1] \\ A_0(z_0) = a_0 \end{cases}$$

and $\forall k \in \{1, \dots, K-1\}$

$$(\mathcal{P}_k) \quad \begin{cases} \frac{\partial}{\partial z} A_k(z) = \mathcal{D} A_k(z) + \mathcal{N}(A_k(z)) & \forall z \in [z_k, z_{k+1}] \\ A_k(z_k) = A_{k-1}(z_k). \end{cases}$$

Obviously for all $k \in \{0, \dots, K-1\}$ we have

$$\forall z \in [z_k, z_{k+1}] \quad A(z) = A_k(z).$$

Let us consider one of the problems (\mathcal{P}_k) for a value of $k \in \{0, \dots, K-1\}$. Such a problem reads

$$\begin{cases} \frac{\partial}{\partial z} A_k(z) = \mathcal{D} A_k(z) + \mathcal{N}(A_k(z)) & \forall z \in [z_k, z_{k+1}] \\ A_k(z_k) = a_k \end{cases} \quad (9)$$

where $a_k = A_{k-1}(z_k)$ for all $k \in \{1, \dots, K-1\}$. We introduce as new unknown the mapping

$$A_k^{\text{ip}} : (z, t) \in [z_k, z_{k+1}] \times \mathbb{R} \mapsto \exp(-(z - z_{k+\frac{1}{2}})\mathcal{D}) A_k(z, t) \quad (10)$$

where from a mathematical point of view the exponential term has to be understood in the sense of the continuous group generated by the unbounded linear operator \mathcal{D} [9]. From (9), one can show [6] that the new unknown A_k^{ip} is the solution to the following problem over each subinterval $[z_k, z_{k+1}]$

$$(\mathcal{Q}_k) \quad \begin{cases} \frac{\partial}{\partial z} A_k^{\text{ip}}(z) = \mathcal{G}_k(z, A_k^{\text{ip}}(z)) & \forall z \in [z_k, z_{k+1}] \\ A_k^{\text{ip}}(z_k) = \exp(-(z_k - z_{k+\frac{1}{2}})\mathcal{D}) a_k \end{cases} \quad (11)$$

where $\mathcal{G}_k(z, \cdot) = \exp(-(z - z_{k+\frac{1}{2}})\mathcal{D}) \circ \mathcal{N} \circ \exp((z - z_{k+\frac{1}{2}})\mathcal{D})$.

The major interest in doing such a transformation is that compared to problem (7), the new problem (11) for the unknown A_k^{ip} does not involve explicitly partial derivation with respect to the time variable t anymore. Partial derivation with respect to time now occurs through the operator $\exp(\pm(z - z_{k+\frac{1}{2}})\mathcal{D})$. Problem (11) can be numerically solved using a standard quadrature scheme for ordinary differential equations such as a Runge–Kutta scheme [10] as detailed in Section 2.2.

Of course, in order to get the initial condition for problem (11), one has to compute the mapping $\exp(-(z_k - z_{k+\frac{1}{2}})\mathcal{D}) a_k$. This mapping coincides with the solution for $z = z_{k+\frac{1}{2}}$ to the following linear PDE problem [6]

$$\begin{cases} \frac{\partial}{\partial z} A_k(z) = \mathcal{D} A_k(z) & \forall z \in [z_k, z_{k+\frac{1}{2}}] \\ A_k(z_k) = a_k. \end{cases} \quad (12)$$

Once computed the solution to problem (11), one has to use the inverse mapping of (10) to obtain $A_k(z_{k+1})$ which is the initial condition for problem (\mathcal{P}_{k+1}) . The function

$A_k(z_{k+1}) = \exp((z_{k+1} - z_{k+\frac{1}{2}})\mathcal{D}) A_k^{\text{ip}}(z_{k+1})$ corresponds to the solution for $z = z_{k+1}$ to the following linear PDE problem [6]

$$\begin{cases} \frac{\partial}{\partial z} A_k(z) = \mathcal{D} A_k(z) & \forall z \in [z_{k+\frac{1}{2}}, z_{k+1}] \\ A_k(z_k) = A_k^{\text{ip}}(z_{k+1}). \end{cases} \quad (13)$$

The 2 linear PDE problems (12) and (13) can be numerically solved very efficiently by using the Fourier Integral Transform as shown in Section 2.3. In the next section we first study the way of solving the non-linear ODE problem (11).

2.2. Use of an embedded rk scheme for solving problem (11)

The solution to the non-linear problem (11) can be approximated by standard numerical schemes for ordinary differential equations (ODE). Because it represents a good compromise between accuracy and computational cost, we have selected the classical fourth order Runge–Kutta (RK4) scheme. Moreover the values of its internal quadrature nodes $(0, \frac{1}{2}, \frac{1}{2}, 1)$ imply by symmetry some cancellations in the terms to be computed compared to other 4th order RK schemes when used in conjunction with the IP method, thus reducing the global computational cost of the method. So as to dispose of a costless local error estimator for adaptive step-size purposes, we consider the Embedded Runge–Kutta (ERK) pair schemes of order 3 and 4 defined by the following Butcher extended tableau [8]

0					
$\frac{1}{2}$	$\frac{1}{2}$				
$\frac{1}{2}$	0	$\frac{1}{2}$			
1	0	0	1		
1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{15}$	$\frac{1}{10}$

(14)

where the grey cells correspond to Butcher tableau for the standard RK4 method and the whole array is the Butcher tableau for the RK3 method. This ERK4(3) scheme is referred in the literature as the Dormand and Prince *Runge–Kutta 4(3) T* scheme [7].

Embedded Runge–Kutta schemes [10,11] are special RK schemes designed to deliver two approximations of the solution of the ODE problem under consideration, corresponding to 2 different convergence orders. These 2 approximations of the solution can be considered as an accurate approximate solution (here the one computed with the 4th order RK scheme) and a coarse approximate solution (the one computed with the 3rd order RK scheme). Combined in a specific way they deliver an estimation of the local error committed while approaching the solution with the lower order method as detailed in Section 3.1.

For $k \in \{0, \dots, K-1\}$, let $u_k^{\text{ip},[3]}$ (resp. $u_k^{\text{ip},[4]}$) denote the approximate solution to problem (11) resulting from the use of the 3rd order (resp. 4th order) RK method. The 4th order accurate solution is given by

$$u_k^{\text{ip},[4]}(z_{k+1}) = A_k(z_k) + h_k \Phi_4(z_k, A_k(z_k); h_k) \quad (15)$$

where the mapping Φ_4 is the *increment function* [10] of the RK4 method defined according to *Butcher Tableau* (14) as

$$\Phi_4(z_k, A_k(z_k); h_k) = \frac{1}{6} (\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4)$$

with

$$\begin{aligned} \alpha_1 &= \mathcal{G}_k(z_k, A_k(z_k)) = \exp\left(\frac{h_k}{2} \mathcal{D}\right) \mathcal{N}\left(\exp\left(-\frac{h_k}{2} \mathcal{D}\right) A_k(z_k)\right) \\ \alpha_2 &= \mathcal{G}_k\left(z_k + \frac{h_k}{2}, A_k(z_k) + \frac{h_k}{2} \alpha_1\right) = \mathcal{N}\left(A_k(z_k) + \frac{h_k}{2} \alpha_1\right) \\ \alpha_3 &= \mathcal{G}_k\left(z_k + \frac{h_k}{2}, A_k(z_k) + \frac{h_k}{2} \alpha_2\right) = \mathcal{N}\left(A_k(z_k) + \frac{h_k}{2} \alpha_2\right) \\ \alpha_4 &= \mathcal{G}_k(z_k + h_k, A_k(z_k) + h_k \alpha_3) \\ &= \exp\left(-\frac{h_k}{2} \mathcal{D}\right) \mathcal{N}\left(\exp\left(\frac{h_k}{2} \mathcal{D}\right) [A_k(z_k) + h_k \alpha_3]\right). \end{aligned}$$

Similarly the 3rd order accurate solution is

$$u_k^{\text{ip},[3]}(z_{k+1}) = A_k(z_k) + h_k \Phi_3(z_k, A_k(z_k); h_k) \quad (16)$$

where the mapping Φ_3 is the *increment function* of the RK3 method defined according to *Butcher Tableau* (14) as

$$\Phi_3(z_k, A_k(z_k); h_k) = \frac{1}{6} (\alpha_1 + 2\alpha_2 + 2\alpha_3) + \frac{1}{15} \alpha_4 + \frac{1}{10} \alpha_5$$

with

$$\begin{aligned} \alpha_5 &= \mathcal{G}_k\left(z_k + h, A_k(z_k) + \frac{h_k}{6} (\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4)\right) \\ &= \mathcal{G}_k(z_{k+1}, u^{\text{ip},[4]}(z_{k+1})) \\ &= \exp\left(-\frac{h_k}{2} \mathcal{D}\right) \mathcal{N}\left(\exp\left(\frac{h_k}{2} \mathcal{D}\right) u^{\text{ip},[4]}(z_{k+1})\right). \end{aligned}$$

Using the change of unknown (10) back we deduce the following 2 approximations of the solution A_k to problem (\mathcal{P}_k) at grid point z_{k+1}

$$\begin{aligned} u_{k+1}^{[4]} &= \exp\left(\frac{h_k}{2} \mathcal{D}\right) \left[A_k(z_k) + \frac{h_k}{6} (\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4)\right] \\ u_{k+1}^{[3]} &= \exp\left(\frac{h_k}{2} \mathcal{D}\right) \left[A_k(z_k) + \frac{h_k}{30} (5\alpha_1 + 10\alpha_2 + 10\alpha_3 + 2\alpha_4 + 3\alpha_5)\right] \end{aligned} \quad (17)$$

where $u_{k+1}^{[3]}$ (resp. $u_{k+1}^{[4]}$) denotes the approximate solution to problem (\mathcal{P}_k) at grid point z_{k+1} resulting from the use of the 3rd order (resp. 4th order) RK method.

Actually we are only interested in computing the approximate solutions to problem (9) given by (17) and the use of the new unknown A_k^{ip} is a go-between in the computational approach. Therefore we recast the above computational procedure in the following way in order to reduce the cost of the method. For $k \in \{0, \dots, K-1\}$, step k of the ERK4(3)-IP method reads:

$$\begin{aligned} u_k^{\text{ip}}(z_k) &= \exp\left(\frac{h_k}{2} \mathcal{D}\right) A_{k-1}(z_k) \\ \alpha_1 &= \exp\left(\frac{h_k}{2} \mathcal{D}\right) \alpha'_{5,k} \\ \alpha_2 &= \mathcal{N}\left(u_k^{\text{ip}}(z_k) + \frac{h_k}{2} \alpha_1\right) \\ \alpha_3 &= \mathcal{N}\left(u_k^{\text{ip}}(z_k) + \frac{h_k}{2} \alpha_2\right) \\ \alpha'_4 &= \mathcal{N}\left(\exp\left(\frac{h_k}{2} \mathcal{D}\right) [u_k^{\text{ip}}(z_k) + h_k \alpha_3]\right) \\ \beta &= \exp\left(\frac{h_k}{2} \mathcal{D}\right) \left[u_k^{\text{ip}}(z_k) + \frac{h_k}{6} (\alpha_1 + 2\alpha_2 + 2\alpha_3)\right] \\ u_{k+1}^{[4]} &= \beta + \frac{h_k}{6} \alpha'_4 \\ \alpha'_{5,k+1} &= \mathcal{N}(u_{k+1}^{[4]}) \\ u_{k+1}^{[3]} &= \beta + \frac{h_k}{30} (2\alpha'_4 + 3\alpha'_{5,k+1}). \end{aligned} \quad (18)$$

We refer to Section 2.4 for details in the way the non-linear operator \mathcal{N} is computed and to Section 3.1 for indications in the way the local error at step k is estimated from the knowledge of $u_{k+1}^{[3]}$ and $u_{k+1}^{[4]}$. It is worth mentioning that compared to the standard RK4 scheme the over-cost of using the ERK4(3) scheme as given by (18) is low. In both cases, the number of evaluations of the non-linear operator \mathcal{N} is 4 and we have 4 evaluations of the $\exp(\frac{h_k}{2} \mathcal{D})$ operator. The extra cost per step is limited to 2 additions and 3 multiplications and the need to keep in memory 2 intermediate results.

2.3. Solving the linear PDE problems (12) and (13)

Let us now examine how problems (12) and (13) are solved or actually how the terms involving the operator $\exp(\frac{h_k}{2} \mathcal{D})$ in the computational sequence (18) are obtained. The 2 problems (12) and (13) are in the form

$$\begin{cases} \frac{\partial}{\partial z} U(z) = \mathcal{D} U(z) & \forall z \in [a, b] \\ U(a) = \varphi \end{cases} \quad (19)$$

where $U(z) : t \in \mathbb{R} \mapsto U(z, t) \in \mathbb{C}$ and the initial condition φ is a known data.

The solution to problem (19) can be computed by using the Fourier Integral Transform (FIT) approach. For a fixed value of the space variable z , let $\hat{U}(z)$ be the FIT of $U(z)$ with respect to the time variable t :

$$\forall v \in \mathbb{R} \quad \hat{U}(z, v) = \int_{-\infty}^{+\infty} U(z, t) e^{2i\pi vt} dt.$$

From (19) we deduce that \hat{U} satisfies

$$\begin{cases} \frac{\partial}{\partial z} \hat{U}(z) = \hat{a}_v \hat{U}(z) & \forall z \in [a, b] \\ \hat{U}(a) = \hat{\varphi} \end{cases} \quad (20)$$

where $\widehat{d}_v = -\frac{\alpha}{2} + i \sum_{n=2}^N \frac{\beta_n}{n!} (2\pi v)^n$. The solution to the linear first order ODE problem (20) is

$$\widehat{U}(z) = \widehat{\varphi} e^{\widehat{d}_v(z-a)}. \quad (21)$$

We deduce that $U(b) = \mathcal{F}^{-1}[\widehat{U}(b)] = \mathcal{F}^{-1}[\widehat{\varphi} e^{\widehat{d}_v(b-a)}]$ where \mathcal{F}^{-1} denotes the inverse Fourier Transform operator. As a consequence, the solution to problem (12) at grid point $z_{k+\frac{1}{2}}$ reads

$$A_k^+(z_{k+\frac{1}{2}}) = \exp\left(\frac{h}{2} \mathcal{D}\right) A_{k-1}(z_k) = \mathcal{F}^{-1}\left[e^{\widehat{d}_v \frac{h}{2}} \widehat{A}_{k-1}(z_k)\right]$$

and the solution to problem (13) at grid point z_{k+1} reads

$$A_k^-(z_{k+1}) = \exp\left(\frac{h}{2} \mathcal{D}\right) A_k^{\text{ip}}(z_{k+1}) = \mathcal{F}^{-1}\left[e^{\widehat{d}_v \frac{h}{2}} \widehat{A}_k^{\text{ip}}(z_{k+1})\right].$$

In the computational sequence (18) all the terms involving the $\exp(\frac{h}{2} \mathcal{D})$ operator can be computed in a similar way.

2.4. Computation of the non-linear operator

The non-linear operator \mathcal{N} defined in (6) can be efficiently evaluated by means of the FIT since in the frequency domain time derivation of functions amounts for multiplying the FIT of the function by a factor $-2i\pi v$. Namely for any given function $g : t \in \mathbb{R} \mapsto g(t) \in \mathbb{C}$, we have $\forall v \in \mathbb{R}$

$$\widehat{\mathcal{N}(g)}(v) = i\gamma \left(1 + \frac{v}{v_0}\right) \times \mathcal{F}\left[(1 - f_R)g|g|^2 + f_R g (h_R * |g|^2)\right](v)$$

where \mathcal{F} denotes the Fourier Transform operator. Moreover, using the properties of the FIT with respect to convolution we may compute the term $h_R * |g|^2$ as follows

$$h_R * |g|^2 = \mathcal{F}^{-1}[\widehat{h_R} \times \widehat{|g|^2}] \quad (22)$$

where $\widehat{h_R}$ and $\widehat{|g|^2}$ denote respectively the FIT of h_R and $|g|^2$. Therefore, computation of $\mathcal{N}(g)$ can be achieved in the following steps:

- compute the FIT $\widehat{h_R}$ and $\widehat{|g|^2}$ of h_R and $|g|^2$;
- multiply these 2 mappings and compute the inverse FIT of the result to obtain convolution product $h_R * |g|^2$;
- compute the FIT of $g((1 - f_R)|g|^2 + f_R(h_R * |g|^2))$;
- multiply the result by the mapping $v \mapsto i\gamma(1 + v/v_0)$;
- compute the inverse FIT of this last product to get $\mathcal{N}(g)$.

It is worth mentioning that the evaluation of the non-linear operator \mathcal{N} requires 3 Fourier Transform and 2 inverse Fourier Transform evaluations. The `SPiP` program uses the `FFTW` library for computing the Fourier transforms [12]. `FFTW` for "Fastest Fourier Transform in the West" is a software library for computing discrete Fourier transforms (DFT) developed by Matteo Frigo and Steven G. Johnson at the Massachusetts Institute of Technology. `FFTW` is known as the fastest free software implementation of the Fast Fourier transform (FFT) algorithm. It can compute DFT of real-valued and complex-valued arrays of arbitrary size n with a complexity in $\mathcal{O}(n \log(n))$.

The non-linear operator \mathcal{N} defined by (8) for the NLSE can be computed in a similar but simpler way.

3. Adaptive step-size control in the ERK4(3)-IP method

3.1. Local error estimate

Since the IP method is based on the change of unknown defined in (10), the only approximation in the ERK4(3)-IP method is related to the ERK4(3) scheme used to solve the non-linear ODE

problem (11). For $k \in \{0, \dots, K-1\}$, let $u_k^{[3]}$ (resp. $u_k^{[4]}$) denotes the approximate value of the slowly varying optical pulse envelope A at grid point z_k delivered by the 3rd order (resp. the 4th order) RK scheme. Assuming that the solution value at grid point z_k is regarded as exact (because we are concerned by an estimation of the local error), the local error at grid point z_{k+1} and time t for the 3rd order (resp. 4th order) RK schemes used for solving (11) are respectively given by [10,11]

$$\begin{aligned} \ell_{k+1}^{[3]}(t) &= A(z_{k+1}, t) - u_{k+1}^{[3]}(t) = \psi_3(t, z_k, u_k^{[3]}) h_k^4 + \mathcal{O}(h_k^5) \\ \ell_{k+1}^{[4]}(t) &= A(z_{k+1}, t) - u_{k+1}^{[4]}(t) = \psi_4(t, z_k, u_k^{[4]}) h_k^5 + \mathcal{O}(h_k^6) \end{aligned} \quad (23)$$

where ψ_3 (resp. ψ_4) are functions of the elementary differentials of order 3 (resp. 4) of g_k . By difference of these 2 relations we obtain

$$u_{k+1}^{[4]}(t) - u_{k+1}^{[3]}(t) = \psi_3(t, z_k, u_k^{[3]}) h_k^4 + \mathcal{O}(h_k^5).$$

Thus the local error for the 3rd order RK method at grid point z_{k+1} can be approximated, with an error in $\mathcal{O}(h_k^5)$, in the following way

$$\ell_{k+1}^{[3]}(t) \approx \psi_3(t, z_k, u_k^{[3]}) h_k^4 + \mathcal{O}(h_k^5) \approx u_{k+1}^{[4]}(t) - u_{k+1}^{[3]}(t). \quad (24)$$

The \mathbb{L}^2 -local error at grid point z_{k+1} is computed as follows

$$\begin{aligned} L_{k+1}^{[3]} &= \|\ell_{k+1}^{[3]}\|_{\mathbb{L}^2(\mathbb{R}, \mathbb{C})} = \|\widehat{\ell_{k+1}^{[3]}}\|_{\mathbb{L}^2(\mathbb{R}, \mathbb{C})} \\ &\approx \left(\int_{\mathbb{R}} \left| \widehat{u}_{k+1}^{[4]}(v) - \widehat{u}_{k+1}^{[3]}(v) \right|^2 dv \right)^{\frac{1}{2}} \\ &\approx \left(h_v \sum_{j=0}^{J-1} \left| \widehat{u}_{k+1}^{[4]}(v_j) - \widehat{u}_{k+1}^{[3]}(v_j) \right|^2 \right)^{\frac{1}{2}} \end{aligned} \quad (25)$$

where $(v_j)_{j=0, \dots, J}$ denotes a constant step-size h_v sampling of the observed spectral window and the last approximation results from the use of the left rectangle quadrature rule.

Actually, even if the local error estimate (24) holds only for the 3rd order method, in practice we use the value given by the 4th order method, which is more accurate, as the approximation of the solution at grid point z_{k+1} . This approach, referred in the literature [10,11] as the *local extrapolation mode* for ERK methods, slightly overestimates the actual local error, which is safe but actually not fully optimal.

3.2. Step-size control

For step-size control, a tolerance "tol" is given as bound on the local error estimate. A step-size control strategy [10] consists in rejecting the current step-size if it gives an estimated local error higher than the specified tolerance; in accepting the solution computed with this step-size otherwise. There are 2 criteria usually employed for step-size control purposes. The criterion of error per step (EPS) selects the step size h_k at each step so that the local error is lower than the tolerance tol whereas the criterion of error per unit step (EPUS) selects the step size h_k at each step so that the local error is lower than $\text{tol} \times h_k$. It is clear that for sufficiently small tolerance value EPUS criterion selects a smaller step-size than EPS criterion. In the `SPiP` program the EPS criterion is used. When the current step-size is rejected, a new smaller step-size has to be chosen to recompute the solution over the current step. As well, when the current step-size meets the tolerance requirement for the local error it has to be scaled up for the next step computations. When we assume that the leading term in the asymptotic expansion (24) of the local error dominates, then from (24) and (25) there exists $C > 0$ such that

$$L_{k+1}^{[3]} = \|\ell_{k+1}^{[3]}\|_{\mathbb{L}^2(\mathbb{R}, \mathbb{C})} = C h_k^4.$$

The optimal step-size h_{opt} is the one for which the local error estimate $L_{k+1}^{[3]}$ is the closest to the prescribed tolerance tol , i.e. $C h_{\text{opt}}^4 = \text{tol}$. By eliminating the constant C from these 2 relations we obtain

$$h_{\text{opt}} = h_k \sqrt[4]{\frac{\text{tol}}{L_{k+1}^{[3]}}}.$$

In the previous relationship it is common to consider an estimate for the relative local error and a relative tolerance “tol”. It is the choice we adopt in the SPIP program.

For robustness the step-size control has to be designed in order to respond as smoothly as possible to real or apparent abrupt changes in behaviour. This means that the step-size should not vary from one step to the next by an excessive ratio. That is the reason why we impose that the new step-size does not exceed twice the current step-size above and half the current step-size below. Moreover, in order to avoid situations where the specified tolerance is ever exceeded resulting in rejecting too many steps, a safety factor is introduced. Namely, if h_{opt} is the value of the step-size estimated to give a predicted truncation error equal to the tolerance, then the smaller value $0.9h_{\text{opt}}$ for instance is used instead.

Following these requirements, we use the following step-size control formula in the SPIP program

$$h_{\text{new}} = \max \left(0.5, \min \left(2.0, 0.9 \sqrt[4]{\frac{\text{tol}}{\text{err}}} \right) \right) h_k \quad (26)$$

where “tol” denotes the tolerance value specified by the user as a bound on the relative local error and “err” denotes the estimation of the relative local error for the current step deduced from (25). The constant values 0.9, 0.5 and 2.0 are somewhat arbitrary and have to be regarded as design parameters.

4. The SPIP program

4.1. Algorithm for the ERK4(3)-IP method with step-size control

Since the evaluation of the non-linear operator \mathcal{N} and the $\exp(\frac{h_k}{2} \mathcal{D})$ operator both involve the use of Fourier Transforms, it is convenient to slightly modify the computational sequence (18) in order to save some Fourier Transforms by optimizing the occurrence of successive FT and inverse FT. This leads to the following algorithm for the ERK4(3)-IP method.

ERK4(3)-IP algorithm

Input: - Array $[t_j]_{j=1,\dots,J}$ contains the time sampling points
 - Array u contains the incident slowly varying pulse envelope sampled in time
 - Array $[v_j]_{j=1,\dots,J}$ contains the frequency sampling points
 - Real number α : linear attenuation coefficient
 - Array $[\beta_n]_{n=2,\dots,n_{\text{max}}}$ contains the dispersion coefficients
 - Real number tol : the tolerance value for the local error
 - Real number h_{initial} : the initial step size value

Output: Array u contains the outgoing slowly varying pulse envelope at fibre end sampled in time

```

1: {Initialisation}
2: for  $j = 1, \dots, J$  do
3:    $\hat{d}[j] \leftarrow -\frac{1}{2}\alpha + i \sum_{n=2}^{n_{\text{max}}} \frac{\beta_n}{n!} (2\pi v_j)^n$ 
4: end for
5:  $\hat{u} \leftarrow \text{FFT}(u, \text{forward})$ 
6:  $\hat{Nu} \leftarrow \text{COMPUTE\_FTN}(u, \hat{u})$ 
7:  $z_k = 0, h = h_{\text{initial}}$ 
8: {Loop over the propagation sub-interval}
9: while  $z_k < L$  do
```

```

10: for  $j = 1, \dots, J$  do
11:    $\hat{e}[j] \leftarrow \exp(\frac{h}{2}\hat{d}[j])$ 
12:    $\hat{u}_{\text{ip}}[j] \leftarrow \hat{e}[j] \times \hat{u}[j]$ 
13: end for
14: for  $j = 1, \dots, J$  do
15:    $\hat{\alpha}_1[j] \leftarrow \hat{e}[j] \times \hat{Nu}[j]$ 
16:    $\hat{u}_2[j] \leftarrow \hat{u}_{\text{ip}}[j] + \frac{h}{2}\hat{\alpha}_1[j]$ 
17: end for
18:  $u_2 \leftarrow \text{FFT}(\hat{u}_2, \text{backward})$ 
19:  $\hat{\alpha}_2 \leftarrow \text{COMPUTE\_FTN}(u_2, \hat{u}_2)$ 
20: for  $j = 1, \dots, J$  do
21:    $\hat{u}_3[j] \leftarrow \hat{u}_{\text{ip}}[j] + \frac{h}{2}\hat{\alpha}_2[j]$ 
22: end for
23:  $u_3 \leftarrow \text{FFT}(\hat{u}_3, \text{backward})$ 
24:  $\hat{\alpha}_3 \leftarrow \text{COMPUTE\_FTN}(u_3, \hat{u}_3)$ 
25: for  $j = 1, \dots, J$  do
26:    $\hat{u}_4[j] \leftarrow \hat{e}[j] \times (\hat{u}_{\text{ip}}[j] + h\hat{\alpha}_3[j])$ 
27: end for
28:  $u_4 \leftarrow \text{FFT}(\hat{u}_4, \text{backward})$ 
29:  $\hat{\alpha}_4 \leftarrow \text{COMPUTE\_FTN}(u_4, \hat{u}_4)$ 
30: for  $j = 1, \dots, J$  do
31:    $\hat{r}[j] \leftarrow \hat{e}[j] \times (\hat{u}_{\text{ip}}[j] + \frac{h}{6}\hat{\alpha}_1[j] + \frac{h}{3}\hat{\alpha}_2[j] + \frac{h}{3}\hat{\alpha}_3[j])$ 
32:    $\hat{u}_1[j] \leftarrow \hat{r}[j] + \frac{h}{6}\hat{\alpha}_4[j]$  {RK4 solution}
33: end for
34:  $u_1 \leftarrow \text{FFT}(\hat{u}_1, \text{backward})$ 
35:  $\hat{\alpha}_5 \leftarrow \text{COMPUTE\_FTN}(u_1, \hat{u}_1)$ 
36: for  $j = 1, \dots, J$  do
37:    $\hat{v}_1[j] \leftarrow \hat{r}[j] + \frac{h}{30}(2\hat{\alpha}_4[j] + 3\hat{\alpha}_5[j])$  {RK3 solution}
38: end for
39: {Step-size control}
40:  $\text{err} \leftarrow 0$  [ $L^2$  error]
41: for  $j = 1, \dots, J-1$  do
42:    $\text{err} \leftarrow \text{err} + |\hat{u}_1[j] - \hat{v}_1[j]|^2$ 
43: end for
44:  $\text{err} \leftarrow \sqrt{h_t \text{err}}$  {see Formula (25)}
45:  $h_{\text{opt}} = \max \left( 0.5, \min \left( 2.0, 0.9 \sqrt[4]{\frac{\text{tol}}{\text{err}}} \right) \right) h$  {Optimal step-size for the given prescribed tolerance, see Formula (26)}
46: if  $\text{err} \leq \text{tol}$  then
47:   {the current local error matches the tolerance}
48:    $z_k = z_k + h$  {New grid point is confirmed}
49:    $h = \min(h_{\text{opt}}, L - z_k)$  {New step-size value}
50:    $u \leftarrow u_1$  {Array  $u$  contains the time sampled values  $[A_k(z_k, t_j)]_{j=1,\dots,J}$  of the pulse envelope at grid point  $z_k$ }
51:    $\hat{u} \leftarrow \hat{u}_1$  {Update}
52:    $\hat{Nu} \leftarrow \hat{\alpha}_5$ 
53: else
54:   {The current local error does not match the tolerance}
55:    $h = h_{\text{opt}}$  {New computation from  $z_k$  with smaller step-size  $h_{\text{opt}}$  is necessary}
56: end if
57: end while
```

$\text{FFT}(u, \text{forward})$ stands for a call to the Fast Fourier Transform (FFT) algorithm to compute the Discrete Fourier Transform (DFT) of array u , whereas $\text{FFT}(u, \text{backward})$ stands for a call to FFT algorithm to compute the inverse DFT of array u . COMPUTE_FTN refers to the following function.

FUNCTION $\hat{g} = \text{COMPUTE_FTN}(f, \hat{f})$

Compute the Fourier Transform of $g : t \mapsto \mathcal{N}(f)(z, t)$ for a given z where \mathcal{N} refers to the non-linear operator defined by (6) for the GNLSE and by (8) for the NLSE.

Input: - Array f contains the time sampling of function f for the given z
 - Array \hat{f} contains the sampled FT of f
 - Array $[v_j]_{j=1,\dots,J}$ contains the frequency sampling points

- Real number ω_0 : wave frequency of the electric field
- Real number γ : non-linear coefficient
- Array h_R containing the sampling of the FIT of the Raman response function
- Real number f_R : fractional contribution of the delayed Raman response to the non-linear polarization

Output: Array \hat{g} contains the sampled FIT of g

```

1: if GNLSE then
2:   {GNLSE case}
3:   for j = 1, ..., J do
4:      $op_1[j] \leftarrow |f[j]|^2$ 
5:   end for
6:    $\widehat{op}_1 \leftarrow \text{FFT}(op_1, \text{forward})$ 
7:   for j = 1, ..., J do
8:      $\widehat{op}_2[j] \leftarrow \widehat{op}_1[j] \times \widehat{h}_R[j]$ 
9:   end for
10:   $op_2 \leftarrow \text{FFT}(\widehat{op}_2, \text{backward})$  {Array  $op_2$  contains the
    convolution product  $h_R * |f|^2$ }
11:  for j = 1, ..., N do
12:     $op_3[j] \leftarrow f[j] \times ((1 - f_R) op_1[j] + f_R op_2[j])$ 
13:  end for
14:   $\widehat{op}_3 \leftarrow \text{FFT}(op_3, \text{forward})$ 
15:  for j = 1, ..., J do
16:     $\widehat{g}[j] \leftarrow i\gamma(1 + \frac{2\pi v_j}{\omega_0}) \widehat{op}_3[j]$ 
17:  end for
18: else
19:   {NLSE case}
20:   for j = 1, ..., J do
21:      $op_1[j] \leftarrow f[j] \times |f[j]|^2$ 
22:   end for
23:    $\widehat{g} \leftarrow \text{FFT}(op_1, \text{forward})$ 
24: end if

```

When the solution is required at every grid points it suffices to modify line 50 of the ERK4(3)-IP algorithm to record or draw the solution u .

As mentioned earlier, the main cost of the algorithm lies in the number of evaluations of the non-linear operator \mathcal{N} which is 4 per step and to a lower degree in the 4 evaluations of the $\exp(\frac{h_k}{2}\mathcal{D})$ operator. In total, it amounts for 16 FFT per step.

4.2. Installation instructions

The SPIP program requires 2 external libraries:

- FFTW a C subroutine library for computing the discrete Fourier transform (DFT) developed by Matteo Frigo and Steven G. Johnson, see [12] and <http://www.fftw.org>
- Gnuplot a portable command-line driven graphing utility, see [14] and <http://www.gnuplot.info>. Display requests to a Gnuplot session from SPIP is achieved thanks to the `gnuplot_i` module developed by Nicolas Devillard, see <http://ndevilla.free.fr/gnuplot/>.

4.2.1. Under linux

Unzip the SPIP archive `spip-1.1.zip` in a suitable location of the home directory by typing in a shell window:

```
$ unzip spip-1.1.zip
```

Go to the Linux directory:

```
$ cd spip-1.1/linux
```

Compile the C program thanks to the Makefile available in the directory:

```
$ make
```

The executable is named `spipxx`. Optionally, you can delete object files by typing:

```
$ make clean
```

4.2.2. Under windows

You can use the Unix-like environment and command-line interface Cygwin for Microsoft Windows (<http://cygwin.com>) and proceed pretty much as for the Linux installation. Alternatively, you can use a C language IDE (e.g. Code::Blocks, <http://www.codeblocks.org>) to compile the sources. A Code::Blocks project `spip_windows.cbp` is available in the archive under the directory `windows` for convenience.

Note that display requests to a Gnuplot session from SPIP are not available under Windows. It is however possible to display the slowly varying optical pulse envelope using the Matlab/Octave scripts provided in the archive, see Section 4.4 for details.

4.3. Running the program

Under Linux, the SPIP program is launched by typing in a shell window:

```
$ ./spipxx
```

Under Windows, double-click on the `spipxx` program icon located in the Windows directory of the SPIP archive to launch the program.

In the command line user interface, the user is first asked to provide the name and path of the folder where the result files will be stored.

```
[?] Directory where result files will be stored: res
```

Typing a carriage return key will set this directory to the current directory by default. If the folder does not exist, the program will end to let the user create it.

Then the user is asked to provide a keyword for the present simulation. The result files generated by the SPIP program will automatically bear this keyword in their name with an additional unique key corresponding to the elapsed time in second from Epoch.

```
[?] Keyword for the present simulation: green
```

Then the type of equation to solve must be specified: Non-Linear Schrödinger Equation or Generalized Non-Linear Schrödinger Equation.

```
[?] Equation type [0 for NLS or 1 for GNLS] = 1
```

Four pre-defined incident shapes for the slowly varying electric field pulse envelope are available: Soliton, Gaussian and Super-Gaussian, Hyperbolic Secant, see [1]. The mathematical expression for these pulses are given in the SPIP documentation available in the SPIP archive. Additionally, it is possible to provide any other incident pulse shape in the program. To do so, one must append the expression of the incident pulse in file `inpulse_shape.c` in the space provided and one must recompile the program.

```
[?] Shape of the incident pulse envelope
[1: Soliton, 2: Gaussian, 3: Hyperbolic Secant,
 4: Super-Gaussian, 5: user-defined] = 2
```

The next stage consists in providing the incident pulse envelope features. The items may vary depending on the selected shape. For the Gaussian pulse they are: the wavelength of the pulse, its Peak Power, the value of the chirp constant and the half-width at 1/e-intensity point of the pulse.

```
[?] Central wavelength of the pulse [nm] = 1064
[?] Peakpower [W] = 100
[?] Chirp constant = 0
[?] Width parameter of the pulse [ps]: T0 = 2.8365
```

It is followed by the features of the optical fibre, i.e. the values of the parameters involved in equation (2) or (3). For the GNLSE they are: fibre length L , linear loss/gain coefficient α , number n_{\max} of non-zero dispersion coefficients and values of these coefficients $\beta_2, \dots, \beta_{n_{\max}}$, non-linear coefficient γ and fractional contribution of the delayed Raman response f_R .

```
[?] Fibre length [km] = 96.77e-3
[?] Linear loss/gain coefficient alpha [km^-1] = 0.046
[?] Number of non-zeros beta_k (k>=2) coefficients = 2
[?] Value of beta_2 [ps^2 km^-1] = 19.83
[?] Value of beta_3 [ps^3 km^-1] = 0.031
[?] non-linear coefficient gamma [W^-1 km^-1] = 4.3
[?] Fractional contribution of the delayed Raman
    response: fr = 0.245
```

Note that the expression of the Raman time response function h_R provided in [1] is used in the `spip` program but this function can be easily modified in file `hraman.c` (the program must then be re-compiled).

In a last stage, the user must provide the values of the tuning parameters of the ERK4(3)-IP method: size of the time window and number of sampling points for the FFT as well as the initial step-size and tolerance for the adaptive step-size method.

```
[?] Size of the time window; winT [ps] = 50
[?] Number of time sampling points 2^p with p = 14
[?] Initial step-size length [m] = 0.1
[?] Tolerance value = 1e-6
```

Before computations start, the variation with respect to time of the modulus, real and imaginary parts of the slowly varying pulse envelope $A(z = 0, t)$ at the fibre entrance are depicted in a Gnuplot graphics window (under Linux only). In a second window the Fourier Transform of these quantities are depicted, see Fig. 1. The 2 figures are automatically recorded in the results directory in PNG format into 2 files named `in_spec_xxx.png` and `in_time_xxx.png` respectively, where `xxx` stands for the key label of the simulation.

During the computations, the current step number, the current step-size and the progress of the calculations are displayed on the console window.

Once the computations are achieved, the time variation of the modulus, real and imaginary parts of the slowly varying pulse envelope $A(L, t)$ at the fibre end are depicted in a Gnuplot graphics window (under Linux only). In a second graphics window the Fourier Transform of the modulus, real and imaginary parts of the slowly varying electric field pulse envelope $A(L, t)$ at the fibre end are depicted, see Fig. 2. The 2 figures are also automatically recorded in the results directory into 2 files in PNG format named `out_spec_xxx.png` and `out_time_xxx.png` respectively, where `xxx` stands for the label key of the simulation. Additionally, the L^2 norm, the L^1 norm and the L^∞ norm of the solution are computed. They are respectively defined by:

$$\|A(L)\|_2 = \left(\int_{\mathbb{R}} |A(L, t)|^2 dt \right)^{\frac{1}{2}}, \quad \|A(L)\|_1 = \int_{\mathbb{R}} |A(L, t)| dt,$$

$$\|A(L)\|_\infty = \sup_{t \in \mathbb{R}} |A(L, t)|. \quad (27)$$

```
Dispersion length = 4.057354e-01 km
Non-linear length = 2.325581e-03 km
Number of spatial steps = 300
```

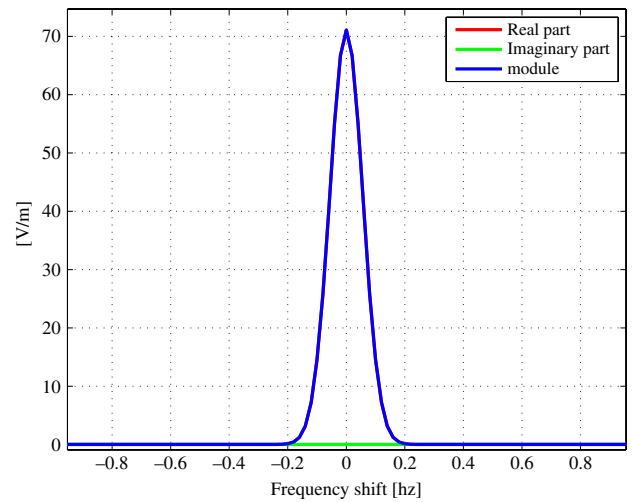
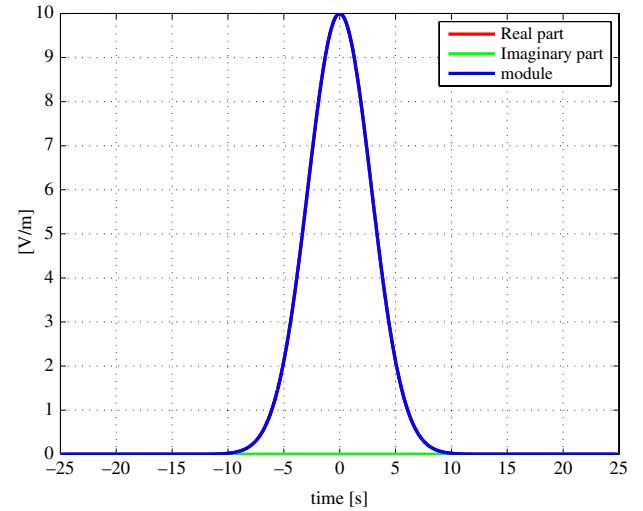


Fig. 1. Screen-shot of the Gnuplot graphics windows where are depicted the time variation of the modulus, real and imaginary parts of the slowly varying pulse envelope $A(0, t)$ at the fibre entrance (top figure) and the Fourier Transform of these quantities (bottom figure, zoom on the area of interest).

```
L^2-norm of the solution at fibre end = 2.237221e+01
L^1-norm                             = 1.161340e+02
L^infinity-norm                       = 4.979086e+00
```

It is also possible to record the time evolution of the solution along the fibre by setting the variable `REC` to the value 1 in the `spip.h` header file. Note that this option is not set by default since it increases the computational time and produces a very large file on the disc (e.g. around 78 Mo for the present simulation).

At the end of the simulation, the directory specified to store the results files contains the following files (`xxx` stands for the key label of the current execution):

- `data_xxx.txt`: this ASCII file contains the values of the physical and numerical parameters used for the current simulation.
- `pulse_in_xxx.txt`: this ASCII file contains the incident slowly varying optical pulse envelope sampled over the time window.
- `pulse_out_xxx.txt`: this ASCII file contains the slowly varying optical pulse envelope at the fibre end sampled over the time window.

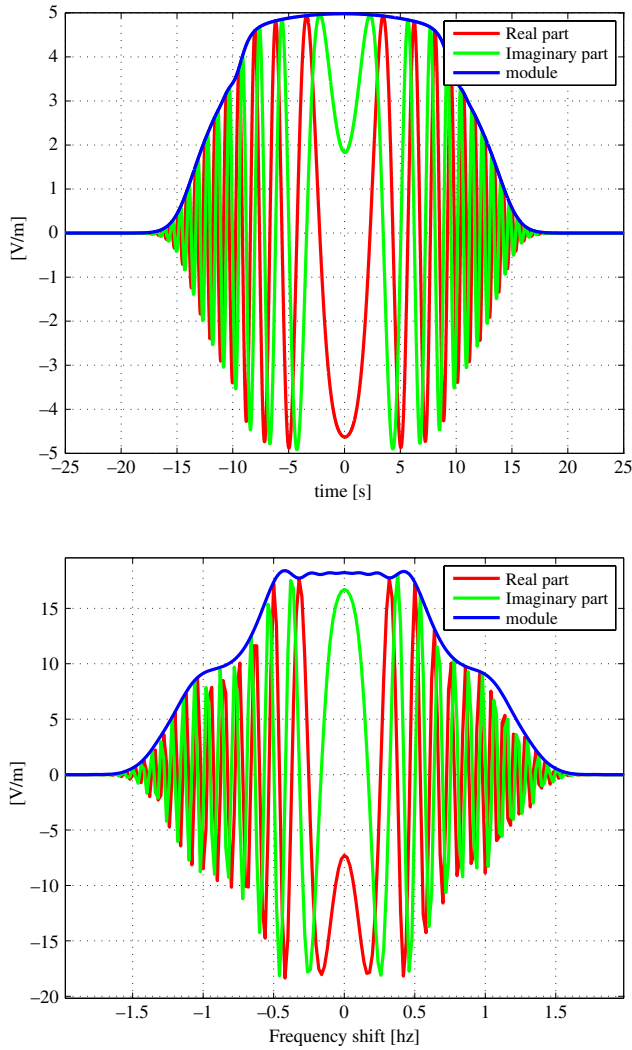


Fig. 2. Screen-shot of the Gnuplot graphics windows where are depicted the time variation of the modulus, real and imaginary parts of the slowly varying pulse envelope $A(L, t)$ at the fibre end (top figure) and the Fourier Transform of these quantities (bottom figure, zoom on the area of interest).

- `in_time_xxx.png`: image of the time variation of the incident slowly varying optical pulse envelope in PNG format as generated by Gnuplot, see Fig. 1.
- `in_spec_xxx.png`: image of the frequency variation of the Fourier Transform of the incident slowly varying optical pulse envelope in PNG format as generated by Gnuplot, see Fig. 1.
- `out_time_xxx.png`: image of the time variation of the slowly varying optical pulse envelope at fibre end in PNG format as generated by Gnuplot, see Fig. 2.
- `out_spec_xxx.png`: image of the frequency variation of the Fourier Transform of the slowly varying optical pulse envelope at fibre end in PNG format as generated by Gnuplot, see Fig. 2.
- `zstep_xxx.txt`: this ASCII file contains the step-size values along the propagation direction computed by the adaptive step-size method.

When the REC variable in the `spip.h` header file has been set to 1 to record the solution at every computational step along the fibre, the directory contains an additional binary file named `pulse_all_xxx.spip`.

4.4. Matlab/Octave tools

Together with the SPIP program are provided several Matlab/Octave scripts to handle SPIP results files. File `plotsol.m` contains a script to draw under Matlab/Octave the slowly varying optical pulse envelope A , from the results files `pulse_in_xxx.txt` and `pulse_out_xxx.txt` generated by the SPIP program. It also plots the Fourier Transform of the slowly varying optical pulse envelope A . The script allows to plot again the solution when the Gnuplot graphical windows of the SPIP program have been closed. It also allows image manipulations thanks to the Matlab/Octave graphics interface. The file `plotspip.m` contains a script to draw the solution at every computational step along the fibre when the REC variable in the `spip.h` header file has been set to 1 and a binary file named `pulse_all_xxx.spip` has been created. Namely, the script draws the slowly varying pulse envelope power $|A|^2$ as a function of time and space (two different views are proposed) and the power spectral density in dBm as a function of wavelength and position along the fibre.

The file `plotstep.m` can be used to draw the variation of the step-size along the fibre length resulting from the adaptive step-size strategy used in the SPIP program.

5. Numerical experiments

We present in this section some numerical experiments carried out with the SPIP program. All the simulations presented here were achieved under Linux Ubuntu 14.04 LTS on a desktop computer equipped with an Intel Core i5-4200M processor and 8 GO RAM. The figures were obtained under Matlab with the various scripts provided with the SPIP program.

5.1. Soliton solution to the NLSE

As a first test example, let us consider the case of the NLSE (2) where $\alpha = 0, f_R = 0, n_{\max} = 2$. When $\beta_2 < 0$, there exists an exact solution to the NLSE known as the optical Soliton [1]. Namely, if the source term is given by

$$a_0 : t \mapsto \frac{N_s}{\sqrt{\gamma L_D} \cosh(t/T_0)} \quad (28)$$

where N_s is the soliton order, T_0 is the pulse half-width and $L_D = -T_0^2/\beta_2$ is the dispersion length, then the solution to the NLSE at the soliton period $z_p = \frac{\pi}{2} L_D$ is given by

$$\forall t \in \mathbb{R} \quad A(z_p, t) = \frac{N_s}{\sqrt{\gamma L_D} \cosh(t/T_0)} e^{i z_p / (2 L_D)}. \quad (29)$$

We consider a 3rd order soliton ($N_s = 3$) and the physical parameters of the numerical experiment are $\lambda = 1550$ nm, $T_0 = 5.673$ ps, $L = 8$ km, $\gamma = 4.3$ W⁻¹ km⁻¹, $\beta_2 = -19.83$ ps² km⁻¹ (SMF-28 fibre). For the simulation, the number of sampling points for the FFT computations was set to 2^{12} and the time window to 200 ps. For adaptive step-size control purposes, the tolerance was set to 10^{-6} and the initial step-size was 10 m. The number of discretization steps along the fibre was found to be 1340 and the computation time was 2.04 s (this time does not include the time devoted to the recording of the solution at every step-size along the fibre). We have depicted in Fig. 3 the time and spectral variations of the slowly varying pulse envelope A along the fibre length; namely we have drawn the power $|A|^2$ as a function of the time and space variables and the power spectral density expressed in dBm (defined as $10 \log_{10}(|A|)$) as a function of the wavelength and position along the fibre. The figures were obtained using the Matlab/Octave tools provided with the SPIP program.

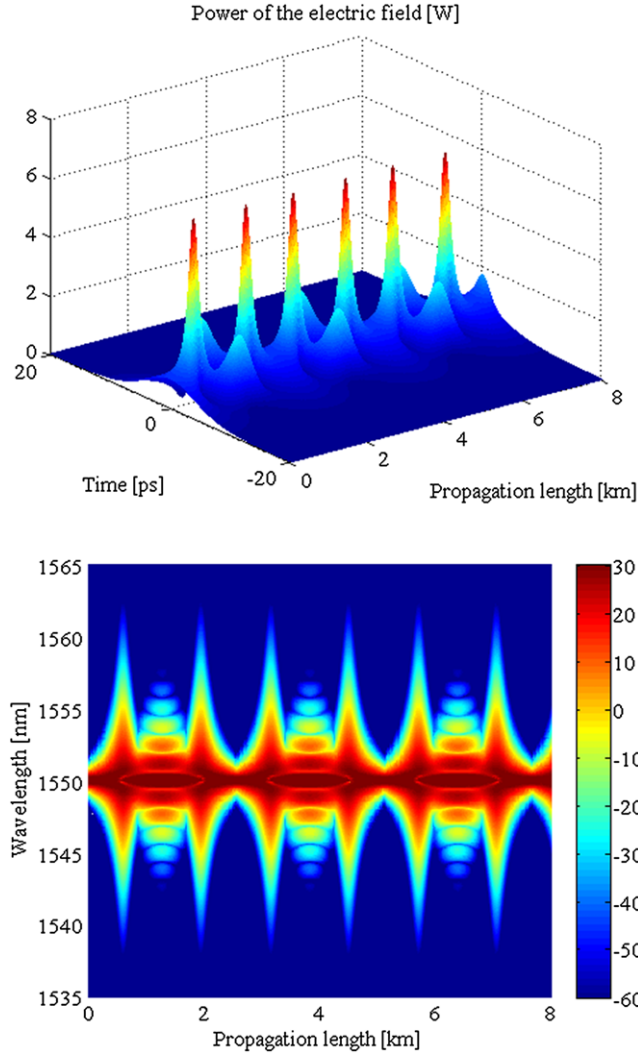


Fig. 3. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of a 3rd order Soliton along a 8 km long fibre.

Since the exact solution is known, the computational error has been measured. After one period ($z = z_p = 2.5493$ km) the relative global error evaluated with the L^2 -norm, defined in (27), is $7.77 \cdot 10^{-5}$ whereas the maximum relative error, defined in (27), is $1.19 \cdot 10^{-4}$. After 3 periods ($z = 3z_p = 7.6479$ km), the quadratic relative global error is $8.01 \cdot 10^{-4}$ whereas the maximum relative error is $1.42 \cdot 10^{-3}$.

The variation of the step-size resulting from the adaptive step-size strategy used in the SPIP program is given in Fig. 4. Since the step-size is adapted so that the local error matches the user defined tolerance at each computational step, small step-sizes indicate an area in the fibre where the slowly varying pulse envelope A varies a lot whereas larger step-sizes indicate areas where the solution varies in a very smooth way.

We have also tested the propagation of a 10th order Soliton in the same fibre. For this experiment, the number of sampling points for the FFT computations was set to 2^{18} and the time window to 200 ps. The tolerance was set to 10^{-9} and the initial step-size was 0.1 m. After one period ($z = z_p = 2.5493$ km), the quadratic relative error is $4.64 \cdot 10^{-4}$ whereas the maximum relative error is $6.33 \cdot 10^{-6}$. The number of steps was 33968 and the simulation time 41 mn. After 3 periods ($z = 3z_p = 7.6479$ km), the quadratic

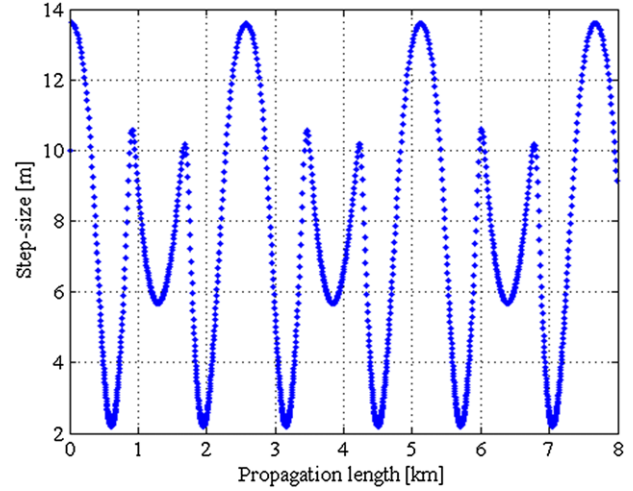


Fig. 4. Variation of the step-size resulting from the adaptive step-size strategy used in the SPIP program for the 3rd order Soliton.

relative global error is $1.38 \cdot 10^{-3}$ whereas the maximum relative error is $1.81 \cdot 10^{-3}$. The number of steps was 101 192. We have depicted in Fig. 5 the time variation along the fibre of the slowly varying pulse envelope power $|A|^2$ (top figure) and the power spectral density expressed in dBm as a function of the wavelength along the fibre (bottom figure) for the 10th order Soliton over one period. We have also depicted in Fig. 6 the variation of the step-size resulting from the adaptive step-size strategy for the 10th order Soliton over one period.

5.2. Solving the GNLSE (I)

We consider the case of the GNLSE (3) with the following set of physical parameters : $\lambda = 1064$ nm, $\gamma = 4.3 \text{ W}^{-1} \text{ km}^{-1}$, $n_{\max} = 3$, $\beta_2 = 19.83 \text{ ps}^2 \text{ km}^{-1}$, $\beta_3 = 0.031 \text{ ps}^3 \text{ km}^{-1}$, $\alpha = 0.046 \text{ km}^{-1}$, $L = 96.77$ m, $f_R = 0.245$. The Hyperbolic Secant pulse at the fibre entrance ($z = 0$) is expressed as

$$\forall t \in \mathbb{R} \quad a_0(t) = \sqrt{P_0} \frac{\exp(-\frac{1}{2} i C t^2 / T_0^2)}{\cosh(t/T_0)} \quad (30)$$

where $T_0 = 2.8365$ ps is the pulse half-width, $C = 1$ is the chirp parameter and $P_0 = 100$ W is the pulse peak power. The number of sampling points for the FFT computations was set to 2^{14} and the time window width to 100 ps. The tolerance for the adaptive step-size control was set to 10^{-6} and the initial step-size was 1 m. The number of discretization steps along the fibre was found to be 301 and the computation time was 3.72 s. We have depicted in Fig. 7 the time variation along the fibre of the slowly varying pulse envelope power $|A|^2$ (top figure) and the power spectral density expressed in dBm as a function of the wavelength along the fibre (bottom figure).

5.3. Solving the GNLSE (II)

We use the SPIP program to solve the GNLSE on a test example chosen to match with a typical case of high speed data propagation through a $L = 20$ km single mode fibre in optical telecommunication with a data's carrier frequency located in the C band of the infra-red spectrum ($\lambda = 1550$ nm). The following set of fibre's parameters was used for the simulation: $\alpha = 0.046 \text{ km}^{-1}$, $\gamma = 4.3 \text{ W}^{-1} \text{ km}^{-1}$, $f_R = 0.245$, $n_{\max} = 3$, $\beta_2 = -19.83 \text{ ps}^2 \text{ km}^{-1}$, $\beta_3 = 0.031 \text{ ps}^3 \text{ km}^{-1}$. The source term $a_0 = A(z = 0)$ was represented as a first order Gaussian pulse:

$$a_0 : t \mapsto \sqrt{P_0} e^{-\frac{1}{2}(t/T_0)^2}$$

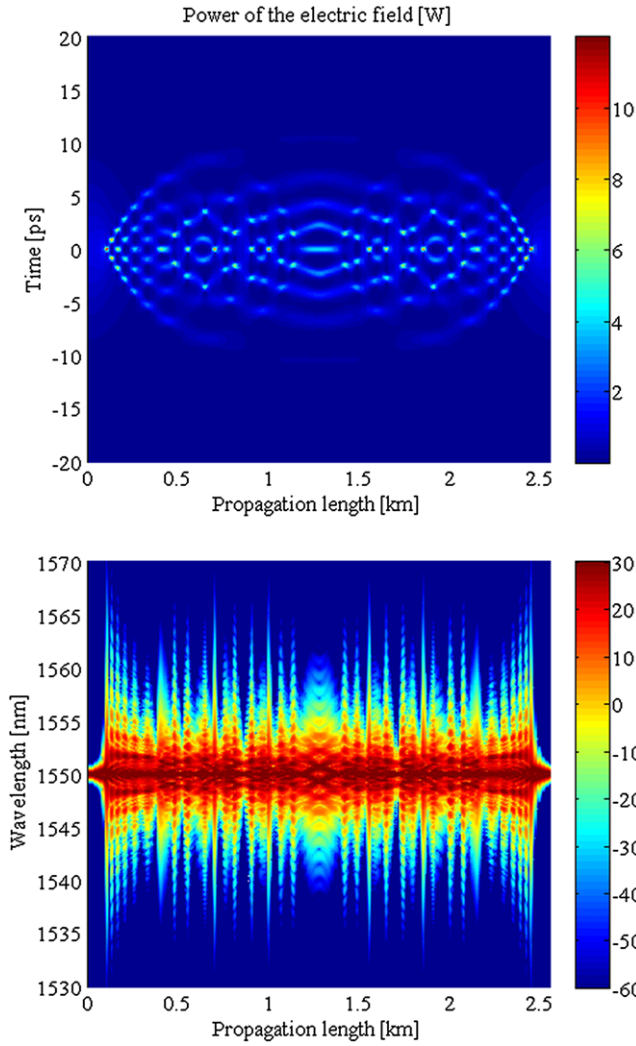


Fig. 5. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of a 10th order Soliton along one period.

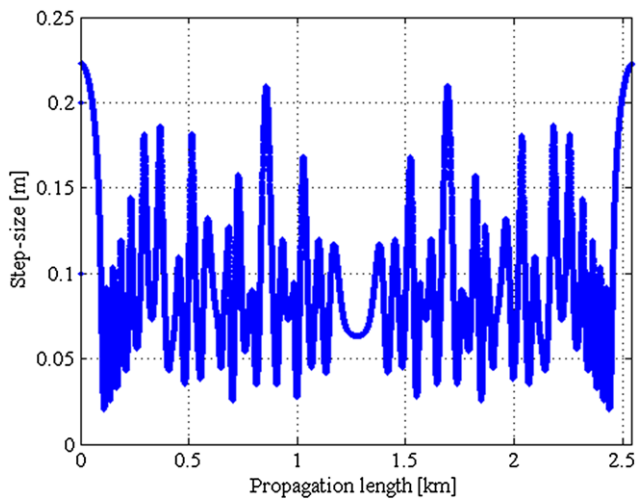


Fig. 6. Variation of the step-size resulting from the adaptive step-size strategy used in the *SPiP* program for the 10th order Soliton over one period.

where $T_0 = 6.8$ ps is the pulse half-width at $1/e$ intensity point and $P_0 = 25$ mW is the pulse peak power.

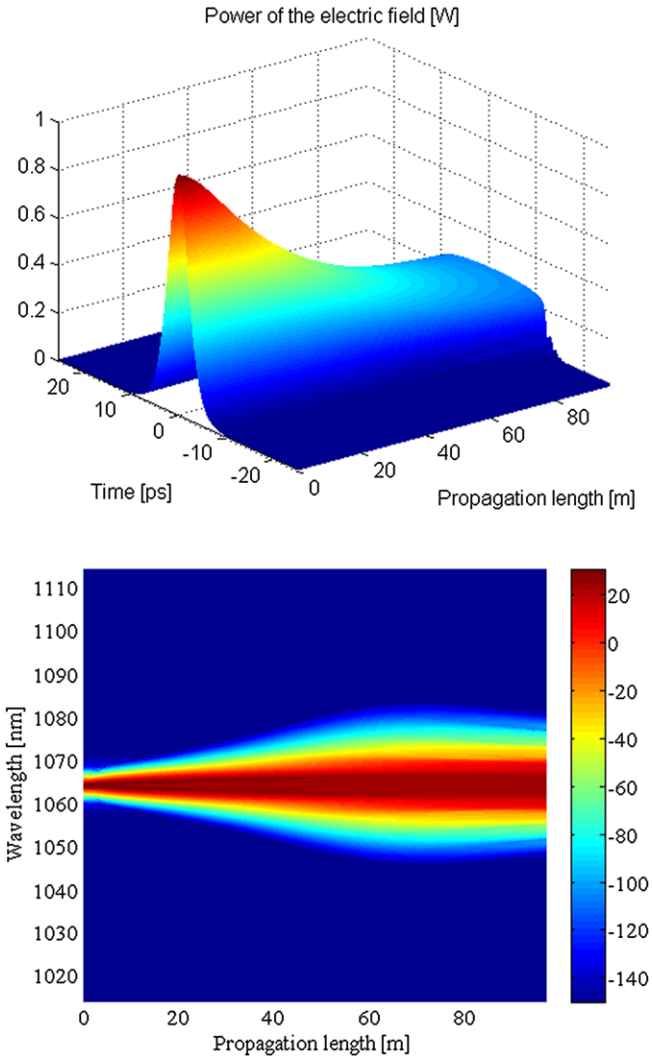


Fig. 7. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of a Hyperbolic Secant pulse along a 96.77 m long fibre.

The number of sampling points for the FFT computations was set to 2^{14} and the time window size to 500 ps. The tolerance for the adaptive step-size control was set to 10^{-9} and the initial step-size was 0.1m. The number of discretization steps along the fibre was found to be 99 and the computation time was 1.87 s. We have depicted in Fig. 8 the time variation along the fibre of the slowly varying pulse envelope power $|A|^2$ (top figure) and the power spectral density expressed in dBm as a function of the wavelength along the fibre (bottom figure).

5.4. Soliton collisions

We now present numerical simulation results for the collision of 2 first order Solitons [1]. It is known that when two neighbouring Solitons are launched with the same phase, they are initially attracted towards each other and then the two pulses periodically coalesce to form one pulse and separate [13]. The source term was

$$a_0 : t \in \mathbb{R} \mapsto \frac{1}{\sqrt{\gamma L_D}} \left(\frac{1}{\cosh((t - T_1)/T_0)} + \frac{Re^{i\phi}}{\cosh(R(t + T_1)/T_0)} \right)$$

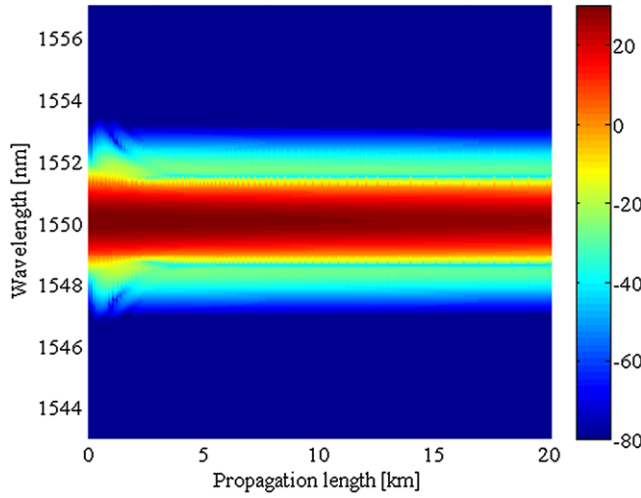
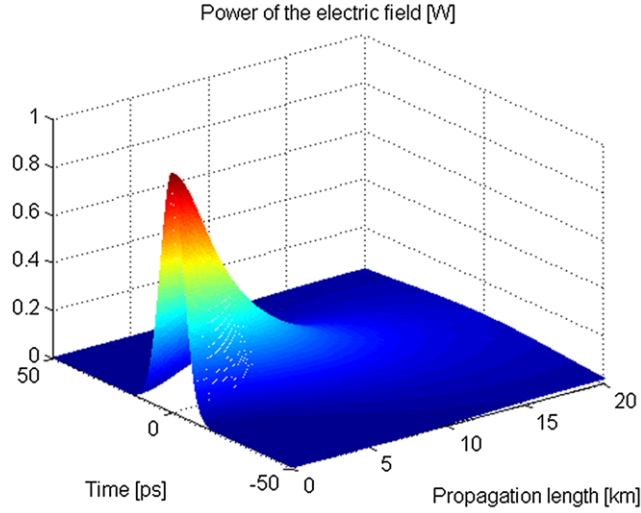


Fig. 8. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of a Gaussian pulse along a 20 km long fibre.

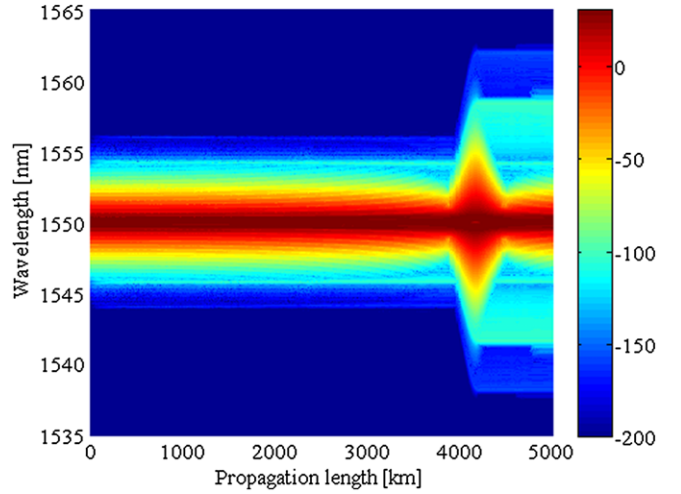
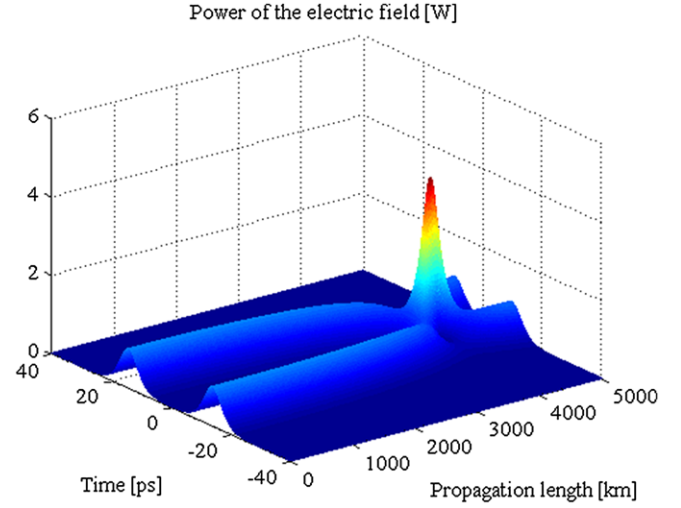


Fig. 9. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of two neighbouring Solitons in a 5000 km long fibre.

where T_0 is the pulse half-width, $L_D = -T_0^2/\beta_2$ is the dispersion length, R accounts for the relative amplitude, ϕ for the relative phase shift and T_1 for the initial separation time. To solve the NLSE, the following physical parameters were taken for the numerical experiment: $L = 5000$ km, $\lambda = 1550$ nm, $\gamma = 2.2 \text{ W}^{-1} \text{ km}^{-1}$, $\beta_2 = -0.1 \text{ ps}^2 \text{ km}^{-1}$, $T_0 = 4$ ps, $T_1 = 100$ ps, $R = 1$ and $\phi = 0$. The time windows were 400 ps wide and the number of FFT nodes was 2^{14} . The initial step-size was set to 1 km and the tolerance to 10^{-6} . The number of discretization steps along the fibre was found to be 480 and the computation time was 3.80 s.

We have depicted in Fig. 9 the evolution of the slowly varying pulse envelope power $|A|^2$ as a function of time and position along the fibre and the evolution of the power spectral density in dBm as a function of wavelength and position along the fibre. We can easily identify in the two figures the position of the Solitons collision. With the values considered for this simulation, the collision of the two Solitons is predicted to happen at a distance of 4161 km [1]. This is confirmed by the plot of the variation of the step-size resulting from the adaptive step-size strategy, see Fig. 10. When the two Solitons catch up, the numerical solution varies a lot over a small distance and therefore the step-size decreases so that the local error meets the prescribed tolerance. The minimal step-size

value in the area of collision is found to occur at a distance of 4165 km.

5.5. Super-continuum generation

Finally, we consider a typical problem for super-continuum generation in Photonic Crystal Fibre (PCF) employing femto-second lasers and pumping in the anomalous dispersion regime [5]. The initial pulse was a 6th order Soliton as defined in (28) with $T_0 = 28.4$ fs, $\lambda = 850$ nm and the fibre parameters were: $L = 10$ cm, $\alpha = 0.046 \text{ km}^{-1}$, $f_R = 0.18$, $\gamma = 45 \text{ W}^{-1} \text{ km}^{-1}$, $\beta_2 = -12.76 \text{ ps}^2 \text{ km}^{-1}$, $\beta_3 = 8.119 \cdot 10^{-2} \text{ ps}^3 \text{ km}^{-1}$, $\beta_4 = -1.321 \cdot 10^{-4} \text{ ps}^4 \text{ km}^{-1}$, $\beta_5 = 3.032 \cdot 10^{-7} \text{ ps}^5 \text{ km}^{-1}$, $\beta_6 = -4.196 \cdot 10^{-10} \text{ ps}^6 \text{ km}^{-1}$ and $\beta_7 = 2.57 \cdot 10^{-13} \text{ ps}^7 \text{ km}^{-1}$. The simulation parameters were the following: time windows width 6 ps, number of FFT nodes 2^{14} , initial step-size 10^{-4} m and tolerance 10^{-6} . The computation time was 14.9 s and the number of steps in the adaptive step-size strategy was found to be 1475. We have depicted in Fig. 11 the evolution of the slowly varying pulse envelope power $|A|^2$ as a function of time and position along the fibre and the evolution of the power spectral density in dBm as a function of wavelength and position along the fibre. We also

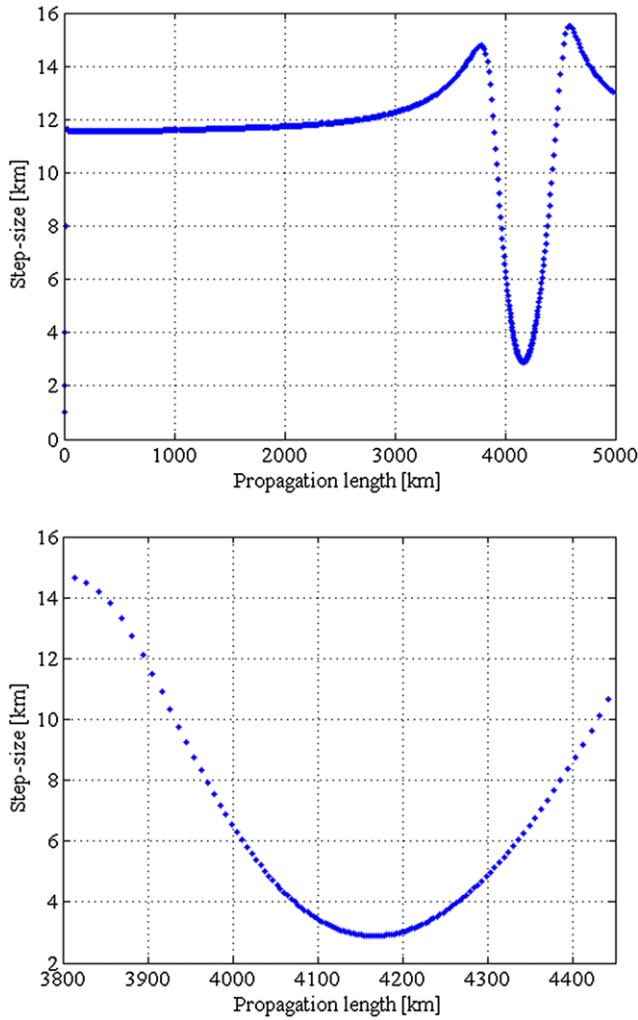


Fig. 10. Variation of the step-size resulting from the adaptive step-size strategy (top figure) and zoom in the area of collision (bottom figure).

provide in Fig. 12 the variation of the step-size along the fibre. One can observe how the step-size is adapted in the area where the spectral broadening caused by the interaction between self-phase modulation and group velocity dispersion takes place in the first centimetre of the fibre [5].

6. Conclusion

We have presented the open-source software SPIP that solves the Generalized Non-Linear Schrödinger Equation (GNLSE) as well as the Non-Linear Schrödinger Equation (NLSE) by the Interaction Picture method combined with an adaptive step-size control strategy based on the use of a dedicated Embedded Runge–Kutta method. We have shown the results provided by the SPIP program on typical problems involving the GNLSE/NLSE in optics.

Acknowledgements

The present work has been undertaken under the framework of the Green Laser Project supported by Région Bretagne, France. The authors would like to thank F. Mahé, F. Méhat and R. Texier-Picard from the Research Institute of Mathematics in Rennes (IRMAR CNRS UMR 6625), France, for their involvement in the mathematical study of the IP method as well as T. Chartier from

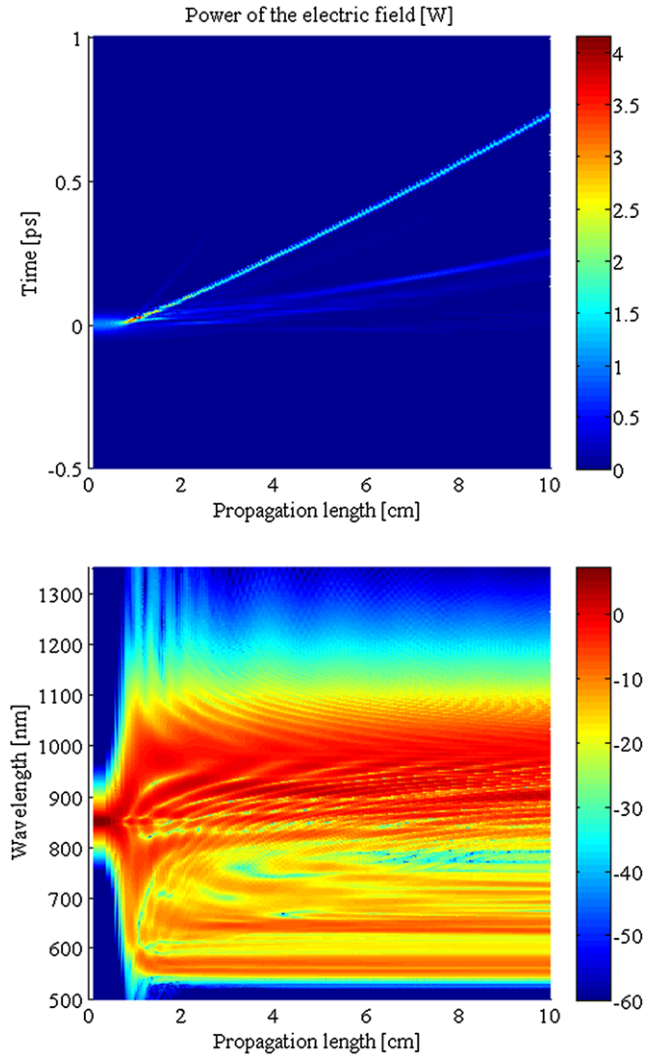


Fig. 11. Slowly varying pulse envelope power $|A|^2$ in W (top figure) as a function of time and position along the fibre and power spectral density in dBm (bottom figure) as a function of wavelength and position along the fibre for the propagation of a 6th order Soliton pulse into a 10 cm PCF.

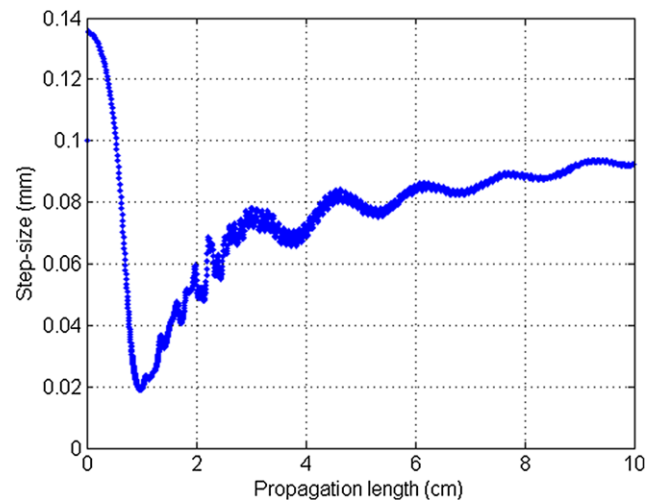


Fig. 12. Variation of the step-size resulting from the adaptive step-size strategy for the super-continuum generation.

Foton laboratory in Lannion (CNRS UMR 6082), France, for his support.

References

- [1] G. Agrawal, *Nonlinear Fibre Optics*, fourth ed., Academic Press, 2013.
- [2] K. Okamoto, *Fundamentals of Optical Waveguides*, Optics and Photonics, Elsevier, 2006.
- [3] B. Caradoc-Davies, *Vortex dynamics in Bose-Einstein condensate* (Ph.D. thesis), University of Otago, NZ, 2000.
- [4] M. Davis, *Dynamics in Bose-Einstein condensate* (Ph.D. thesis), University of Oxford, UK, 2001.
- [5] J. Hult, *A fourth-order Runge-Kutta in the Interaction Picture method for simulating supercontinuum generation in optical fibres*, *J. Lightwave Technol.* 25 (12) (2007) 3770–3775.
- [6] S. Balac, A. Fernandez, F. Mahé, F. Méhats, R. Texier-Picard, *The Interaction Picture method for solving the nonlinear Schrödinger equation in optics*, *ESAIM Math. Model. Numer. Anal.* (2015). <http://dx.doi.org/10.1051/m2an/2015060>. in press.
- [7] J. Dormand, P. Prince, *A family of embedded Runge-Kutta formulae*, *J. Comput. Appl. Math.* 6 (1980) 19–26.
- [8] S. Balac, F. Mahé, *Embedded Runge-Kutta scheme for step-size control in the Interaction Picture method*, *Comput. Phys. Comm.* 184 (2013) 1211–1219.
- [9] A. Pazy, *Semigroups of linear operators and applications to partial differential equations*, in: *Applied Mathematical Sciences*, vol. 44, Springer, 1992.
- [10] J. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley and Sons, 2008.
- [11] E. Hairer, S.P. Norsett, G. Wanner, *Solving Ordinary Differential Equation (I): Nonstiff Problems*, Springer-Verlag, 1993.
- [12] M. Frigo, S. Johnson, *The design and implementation of FFTW3*, *Proceedings of the IEEE* 2 (93) (2005) 216–231.
- [13] C. Desem, P. Chu, *Reducing soliton interaction in single-mode optical fibres*, *Optoelectronics, IEE Proc. J.* 134 (3) (1987) 145–151.
- [14] P. Janert, *Gnuplot in Action, Understanding Data with Graphs*, Manning Publications, 2009.