Comparison of adaptive step-size control strategies for solving the Generalised Non-Linear Schrodinger Equation in optics by the Interaction Picture method

Stéphane Balac, Arnaud Fernandez

Comparison of adaptive step-size control strategies for solving the Generalised Non-Linear Schrodinger Equation in optics by the Interaction Picture method

Stéphane Balac and Arnaud Fernandez

UEB, Université Européenne de Bretagne, Université de Rennes I
CNRS, UMR 6082 FOTON, Enssat, 6 rue de Kerampont, CS 80518,
22305 Lannion, France

Abstract. This report is devoted to the study and to the comparison of methods for estimation of the local error and for adaptive step-size control when solving the generalised nonlinear Schrodinger equation (GNLSE) in optics by the Interaction Picture (IP) method or by the Symmetric Split-Step (S3F) method. Namely, we propose and study the use of an embedded Runge-Kutta method to solve the nonlinear problem involved in the IP or S3F method and to deliver a local error estimate at each step of the discretisation grid for the purpose of an adaptive step-size control. This method preserves the advantages of the RK4 method exploited in the IP or S3F methods and do not add any extra computational cost. We compare this method to other standard methods for estimating the local error such as the "step doubling method" or energy conservation based methods.

Keywords. embedded Runge-Kutta method, adaptive step-size control, Schrodinger Equation, Interaction Picture method, Split-Step method, optical fibres

Document last modification : 10 October 2012

Contents

1. Introduction

The propagation of pulses in optical fibres is described by the generalised nonlinear Schrodinger equation (GNLSE), which takes into account fiber losses, nonlinear effects, and higher-order chromatic dispersion [1]. The GNLSE is a partial differential equation not amenable to analytical solution; the use of numerical approximation techniques is therefore mandatory. At present time, numerical methods based on split-step Fourier transform approaches are the most widely used for simulating wave propagation in optical fibres [1–5]. Recently a "fourth-order Runge-Kutta method in the interaction picture method" (RK4-IP method) has been proposed [6] as a very promising alternative to the split step methods for solving the GNLSE. When solving the GNLSE in the context of optics by the Symmetric Split-Step Fourier method (S3F method) or by the Interaction Picture method (IP method), the nonlinear ordinary differential equation (ODE) resulting from the splitting is mostly solved by the standard fourth order Runge-Kutta (RK4) scheme [7,8] for mainly 2 reasons. First of all, this RK4 scheme offers a good compromise between accuracy and cost of the computations. Moreover, this RK4 scheme exhibits nice properties of symmetry that can be capitalised to reduce the cost of the whole computational process, mainly by diminishing the number of Fourier transforms to be computed. A precise comparison between the RK4-IP method and the Symmetric Split-Step Fourier method with fourth-order Runge-Kutta method (S3F-RK4 method) has been achieved in [9]. It is shown that the RK4-IP method has a cost similar to the S3F-RK4 method due

to a computational approach very similar to the one involved in the Split-Step methods but exhibits a convergence rate proportional to h^4 where h denotes the spatial discretisation step whereas the S3F-RK4 method is limited by the second order accuracy of the Strang symmetric splitting formula [4,5,10] and exhibits a convergence rate proportional to h^2.

However, any numerical method for solving the GNLSE will perform poorly if the approximate solution is computed on a mesh grid with a constant step h. Ideally the step-size between 2 successive grid points should be selected automatically to maintain the error lower than a given value in order to achieve both reliability and efficiency of the computations. There are several ways to estimate the local error in each point of the RK4 mesh grid and to select a value for the size of the next step. The most common and general way to estimate the local error is by a process known as "step doubling" [11]. To estimate the local error, this method requires for each step the computation of a coarse solution and a fine solution (usually obtained by dividing by 2 the step-size used for the coarse solution). This way for estimating the local error is accountable for an over computational cost of around 50% more than the same method without local error estimate for the same accuracy of the computations. A cheaper adaptive step-size method dedicated to the GNLSE is propound in [12]. It is based on the conservation of a physical quantity termed "the optical photon number" during the propagation of an electromagnetic field along a fibre when loss in the fibre is neglected. It is therefore possible to calculate the "photon number error", which is related to a certain local error, at each grid point to retrieve information about the numerical error over one computational step of the IP method applied to GNLSE. The interest of this approach, specific to GNLSE for lose-less fibres, is that the computation of the photon number error can be done at a very cheap numerical cost. In [12] it is stated that the method can be extend to the GNLSE for loss fibres and a formula where the loss over each computational step is estimated by linear interpolation is proposed.

In this document we present another way to estimate the local error based on an embedded Runge-Kutta method [7,8]. This method is very general and do not require the assumption of negligible fiber losses. Moreover it preserves the advantages of the RK4 method exploited in the IP method and do not add any extra computational cost. We also compare the efficiency of these 4 approaches for estimating the local error in the context of adaptive step-size approach for the GNLSE.

The document is organised as follows. In section 2 we present the physical framework for the GNLSE in the context of optical pulse propagation in fibers and a description of the IP method is given. In section 3 we recall the main features of the Runge-Kutta methods emphasising on the notion of local and global errors. The use of a fourth order Runge-Kutta method is then presented in conjunction with the IP method to solve the GNLSE. Section 4 is devoted to the presentation of the above mentioned methods to estimate the local error in the RK4-IP method in order to carry out adaptive step computations in the RK4-IP method. We study the existence of an embedded Runge-Kutta method that preserves the very nice computational features of the RK4-IP method and at the same time deliver a local error estimate at no extra cost. In section 5 a detailed numerical algorithm is presented for the embedded fourth order Runge-Kutta method in the Interaction picture (ERK4-IP) and numerical results are given.

2. The generalised nonlinear Schrodinger equation

2.1. Physical framework

Wave propagation into an optical fibre with group index n_g is governed by the generalised nonlinear Schrödinger equation (GNLSE). This particular form of the Schrödinger equation is obtained from the general set of Maxwell equations taking advantage of a certain number of assumptions made possible from the very specific characteristics of (quasi-)monochromatic wave propagation at pulsation ω_0 in a medium such as a fibre [1]. 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 polarised 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 t and position $\mathbf{r} = (x, y, z)$ as

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

where $A(z, t)$ represents the slowly varying electric pulse envelope, $F(x, y)$ is the electric wave transverse representation, k is the wave number and t the local time of the moving frame travelling along with the pulse at the group velocity $v_g = c/n_g$. The relationship between τ and the group velocity v_g is : $t = \tau - \frac{z}{v_g}$. The expression of the electric wave transverse representation F can most of the time be computed explicitly using the classical method of separation of variables for PDE [1]. For instance, for circular constant transverse section fibres, it is expressed in terms of Bessel functions.

In this paper we consider the case where the evolution of the slowly varying pulse envelope A is governed by the following form of the GNLSE

$$\begin{aligned}\frac{\partial}{\partial z}A(z,t) = -\frac{\alpha}{2}A(z,t) + \left(\sum_{n=2}^{n_{\max}} \mathrm{i}^{n+1}\frac{\beta_n}{n!}\frac{\partial^n}{\partial t^n}A(z,t)\right) \\ + \mathrm{i}\gamma\left(\mathrm{Id} + \frac{\mathrm{i}}{\omega_0}\frac{\partial}{\partial t}\right)\cdot\left[A(z,t)\left((1-f_R)\left|A(z,t)\right|^2 + f_R\int_0^\infty h_R(\tau)|A(z,t-\tau)|^2 d\tau\right)\right]\end{aligned} \tag{2}$$

where α is the coefficient of linear attenuation, β_n, $n \geqslant 2$ are the linear dispersion coefficients, γ is the nonlinear parameter and Id stands for the identity operator. In equation (2) non linear dispersion is taken into account through the simplified optical shock parameter $\tau_{shock} = 1/\omega_0$. Instantaneous Kerr effect manifests itself through the term $(1 - f_R)\left|A\right|^2$. The delayed Raman contribution in the time domain is taken into account through the convolution product between the instantaneous power $\left|A\right|^2$ and the Raman time response function for silica-core fibres h_R (an analytical form is proposed in [1]). The constant f_R represents the fractional contribution of the delayed Raman response to nonlinear polarisation. The partial differential equation (2) is considered together with the following boundary condition for $z = 0$

$$\forall t \in \mathbb{R} \qquad A(0, t) = a_0(t) \tag{3}$$

where a_0 is a given function in the Hilbert space $\mathbb{L}^2(\mathbb{R}, \mathbb{C})$.

2.2. Mathematical framework

A precise mathematical framework for the IP method applied to the GNLSE (2), including existence and uniqueness results in appropriate functional spaces and

regularity results is at present time under study. Therefore our presentation remains somehow formal from a mathematical point of view. We denote by $\mathcal{D}$ the unbounded linear operator

$$\mathcal{D} : A \in E \longmapsto -\frac{1}{2}\alpha A - \sum_{n=2}^{n_{\max}} \beta_n \frac{\mathrm{i}^{n-1}}{n!}\partial_t^n A \tag{4}$$

and we introduce the nonlinear operator

$$\mathcal{N} : A \in E \longmapsto \mathrm{i}\gamma\, \mathcal{T}_t\big[(1-f_r)A|A|^2 + f_r A\,(h_R \star_t |A|^2)\big] \tag{5}$$

where $\star_t$ stands for the convolution operator with respect to the time variable and $\mathcal{T}_t$ refers to the differential operator $\mathrm{Id} + \frac{1}{\omega_0}\frac{\partial}{\partial t}$. We want to point out that although the computation of the quantity $\mathcal{N}(A)(z,t)$ may appear to be cumbersome, it's nothing of the sort. It can be achieved in an efficient way by using the properties of the Fourier transform with respect to derivation and convolution, see section 5.3 and [13] for details.

2.2.1. Splitting approaches We are interested in solving the following PDE problem for the unknown A

$$(\mathcal{P}) \quad \begin{cases} \dfrac{\partial}{\partial z}A(z,t) = \mathcal{D}A(z,t) + \mathcal{N}(A)(z,t) & \forall z \in [0,L]\ \forall t \in \mathbb{R} \\ A(0,t) = a_0(t) & \forall t \in \mathbb{R} \end{cases} \tag{6}$$

where the operators $\mathcal{D}$ and $\mathcal{N}$ do not commute with each other. This PDE problem is not amenable to analytical solution and the use of numerical approximation techniques is required. It could be solved by general approximation methods for PDE such as the Finite Difference Time Domain (FDTD) method or the Finite Element Method (FEM) but the particular structure of the PDE (2) enables more efficient numerical approaches. Moreover, both the FDTD and the FEM suffer from CFL restrictions between time step and spatial step in order to satisfy stability. Classical numerical methods for solving (6) are based on a Split-Step Fourier transform approach and among them the Symmetric Split-Step Fourier method (S3F method) is the most widely used [1–5]. Recently a "fourth-order Runge-Kutta method in the interaction picture method" (RK4-IP method) has been proposed [6] as a very promising alternative to Split-Step methods for solving the GNLSE. Both methods (S3F and IP) exhibit a computational inner structure very similar and therefore a computational cost very comparable even if the IP method has a convergence rate proportional to h^4 whereas the S3F method is limited by the second order accuracy of Strang splitting formula and has a convergence rate proportional to h^2 [9].

For numerical purposes, the interval $[0,L]$ corresponding to the fiber length is divided into K sub-intervals where the spatial grid points are denoted by z_k, $k = \{0,\dots,K\}$ such that $]0,L] = \cup_{k=0}^{K-1}]z_k, z_{k+1}]$ where $0 = z_0 < z_1 < \cdots < z_{K-1} < z_K = L$. For all $k \in \{0,\dots,K-1\}$ the step length between z_k and z_{k+1} is denoted h_k and we also set $z_{k+\frac{1}{2}} = z_k + \frac{h_k}{2}$. The IP method for problem (6) consists in solving over each sub-interval $[z_k, z_{k+1}]$ the following 3 nested problems [9, 13]

$$\begin{cases} \dfrac{\partial}{\partial z}A_k(z,t) = \mathcal{D}A_k(z,t) & \forall z \in [z_k, z_{k+\frac{1}{2}}]\ \forall t \in \mathbb{R} \\ A_k(z_k,t) = A_{k-1}(z_k,t) & \forall t \in \mathbb{R} \end{cases} \tag{7}$$

where for $k \geqslant 1$ the mapping $t \in \mathbb{R} \mapsto A_{k-1}(z_k, t)$ represents the solution at grid point z_k computed during step $k-1$ and for $k = 0$ we have $A_{-1}(z_0, t) = a_0(t)\ \forall t \in \mathbb{R}$,

$$\left\{ \begin{array}{ll} \dfrac{\partial}{\partial z} A_k^{\mathrm{ip}}(z,t) = \mathcal{G}_k(z, t, A_k^{\mathrm{ip}}) & \forall z \in [z_k, z_{k+1}]\ \forall t \in \mathbb{R} \\ A_k^{\mathrm{ip}}(z_k, t) = A_k(z_{k+\frac{1}{2}}, t) & \forall t \in \mathbb{R} \end{array} \right. \tag{8}$$

where $t \in \mathbb{R} \mapsto A_k(z_{k+\frac{1}{2}}, t)$ represents the solution to problem (7) at point $z_{k+\frac{1}{2}}$ and

$$\left\{ \begin{array}{ll} \dfrac{\partial}{\partial z} A_k(z,t) = \mathcal{D} A_k(z,t) & \forall z \in [z_{k+\frac{1}{2}}, z_{k+1}]\ \forall t \in \mathbb{R} \\ A_k(z_k, t) = A_k^{\mathrm{ip}}(z_{k+1}, t) & \forall t \in \mathbb{R} \end{array} \right. \tag{9}$$

where $t \mapsto A_k^{\mathrm{ip}}(z_{k+1}, t)$ represents the solution to problem (8) at grid point z_{k+1}. In problem (8) we have set

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

where the exponential terms have to be understood in the sense of the continuous group generated by the unbounded linear operator $\mathcal{D}$ [14]. For all $k \in \{0, \ldots, K-1\}$ the mapping $t \mapsto A_k(z_{k+1})$ coincides with the solution A to problem (6) at grid point z_{k+1}.

As mentioned before, IP method and S3F method are rather similar in their inner structure since with the S3F method we solve over each sub-interval $[z_k, z_{k+1}]$ the following 3 connected problems :

$$\left\{ \begin{array}{ll} \dfrac{\partial}{\partial z} A_k(z,t) = \mathcal{D} A_k(z,t) & \forall z \in [z_k, z_{k+\frac{1}{2}}]\ \forall t \in \mathbb{R} \\ A_k(z_k, t) = A_{k-1}(z_k, t) & \forall t \in \mathbb{R} \end{array} \right. \tag{10}$$

where for $k \geqslant 1$ the mapping $t \in \mathbb{R} \mapsto A_{k-1}(z_k, t)$ represents the solution at grid point z_k computed during step $k-1$ and for $k = 0$ we have $A_{-1}(z_0, t) = a_0(t)\ \forall t \in \mathbb{R}$,

$$\left\{ \begin{array}{ll} \dfrac{\partial}{\partial z} B_k(z,t) = \mathcal{N}(B_k)(z,t) & \forall z \in [z_k, z_{k+1}]\ \forall t \in \mathbb{R} \\ B_k(z_k, t) = A_k(z_{k+\frac{1}{2}}, t) & \forall t \in \mathbb{R} \end{array} \right. \tag{11}$$

where $t \mapsto A_k(z_{k+\frac{1}{2}}, t)$ represents the solution to problem $(\mathcal{P}_k^1)$ at point $z_{k+\frac{1}{2}}$

$$\left\{ \begin{array}{ll} \dfrac{\partial}{\partial z} A_k(z,t) = \mathcal{D} A_k(z,t) & \forall z \in [z_k + \frac{h_k}{2}, z_{k+1}]\ \forall t \in \mathbb{R} \\ A_k(z_k, t) = B_k(z_{k+1}, t) & \forall t \in \mathbb{R} \end{array} \right. \tag{12}$$

where $t \mapsto B_k(z_{k+1}, t)$ represents the solution to problem (11) at node z_{k+1}. Here, for all $k \in \{0, \ldots, K-1\}$ the mapping $t \mapsto A_k(z_{k+1})$ coincides with an approximation of the solution A to problem (6) at grid point z_{k+1}. One can notice that problems (8) and (10) are the same as well as problems (9) and (12). The difference between the 2 computational approaches involved in the IP and S3F methods lies in the nonlinear problems (11) and (8).

The main difference between the 2 splitting formulations can be explained as follows, see [9] for details. The splitting (7)–(9) is exact since it originates from the following change of unknown over each sub-interval $[z_k, z_{k+1}]$

$$A_k^{\mathrm{ip}} : (z,t) \in [z_k, z_{k+1}] \times \mathbb{R} \longmapsto \exp(-(z - z_{k+\frac{1}{2}})\mathcal{D}) \cdot A_k(z,t) \tag{13}$$

whereas the splitting (10)–(12) results from the symmetric Strang formula [10] and corresponds to a second order approximation of the original problem over each sub-interval $[z_k, z_{k+1}]$. In this case the splitting error is due to the non-commutativity of the operators $\mathcal{N}$ and $\mathcal{D}$ in accordance with the celebrated Baker-Campbell-Hausdorf formula [5, 15].

From now we will only consider the IP method but all the forthcoming discussion also applies to the S3F method.

2.2.2. Formal solutions to the linear problems by Fourier transform The solution to the linear PDE problem (10) at node $z_{k+\frac{1}{2}}$ can formally be written as

$$\forall t \in \mathbb{R} \qquad A_k(z_{k+\frac{1}{2}}, t) = \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot A_k(z_k, t) \tag{14}$$

where the exponential term as to be understood in the sense of the continuous group generated by the unbounded linear operator $\mathcal{D}$ [14]. The mapping $t \in \mathbb{R} \mapsto A_k(z_{k+\frac{1}{2}}, t)$ can be computed very efficiently by the mean of the Fourier transform as follows

$$\forall t \in \mathbb{R} \qquad \exp(\tfrac{h_k}{2}\mathcal{D}_t) \cdot A_k(z_k, t) = \mathcal{F}^{-1}\big[\nu \mapsto \widehat{A}_k(z_k, \nu)\ \mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}}\big](t) \tag{15}$$

where $\widehat{A}_k(z_k, \cdot)$ denotes the Fourier transform of $A_k(z_k, \cdot)$, $\widehat{d_\nu} = -\frac{1}{2}\alpha + \mathrm{i}\sum_{n=2}^{n_{\max}} \frac{\beta_n}{n!}(2\pi\nu)^n$ and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform operator. Thus, for all $t \in \mathbb{R}$ the quantity $\exp(\frac{h_k}{2}\mathcal{D}) \cdot A_k(z_k, t)$ can be computed by multiplying the Fourier transform of the mapping $t \mapsto A_k(z_k, t)$ by the mapping $\nu \mapsto \mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}}$ and then applying the inverse Fourier transform to the result. The solution to the linear PDE problem (9) at node z_{k+1} can be obtained in a very similar way.

2.2.3. Computation of the nonlinear terms In the IP method or in the S3F method for the GNLS equation (2) one need to compute for various mapping f the nonlinear terms $\mathcal{N}(f)$ where $\forall t \in \mathbb{R}\ \forall z \in [0, L]$

$$\begin{aligned}\mathcal{N}(f)(z,t) = \mathrm{i}\gamma \Big[I_{\mathrm{d}} + \frac{1}{\omega_0}\frac{\partial}{\partial t}\Big] \cdot \big((1 - f_r) f(z,t)|f(z,t)|^2 \\ + f_r f(z,t)\,(h_R(t) \star |f(z,t)|^2)\big).\end{aligned} \tag{16}$$

One efficient way of computing this quantity is by mean of the Fourier transform. Indeed, time derivation of functions is reduced to multiplying the Fourier transform of the function by a factor $-2\mathrm{i}\pi\nu$. Namely, we have

$$\begin{aligned}\widehat{\mathcal{N}(f)}(z,\nu) = \mathrm{i}\gamma\big(1 - \frac{2\mathrm{i}\pi\nu}{\omega_0}\big) \times \mathcal{F}[t \mapsto (1 - f_r) f(z,t)|f(z,t)|^2 \\ + f_r f(z,t)\,(h_R \star_t |f(z,t)|^2)](\nu).\end{aligned} \tag{17}$$

To compute $h_R(t) \star |f(z,t)|^2$, we use the properties of the Fourier transform with respect to convolution:

$$h_R(t) \star |f(z,t)|^2 = \mathcal{F}^{-1}[\nu \mapsto \widehat{h}_R(\nu) \times \widehat{|f(z,.)|^2}(\nu)](t). \tag{18}$$

Thus, computation of $\mathcal{N}(f)(z,t)$ for all $t \in \mathbb{R}$ and $z \in \mathbb{R}^+$ consists in the following steps:

- compute the Fourier transforms $\widehat{h}_R$ and $\widehat{|f(z,.)|^2}$ of h_R and $t \mapsto |f(z,t)|^2$ respectively
- multiply these 2 mappings and compute the inverse Fourier transform of the result to obtain the mapping $(z,t) \mapsto h_R(t) \star |f(z,t)|^2$
- compute the Fourier transform of the mapping $t \mapsto (1-f_r)f(z,t)|f(z,t)|^2 + f_r f(z,t)\,(h_R \star |f(z,t)|^2)$
- multiply the result by the mapping $\nu \mapsto \mathrm{i}\gamma\left(1 - \frac{2\mathrm{i}\pi\nu}{\omega_0}\right)$
- compute the inverse Fourier transform of this last product.

We now focus on the nonlinear problem (8). The major interest of the Interaction Picture approach is that on the contrary to problem (6), problem (8) doesn't anymore involve explicitly partial derivation with respect to the time variable t. Therefore, it can be numerically solved using a standard quadrature scheme for ordinary differential equations (ODE) such as the "standard" fourth-order Runge-Kutta (RK4) scheme.

3. Interaction Picture method with the standard RK4 scheme

3.1. Overview of the standard RK4 scheme

3.1.1. Main features of RK schemes Let us recall the main features of Runge-Kutta (RK) schemes for solving an initial value problem specified as follows

$$\begin{cases} y'(z) = f(z, y(z)) \quad \forall z \in [0, L] \\ y(0) = y_0 \end{cases}. \tag{19}$$

We refer to [7, 8, 16, 17] for a more detailed presentation and for proof of the results. Given a subdivision $(z_k)_{k=0,\ldots,K}$ of $[0, L]$, the so-called "standard" fourth order Runge-Kutta (RK4) scheme for this problem consists in approaching the exact solution y at grid point z_k by the value y_k computed iteratively through the formula

$$y_{k+1} = y_k + \frac{h_k}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4\right) \tag{20}$$

where $h_k = z_{k+1} - z_k$ and

$$\begin{aligned} \alpha_1 &= f(z_k, y_k) \\ \alpha_2 &= f(z_k + \tfrac{h_k}{2}, y_k + \tfrac{h_k}{2}\alpha_1) \\ \alpha_3 &= f(z_k + \tfrac{h_k}{2}, y_k + \tfrac{h_k}{2}\alpha_2) \\ \alpha_4 &= f(z_k + h_k, y_k + h_k\alpha_3) \end{aligned} \tag{21}$$

Each RK method can be described in a very concise manner by putting its coefficients in an array called a *Butcher tableau* [7]. Butcher tableau for the RK4 method defined by (20)-(21) is

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{22}$$

A presentation of other RK4 methods can be found e.g. in [7], section 322.

Due to the need to compute the 4 terms $\alpha_1, \ldots, \alpha_4$ given by (21) to obtain the value of y_{k+1} from y_k, the RK4 method defined in (20)-(21) is called a 4 stages RK method. In the literature on RK methods, the coefficients $b_1 = \frac{1}{6}, b_2 = \frac{1}{3}, b_3 = \frac{1}{3}$ and $b_4 = \frac{1}{6}$ in front of the values of $\alpha_1, \ldots, \alpha_4$ in (20) are known as the weights of the RK scheme. One can recast relation (20) in the following way

$$y_{k+1} = y_k + h_k\, \Phi(z_k, y_k; h_k) \quad \text{where } \Phi(z_k, y_k; h_k) = \frac{1}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4\right) \tag{23}$$

and the dependence in the variables z_k, y_k and h_k of the α_i terms is given by (21).

We recall that more generally any RK method can be defined through a Butcher tableau in the form

$$\begin{array}{c|c} c & \mathrm{A} \\ \hline & b^\top \end{array}$$

where A is a q by q matrix with real entries and b and c are 2 real vectors in $\mathbb{R}^q$. The corresponding RK approximation scheme for problem (19) consists in computing the solution at grid point z_{k+1} from the value computed at grid point z_k by the integral relation

$$\begin{aligned} y(z_{k+1}) &= y(z_k) + \int_{z_k}^{z_{k+1}} y'(z)\ \mathrm{d}z = y(z_k) + \int_{z_k}^{z_{k+1}} f(z, y(z))\ \mathrm{d}z \\ &= y(z_k) + h_k \int_0^1 f\left(h_k\zeta + z_k, y(h_k\zeta + z_k)\right)\ \mathrm{d}\zeta \\ &\approx y(z_k) + h_k \sum_{i=1}^{q} b_i\, f\left(z_{k,i}, y(z_{k,i})\right) \qquad \text{with } z_{k,i} = z_k + h_k c_i \end{aligned}$$

where the approximation in the last line corresponds to the use of the quadrature formula

$$\int_0^1 \Psi(\zeta)\ \mathrm{d}\zeta \approx \sum_{i=1}^{q} b_i\, \Psi(c_i) \tag{24}$$

and $\forall i \in \{1, \ldots, q\}$ the solution at quadrature node $z_{k,i}$ is given by

$$\begin{aligned} y(z_{k,i}) &= y(z_k) + \int_{z_k}^{z_{k,i}} f(z, y(z))\ \mathrm{d}z = y(z_k) + h_k \int_0^{c_i} f\left(h_k\zeta + z_k, y(h_k\zeta + z_k)\right)\ \mathrm{d}\zeta \\ &\approx y(z_k) + h_k \sum_{j=1}^{q} a_{ij} f\left(z_{k,j}, y(z_{k,j})\right) \end{aligned}$$

where the approximation now corresponds to the quadrature formula

$$\int_0^{c_i} \Psi(\zeta)\ \mathrm{d}\zeta \approx \sum_{j=1}^{q} a_{ij}\ \Psi(c_j). \tag{25}$$

For instance the classical RK4 method (22) results in the use of Simpson quadrature rule.

It is customary for RK method to impose the condition $\mathrm{A}\begin{pmatrix}1 & \dots & 1\end{pmatrix}^\top = c$, *i.e.*

$$\forall i \in \{1, \dots, q\} \qquad c_i = \sum_{j=1}^{q} a_{ij}. \tag{26}$$

This condition amounts to assuming that the quadrature formula (25) integrate exactly at least polynomials of degree 0. The same requirment for the quadrature formula (24) implies that

$$\sum_{i=1}^{q} b_i = 1. \tag{27}$$

Condition (27) is known to ensure the consistency of the RK method; it is also the condition equation for a RK method to be of order 1 at least (see below).

3.1.2. Local error for RK schemes For a numerical method such as the RK4 method (22), it is usually required that the truncation error (also known as the discretisation error), i.e. the error involved by the approximation of $y(z_{k+1})$, where y denotes the solution to problem (19), by the value y_{k+1} given by (23), can be made as small as desired by using a sufficiently small step-size h_k provided the mapping f is sufficiently regular. A method satisfying this requirement is said to be *convergent*. When studying RK methods, one also introduces the concept of *consistency*. A method designed to solve the initial value problem (19) is said to be *consistent with the initial value problem* when

$$\forall k \in \{0, \dots, K\} \qquad \Phi(z_k, y_k; 0) = f(z_k, y_k). \tag{28}$$

It is well known that for a RK method, a necessary and sufficient condition for consistency is that the sum of the Runge-Kutta weights b_i is equal to 1. The RK4 method (22) is consistent since $\sum_{i=1}^{4} b_i = \frac{1}{6} + \frac{1}{3} + \frac{1}{3} + \frac{1}{6} = 1$. (This property can also be checked from definition (28) by considering the expression of Φ and $\alpha_1, \dots, \alpha_4$ given by (21) when $h_k = 0$.) The interest of consistency is that it is a necessary and sufficient condition for convergence of RK methods. Thus RK4 method (22) is convergent.

In the study of numerical methods for solving initial value problems, the two notions of *local error* and *global error* are used simultaneously. The local error ℓ_{k+1} at grid point z_{k+1} is defined as the error made by using relation (23) as an approximation at grid point z_{k+1} to the solution of the initial value problem

$$\begin{cases} \underline{y}'(z) = f(z, \underline{y}(z)) \quad \forall z \in [z_k, z_{k+1}] \\ \underline{y}(z_k) = y_k \end{cases}. \tag{29}$$

It can be interpreted as the error of the approximation method made over the computational step form z_k to z_{k+1} when assuming that the value of the solution at grid point z_k was exact. According to (23), the local error ℓ_{k+1} is given by

$$\ell_{k+1} = \underline{y}(z_{k+1}) - y_{k+1} = \underline{y}(z_{k+1}) - \underline{y}(z_k) - h_k\,\Phi(z_k, \underline{y}(z_k); h_k). \tag{30}$$

Thus the local error ℓ_{k+1} can be seen as a measure of the way the exact solution of the initial value problem satisfies the approximation formula (23).

The local error is used to define the convergence order of a numerical method used to solve an initial value problem. A method is said to have convergence order p where $p \in \mathbb{N}^*$ if

$$\forall k \in \{0, \ldots, K-1\} \qquad \ell_{k+1} = \mathcal{O}(h^{p+1}) \tag{31}$$

where $h = \max_{k\in\{0,\ldots,K-1\}} h_k$. The RK4 method (22) is known to have convergence order 4 when f is sufficiently smooth [7, 17]. More precisely, one can write the local error for the RK4 method as

$$\ell_{k+1} = \psi_5(z_k, y_k)\, h^5 + \mathcal{O}(h^6) \tag{32}$$

where ψ_5 is a function of the *elementary differentials* of order 5 of the function f involved in the definition of the initial value problem (19) (see section 318 of [7] for a precise expression of ψ_5), evaluated here in y_k. The term $\psi(z_k, y_k)\, h^5$ is known as the principal local truncation error.

Of course, when solving an initial value problem by a numerical method, it is the global error that we are interested in, i.e. the difference between the exact solution and its numerical approximation in each grid point. Namely, the global error e_{k+1} at grid point z_{k+1} is defined as

$$e_{k+1} = y_{k+1} - y(z_{k+1}) \tag{33}$$

where $y(z_{k+1})$ denotes the exact solution and y_{k+1} the approximate solution at the grid point z_{k+1}. Whereas the local error refers to the error made in the current computational step of the method assuming that the initial data used for this step are exact, the global error takes into account the fact that the data comes from computations made in the previous steps and therefore suffers from the error accumulated over these steps. The reason why the local error plays a central role in the study of numerical methods for initial value problem is that it is generally not possible to precisely know the global error. Most of the time the only available information on the global error are bounds which are too large to accurately estimate it. Moreover, it can be shown that under some reasonable assumptions on the function f, the global error can be related to the local error in a simple manner [7, 16].

To be comprehensive about numerical approximation in the RK4 method we have to emphasis the fact the definition of local and global errors assume that exact arithmetic is used. Of course, numerical softwares for solving initial value problems on computer work under floating-point arithmetic. The consequence is that round-off error needs as well to be taken into account in the discussion. However, since round-off error depends on the computer in which the numerical method is implemented it is generally not considered in the numerical analysis of a method. One major effect of round-off error is that it increases in proportion to the number of floating-point operations achieved. In practise, it prevents the local or global errors to tend to zero when the step-size h decreases toward zero as suggested by (32).

To conclude this section on RK4 methods, we may point out one of the reason why fourth-order RK methods are so popular: they involve exactly 4 computational stages (the numbers $\alpha_1, \ldots, \alpha_4$ given by (21) in the case considered here) whereas RK methods of higher order p ($p \geqslant 5$) necessarily involve at least 1 stage more than their order (e.g. a fifth order RK method involves at least 6 stages) [18]. This is of importance since usually the cost of the method is proportional to the number of evaluations of the function f and one evaluation is required in each stage of the RK scheme.

3.2. The fourth order Runge-Kutta Interaction Picture method

In the fourth order Runge-Kutta Interaction Picture (RK4-IP) method, one step of the RK4 scheme (22) is used to approach the solution to problem (8) as follows:

$$\forall t \in \mathbb{R} \qquad A_k^{\mathrm{ip}}(z_{k+1}, t) \approx u_{k+1}^{\mathrm{ip}}(t) \tag{34}$$

where

$$u_{k+1}^{\mathrm{ip}}(t) = A_k^{\mathrm{ip}}(z_k, t) + \frac{h_k}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4\right) \tag{35}$$

and

$$\begin{aligned}
\alpha_1 &= \mathcal{G}_k(z_k, t, A_k^{\mathrm{ip}}(z_k, t)) = \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \mathcal{N}(\exp(-\tfrac{h_k}{2}\mathcal{D}) \cdot A_k^{\mathrm{ip}}(z_k, t)) \\
\alpha_2 &= \mathcal{G}_k(z_k + \tfrac{h_k}{2}, t, A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_1) = \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_1) \\
\alpha_3 &= \mathcal{G}_k(z_k + \tfrac{h_k}{2}, t, A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_2) = \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_2) \\
\alpha_4 &= \mathcal{G}_k(z_k + h_k, t, A_k^{\mathrm{ip}}(z_k, t) + h_k\alpha_3) = \exp(-\tfrac{h_k}{2}\mathcal{D}) \cdot \mathcal{N}(\exp(\tfrac{h_k}{2}\mathcal{D}) \cdot [A_k^{\mathrm{ip}}(z_k, t) + h_k\alpha_3])
\end{aligned}$$

The mapping $t \mapsto A_k^{\mathrm{ip}}(z_k, t)$ is obtained by solving problem (8) and it can be expressed as $A_k^{\mathrm{ip}}(z_k, t) = \exp(\frac{h_k}{2}\mathcal{D}) \cdot A_{k-1}(z_k, t)$. By using the change of unknown (13) we deduce that the mapping $t \mapsto A_k(z_{k+1}, t)$ solution to problem (6) at grid point z_{k+1} can be approximated by u_{k+1} where $\forall t \in \mathbb{R}$

$$u_{k+1}(t) = \exp(\tfrac{h}{2}\mathcal{D}) \cdot u_{k+1}^{\mathrm{ip}}(t) = \exp(\tfrac{h}{2}\mathcal{D}) \cdot \big(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4\right)\big). \tag{36}$$

Actually we are only interested in computing u_{k+1} for all $k \in \{0, \ldots, K-1\}$ to obtain an approximate solution to problem (6). The use of the new unknown A_k^{ip} and his approximation u_{k+1}^{ip} is a go-between in the computational approach. We can therefore recast the above computational procedure as follows to reduce the computational cost of the method. We denote by $t \mapsto u_k(t)$ the approximation of $t \mapsto A(z_k, t)$ the solution at grid point z_k and we successively compute

$$\begin{aligned}
u_k^{\mathrm{ip}}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot u_k(t) \\
\alpha_1 &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \mathcal{N}(u_k(t)) \\
\alpha_2 &= \mathcal{N}(u_k^{\mathrm{ip}}(t) + \tfrac{h_k}{2}\alpha_1) \\
\alpha_3 &= \mathcal{N}(u_k^{\mathrm{ip}}(t) + \tfrac{h_k}{2}\alpha_2) \\
\alpha_4' &= \mathcal{N}(\exp(\tfrac{h_k}{2}\mathcal{D}) \cdot [u_k^{\mathrm{ip}}(t) + h_k\alpha_3]) \\
u_{k+1}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \big(u_k^{\mathrm{ip}}(t) + \tfrac{h_k}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3\right)\big) + \tfrac{h_k}{6}\alpha_4'.
\end{aligned} \tag{37}$$

Computation of the $\exp(\frac{h_k}{2}\mathcal{D})$ operator is done according to the computational procedure presented in section 2.2.2. An important point to be mentioned here concerns the choice of the coefficients $c_1 = 0$, $c_2 = \frac{1}{2}$, $c_3 = \frac{1}{2}$ and $c_4 = 1$ of the RK4 method in the efficiency of the RK4-IP computational procedure given by (37). Indeed, in conjunction with the choice of $z'_k = z_k + \frac{h_k}{2}$ in the change of unknown (13), these particular values of the c_i coefficients enables the cancellation of 4 exponential operator terms in the reformulation (37) compared to other RK4 methods and therefore save up computational time. Moreover this choice of the coefficients c_i values implies that every exponential operator term has precisely the same arguments with only the function upon which the operator acts varying. This permits a more efficient implementation compared to other choices. The cost of the computational procedure (37) actually lies in the evaluations of the $\mathcal{N}$ operator defined by (5). As detailed latter, each evaluation requires 4 calls to the Fast Fourier Transform.

4. Adaptive step-size strategies

First of all one must mention that it doesn't really exist "one" method of reference for local error estimation. Indeed assume the asymptotic behavior of the local error is given by (see relation (32))

$$\ell_{k+1} = \psi_5(z_k, y_k)\, h_k^5 + \psi_6(z_k, y_k)\, h_k^6 + \mathcal{O}(h_k^7) \tag{38}$$

and that an estimator of the leading term $\psi_5(z_k, y_k)\, h^5$ in expression (38) of the local error is available. It may occur under certain circumstances during the adaptive step-size control process, especially when h is not too small, that the leading term is small compared to the other terms in the expression (38). In such a case the value of the estimator does not represent an accurate approximation of the local error and adaptive step-size control may fail.

4.1. Local error estimate by step doubling

The idea behind the step doubling method (also known as Richardson extrapolation method) for estimation of the local error is the following [11]. The local error for the RK4 method (22) at grid point z_{k+1} satisfies the following expansion deduced from relation (32)

$$\forall t \in \mathbb{R} \qquad \ell_{k+1}(t) = \psi_5(t, z_k, u_k)\, h_k^5 + \mathcal{O}(h_k^6). \tag{39}$$

Let u_{k+1} be the solution at grid point z_{k+1} computed from grid point z_k using one step of size h_k, and let $\widetilde{u}_{k+1}$ be the solution at grid point z_{k+1} computed from grid point z_k using two steps of size $h_k/2$, in both cases assuming the solution at grid point z_k to be exact (since we are interested in the local error). Moreover assume that the local error after 2 steps is twice the local error after one step (which is actually true only when h tends to 0 since in this reasoning the initial data for the second half-step is actually the approximate solution computed after the first half-step and not the exact one). Then, $\forall t \in \mathbb{R}$

$$\begin{aligned} A(z_{k+1}, t) - u_{k+1}(t) &= \psi_5(t, z_k, u_k)\, h_k^5 + \mathcal{O}(h_k^6) \\ A(z_{k+1}, t) - \widetilde{u}_{k+1}(t) &= 2\psi_5(t, z_k, u_k) \left(\frac{h_k}{2}\right)^5 + \mathcal{O}(h_k^6) \end{aligned} \tag{40}$$

and therefore by difference between these 2 relations

$$\widetilde{u}_{k+1}(t) - u_{k+1}(t) = \psi_5(t, z_k, u_k)\, h_k^5 \left(1 - \frac{1}{2^4}\right) + \mathcal{O}(h_k^6). \tag{41}$$

Thus the local error at grid point z_{k+1} and time t can approximated, with an error behaviour in $\mathcal{O}(h_k^6)$, in the following way

$$\ell_{k+1}(t) \approx \psi_5(t, z_k, u_k)\, h_k^5 \approx \frac{2^4 - 1}{2^4}\ (\widetilde{u}_{k+1}(t) - u_{k+1}(t)). \tag{42}$$

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

$$L_{k+1} = \|\ell_{k+1}\|_{\mathbb{L}^2(\mathbb{R},\mathbb{C})} \approx \frac{2^4 - 1}{2^4}\ \left(\int_{\mathbb{R}} |\widetilde{u}_{k+1}(t) - u_{k+1}(t)|^2\ \mathrm{d}t\right)^{\frac{1}{2}}. \tag{43}$$

We have to point out that relation (42) gives an approximation of the local error corresponding to the solution computed over the coarse grid. However, the fine mesh grid solution is a better approximation of the solution than the coarse mesh grid solution and is thus kept as the approximate solution. The cost of estimating the local error is then the cost of the computation of the coarse mesh grid solution and this cost is approximately half the cost of the computation of the fine mesh grid solution since the step-size is twice larger. Thus, estimating the local error using the step doubling approach is liable of an extra computational cost of 50% more than the cost of the computation of the approximate solution itself. In fact, the additional cost is slightly less than 50% since some computations are shared by each of the 2 methods, and needs to be carried out only once.

Actually, relation (42) gives an approximation of the principal part of the Taylor expansion (40) of the local error. The principal local truncation error is usually large in comparison with the other terms involved in the expression of the local truncation error, which justifies the use of the principal local truncation error to set the step length. However, one must be aware that under special circumstances this can not be the case and the other terms in the expression of the local truncation error can overwhelm the principal local truncation error [16].

Another approach for estimating the error made over one computational step relies on the concept of energy conservation. We detail it now.

4.2. Local error estimate by the Conservation Quantity Error method

In [19] it is shown that when loss in the fiber is neglected (i.e. in the case when $\alpha = 0$ in eq. (2)) the following quantity, referred as the "Optical Photon Number" (OPN), is conserved for the GNLSE

$$P(z) = \int_{\mathbb{R}} \frac{n_{\mathrm{eff}}\, A_{\mathrm{eff}}}{\omega}\ |\widehat{A}(z,\omega)|^2\ \mathrm{d}\omega \tag{44}$$

where A_{eff} is known as the effective mode area is defined from the modal distribution F (see relation (1)) as

$$A_{\mathrm{eff}} = \frac{\left(\iint_{\mathbb{R}^2} |F(x,y)|^2\, dxdy\right)^2}{\iint_{\mathbb{R}^2} |F(x,y)|^4\ \mathrm{d}x\ \mathrm{d}y}$$

and integration in (44) hold over the entire spectrum of the optical wave amplitude assumed to have a bandwidth less than $\frac{\omega_0}{3}$. Moreover as justified in [19], in a low-loss fiber waveguide, the frequency dependence of the coefficient $n_{\text{eff}}\, A_{\text{eff}}$ is extremely weak and, to a good approximation, this dependence can be neglected and the term assumed to be a constant.

In [12] it is made use of the conservation of the OPN to propose a cheap numerical method, termed the Conservation Quantity Error (CQE) method, to estimate the local error in order to set an adaptive step-size control strategy. The justification of the method is the following. The OPN error is defined as

$$\delta P(z_{k+1}) = P(z_{k+1}) - \widetilde{P}(z_{k+1}) = \int_{\mathbb{R}} \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega} \left(|\widehat{A}(z_{k+1},\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega \quad (45)$$

where $P(z_{k+1})$ denotes the OPN computed from (44) by using the exact expression of the electric filed amplitude A and $\widetilde{P}(z_{k+1})$ denotes the approximation of the OPN computed from (44) by using the approximate solution u_k given by (37). On the one hand, if we defined the "local intensity error" as

$$\ell^{\text{int}}_{k+1}(\omega) = |\widehat{A}(z_{k+1},\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2$$

then the OPN error is related to the local intensity error as

$$\delta P(z_{k+1}) = \int_{\mathbb{R}} \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega} \, \ell^{\text{int}}_{k+1}(\omega) \, \mathrm{d}\omega. \quad (46)$$

Thus the Photon Number error is the weighted integral of the local intensity error integrated over all the frequency window for the weight $s(\omega) = \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega}$.

On the other hand, since the OPN is assumed to be a conserved quantity $P(z_{k+1}) = P(z_k)$ and we have

$$\begin{aligned} \delta P(z_{k+1}) &= P(z_{k+1}) - \widetilde{P}(z_{k+1}) = P(z_k) - \widetilde{P}(z_{k+1}) \\ &= \int_{\mathbb{R}} \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega} \left(|\widehat{A}(z_k,\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega. \end{aligned} \quad (47)$$

Since we are interested by a measure of the local error over the interval $[z_k, z_{k+1}]$, it is assumed that $A(z_k,\cdot) = u_k$ and therefore $\widehat{A}(z_k,\cdot) = \widehat{u}_k$ (see section 3.1). As a consequence we have

$$\delta P(z_{k+1}) = \int_{\mathbb{R}} \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega} \left(|\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega. \quad (48)$$

Thus, from a computational point of view the OPN error can be obtained by evaluating the weighted integral of the difference of the square modulus of the Fourier transform of the approximated electric field amplitude between z_k and z_{k+1}. Finally, from (46) and (48) the weighted integral of the local intensity error ℓ^{int}_{k+1} can be computed from the Fourier transform of the electric field amplitude computed at grid point z_k and z_{k+1} as

$$\begin{aligned} L^{\text{int}}_{k+1} &= \int_{\mathbb{R}} \frac{1}{\omega} \, \ell^{\text{int}}_{k+1}(\omega) \, \mathrm{d}\omega = \frac{1}{n_{\text{eff}}\, A_{\text{eff}}} \, \delta P(z_{k+1}) \\ &= \int_{\mathbb{R}} \frac{1}{\omega} \left(|\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega. \end{aligned} \quad (49)$$

The point now is to relate the local intensity error ℓ_{k+1}^{int} to the local error ℓ_{k+1} defined in (30). The local error satisfies, see (32),

$$\forall t \in \mathbb{R} \quad \ell_{k+1}(t) = A(z_{k+1}, t) - u_{k+1}(t) = \psi_5(t, z_k, u_k)\, h_k^5 + \mathcal{O}(h_k^6)$$

and by Fourier transform we obtain

$$\forall t \in \mathbb{R} \quad \widehat{\ell}_{k+1}(\omega) = \widehat{A}(z_{k+1}, \omega) - \widehat{u}_{k+1}(\omega) = \widehat{\psi}_5(\omega, z_k, \widehat{u}_k)\, h_k^5 + \mathcal{O}(h_k^6).$$

Therefore we have

$$\begin{aligned} \ell_{k+1}^{\text{int}}(\omega) &= |\widehat{A}(z_{k+1}, \omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 = |\widehat{u}_{k+1}(\omega) + \widehat{\ell}_{k+1}(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \\ &= 4\mathcal{R}\mathrm{e}\big(\widehat{u}_{k+1}(\omega)\, \widehat{\ell}_{k+1}(\omega)\big) + |\widehat{\ell}_{k+1}(\omega)|^2. \end{aligned} \tag{50}$$

In particular, we deduce the following asymptotic behaviour for the local intensity error

$$\ell_{k+1}^{\text{int}}(\omega) = \phi(\omega, z_k, \widehat{u}_k)\, h_k^5 + \mathcal{O}(h_k^6).$$

Local intensity error ℓ_{k+1}^{int} and local error ℓ_{k+1} have the same asymptotic behaviour and are connected through relation (50).

In [12] it is proposed to extend the CQE method to linear loss fibers by using the following approximation formula deduced from a first order Taylor formula expansion

$$\delta P(z_{k+1}) = P(z_{k+1}) - \widetilde{P}(z_{k+1}) = P(z_k) - \widetilde{P}(z_{k+1}) + h_k\, \frac{\partial P}{\partial z}(z_k) + \mathcal{O}(h_k^2) \tag{51}$$

where the OPN change in the presence of linear loss is expressed as

$$\frac{\partial P}{\partial z}(z_k) = -\int_{\mathbb{R}} \alpha(\omega) \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega}\, |\widehat{A}(z_k, \omega)|^2 \, \mathrm{d}\omega.$$

Thus, in order to take into account fiber losses relation (48) is modified in

$$\delta P(z_{k+1}) \approx \int_{\mathbb{R}} \frac{n_{\text{eff}}\, A_{\text{eff}}}{\omega} \left((1 - h_k\, \alpha(\omega)) |\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega \tag{52}$$

and from (51) the approximation is first order accurate. Finally, the weighted integral of the local intensity error ℓ_{k+1}^{int} is computed from the Fourier transform of the electric field amplitude computed at grid point z_k and z_{k+1} as

$$L_{k+1}^{\text{int}} = \int_{\mathbb{R}} \frac{1}{\omega}\, \ell_{k+1}^{\text{int}}(\omega) \, \mathrm{d}\omega = \int_{\mathbb{R}} \frac{1}{\omega} \left((1 - h_k\, \alpha(\omega)) |\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2 \right) \, \mathrm{d}\omega. \tag{53}$$

4.3. Local error estimate by the Modified Conservation Quantity Error method

In equation (2), energy losses are mainly related to the term αA where A is the coefficient of linear attenuation. An alternate method for estimating the fiber losses consists in considering the following simplified equation coming from (2) when only the linear losses are taken into account:

$$\frac{\partial}{\partial z} A(z, t) = -\frac{\alpha}{2} A(z, t). \tag{54}$$

This ODE equation with t as a parameter is considered over one grid interval $[z_k, z_{k+1}]$. The solution at grid point z_{k+1} can be expressed from its values at grid point z_k as

$$\forall t \in \mathbb{R} \quad A(z_{k+1}, t) = A(z_k, t)\, \mathrm{e}^{-\frac{\alpha}{2}(z_{k+1}-z_k)}. \tag{55}$$

From (44) we deduce that

$$P(z_{k+1}) = \int_{\mathbb{R}} \frac{n_{\mathrm{eff}}\, A_{\mathrm{eff}}}{\omega}\, |\widehat{A}(z_{k+1}, \omega)|^2\, \mathrm{d}\omega = \mathrm{e}^{-\alpha(z_{k+1}-z_k)} P(z_k). \tag{56}$$

Now, from (45) we have

$$\delta P(z_{k+1}) = \int_{\mathbb{R}} \frac{n_{\mathrm{eff}}\, A_{\mathrm{eff}}}{\omega} \left(\mathrm{e}^{-\alpha h_k}\, |\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2\right)\, \mathrm{d}\omega \tag{57}$$

Of course, by using the first order Taylor expansion $\mathrm{e}^x = 1 + x + \mathcal{O}_0(x^2)$, we obtain from (57) the approximation (52) used in the CQE method. Thus with the Modified Conservation Quantity Error (MCQE) method, the weighted integral of the local intensity error $\ell^{\mathrm{int}}_{k+1}$ is computed from the Fourier transform of the electric field amplitude computed at grid point z_k and z_{k+1} as

$$L^{\mathrm{int}}_{k+1} = \int_{\mathbb{R}} \frac{1}{\omega}\, \ell^{\mathrm{int}}_{k+1}(\omega)\, \mathrm{d}\omega = \int_{\mathbb{R}} \frac{1}{\omega} \left(\mathrm{e}^{-\alpha h_k}\, |\widehat{u}_k(\omega)|^2 - |\widehat{u}_{k+1}(\omega)|^2\right)\, \mathrm{d}\omega. \tag{58}$$

4.4. Local error estimate by using an embedded Runge-Kutta method

4.4.1. Overview of embedded Runge-Kutta methods Embedded Runge-Kutta (ERK) methods are special Runge-Kutta (RK) methods designed to deliver two approximations of the solution of the initial value problem under consideration, corresponding to 2 RK schemes of different convergence orders p and q such that $q > p$. These 2 approximations of the solution can be considered as an accurate approximate solution (the one computed with the numerical scheme of higher order) and a coarse approximate solution (the one computed with the one of lower order).

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, see section 3.1), we denote by u_{k+1} (resp. $\widetilde{u}_{k+1}$) the coarse (resp. accurate) approximate solution found at the current grid point z_{k+1}. The local errors for each of the 2 methods are respectively given by

$$\begin{aligned} \ell^{[p]}_{k+1} &= A(z_{k+1}, t) - u_{k+1}(t) = \psi_p(t, z_k, u_k)\, h_k^{p+1} + \mathcal{O}(h_k^{p+2}) \\ \ell^{[q]}_{k+1} &= A(z_k, t) - \widetilde{u}_{k+1}(t) = \psi_q(t, z_k, u_k)\, h_k^{q+1} + \mathcal{O}(h_k^{q+2}) \end{aligned} \tag{59}$$

where ψ_p (resp. ψ_q) is a function of the elementary differentials of order p (resp. q) [7, 8]. By difference of these 2 relations we obtain

$$\widetilde{u}_{k+1}(t) - u_{k+1}(t) = \psi_p(t, z_k, u_k)\, h_k^{p+1} + \mathcal{O}(h_k^{p+2}). \tag{60}$$

Since $q > p$ the quantity $\psi_q(t, z_k)\, h^{q+1} + \mathcal{O}(h^{q+2})$ is considered through the remainder $\mathcal{O}(h^{p+2})$. Thus the local error for the RK method of lower order at grid point z_k can be approximated, with an error $\mathcal{O}(h^{p+2})$, in the following way

$$\forall t \in \mathbb{R} \qquad \ell^{[p]}_{k+1}(t) \approx \psi_p(t, z_k, u_k)\, h^{p+1} + \mathcal{O}(h_k^{p+2}) \approx \widetilde{u}_{k+1}(t) - u_{k+1}(t). \tag{61}$$

In general, ERK methods are constructed with $q = p+1$. One of the most famous ERK method is the Fehlberg 4(5) pair [7,20]. It has 6 stages and delivers a RK method of order 4 with an error estimate computed from a fifth order RK method.

We have to mention here that even if the local error estimate (61) holds only for the lower order method it is customary in practise to use values given by the higher order method as the approximation of the solution at grid point z_k. This is sometimes interpreted as *local extrapolation* in the sense that one can consider that the error estimate is added to the approximate solution as a correction according to the relation

$$\begin{aligned} \widetilde{u}_{k+1}(t) &= A(t, z_{k+1}) + \mathcal{O}(h_k^{p+2}) = u_{k+1}(t) + \psi_p(t, z_k, u_k) h_k^{p+1} + \mathcal{O}(h_k^{p+2}) \\ &= u_{k+1}(t) + \ell_{k+1}^{[p]}(t) + \mathcal{O}(h_k^{p+2}). \end{aligned} \tag{62}$$

While in such a case relation (61) is still used for step-size control purposes it is no longer related asymptotically to the local error. One can report that there exists some ERK methods such as the method of Dormand and Prince [21] that are designed to minimise the local error of the higher order solution.

To conclude with this overview of ERK methods, one has to point out that the main idea behind the concept of ERK pairs is of course to have a large part of the internal computations of the 2 RK scheme in common in order to have a computational cost much lower than the one required when using 2 arbitrary RK methods of order p and q.

4.4.2. Embedded Runge-Kutta methods for the IP method When looking for an ERK method for using in conjunction with the Interaction Picture method, 2 different approaches can be explored.

- the first one would be to look for a fourth order RK method embedded in a fifth order RK method in order to design an adaptive step-size strategy based on the estimation of the local error of the fourth order RK method.
- the second one would be to look for a third order RK method embedded in a fourth order RK method and to use the local extrapolation idea to propagate the solution computed with the fourth order RK method.

As mentioned before the main drawback of the first approach lies in the number of stages required by a fifth order RK method which is at least 6 stages. Even if part of the stages are in common between the 2 embedded methods, this approach implies a significant extra cost of at least 2 stages. In the situation considered here, each stage of the RK method requires one evaluation of the operator $\mathcal{N}$ and accounts for 4 Fast Fourier Transform calls. The excess is therefore of 8 Fast Fourier Transform calls per step which can be considered as prohibitive. For this reason, we will consider here the second above approach. There is an infinite number of third order RK method embedded in a fourth order RK method and requiring 4 computational stages, see e.g [7] section 333. Now, we have to keep in mind that the efficiency of the RK4-IP algorithm is partially due to the values of the c_i coefficients of the RK4 scheme ($c_1 = 0, c_2 = 1/2, c_3 = 1/2$ and $c_4 = 1$). These values are capitalised to reduce the number of exponential operator terms to be computed. Other choices for the values of the coefficients c_i would lead to a larger number of exponential operator terms involved and therefore to an increase of the computational cost. We have to look for an embedded Runge-Kutta method with this constrain in mind. As well we

will only consider here explicit Runge-Kutta methods because there is no reason why using implicit Runge-Kutta methods designed mainly for stiff ordinary differential equations, and because they are much expensive. Additionally, we will assume the weight coefficients $b_i, i = 1, \ldots, 4$ are nonnegative real numbers which corresponds to an usual simplification assumption in the study of RK method. in order to avoid round off error issues for the method.

4.4.3. First attempt for a 4 stages embedded RK4 method Our search of an embedded Runge-Kutta method to be used in conjunction with the IP method starts with a Butcher tableau of the following form for the higher order method

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & a_{21} & & & \\ \frac{1}{2} & a_{31} & a_{32} & & \\ 1 & a_{41} & a_{42} & a_{43} & \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} \tag{63}$$

We recall that the coefficients $c_1 = 0, c_2 = 1/2, c_3 = 1/2$ and $c_4 = 1$ are imposed in order to preserve the efficiency of the numerical implementation of the interaction picture method.

The conditions for this Butcher tableau to define a fourth order RK method are the following [7, 8, 17]. The condition for first order accuracy of the RK method (63) reads

$$b_1 + b_2 + b_3 + b_4 = 1. \tag{64}$$

The condition for the second order accuracy of the RK method (63) reads

$$\frac{1}{2}b_2 + \frac{1}{2}b_3 + b_4 = \frac{1}{2}. \tag{65}$$

There are 2 conditions for the RK method (63) to be at least of the third order :

$$\frac{1}{4}b_2 + \frac{1}{4}b_3 + b_4 = \frac{1}{3} \tag{66}$$

$$\frac{1}{2}a_{32}b_3 + \frac{1}{2}a_{42}b_4 + \frac{1}{2}a_{43}b_4 = \frac{1}{6} \tag{67}$$

If we consider the set formed by the 3 first conditions, we easily obtain by solving the linear system formed by the 3 equations (64)-(65)-(66) that we must necessarily have

$$b_1 = \frac{1}{6}, \quad b_4 = \frac{1}{6} \quad \text{and} \quad b_2 + b_3 = \frac{2}{3}. \tag{68}$$

There are 4 conditions for a RK method to be at least of the fourth order. In the case of the RK method (63) with the specific case corresponding to the values given by (68) they reduce to the following 3 conditions (one of the 4 conditions is automatically satisfied)

$$\begin{aligned} \frac{1}{4}a_{32}b_3 + \frac{1}{12}a_{42} + \frac{1}{12}a_{43} &= \frac{1}{8} \\ \frac{1}{4}a_{32}b_3 + \frac{1}{24}a_{42} + \frac{1}{24}a_{43} &= \frac{1}{12} \\ \frac{1}{12}a_{43}a_{32} &= \frac{1}{24} \end{aligned} \tag{69}$$

From condition (26) we deduce that 4th order RK methods with $c = \begin{pmatrix}0 & 1/2 & 1/2 & 1\end{pmatrix}^\top$ all have Butcher tableaux in the form

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & \frac{1}{2}-\frac{1}{6\lambda} & \frac{1}{6\lambda} & & \\ 1 & 0 & 1-3\lambda & 3\lambda & \\ \hline & \frac{1}{6} & \frac{2}{3}-\lambda & \lambda & \frac{1}{6} \end{array} \tag{70}$$

The standard RK method corresponds to the value $\lambda = \frac{1}{3}$.

Now in order to have a third order RK method embedded in the fourth order RK method defined by the Butcher tableau (63) we have 2 possibilities. The first one is to look for a third order RK method with a Butcher tableau of the following form

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & a_{21} & & & \\ \frac{1}{2} & a_{31} & a_{32} & & \\ 1 & a_{41} & a_{42} & a_{43} & \\ \hline & b'_1 & b'_2 & b'_3 & b'_4 \end{array} \tag{71}$$

This third order RK method would have 4 stages in common with the fourth order RK method (63). The alternative would be to look for a third order RK method with a Butcher tableau of the following form

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & a_{21} & & \\ \frac{1}{2} & a_{31} & a_{32} & \\ \hline & b''_1 & b''_2 & b''_3 \end{array} \tag{72}$$

Let us examine the first possibility. For the method to be of order 3, we must have by analogy with (67) and (68),

$$b'_1 = \frac{1}{6}, \quad b'_4 = \frac{1}{6} \quad \text{and} \quad b'_2 + b'_3 = \frac{2}{3} \tag{73}$$

and

$$\frac{1}{2}a_{32}b'_3 + \frac{1}{12}a_{42} + \frac{1}{12}a_{43} = \frac{1}{6}. \tag{74}$$

From (67) we must also have

$$\frac{1}{2}a_{32}b_3 + \frac{1}{12}a_{42} + \frac{1}{12}a_{43} = \frac{1}{6}. \tag{75}$$

It follows that, unless $a_{32} = 0$, we necessarily have $b'_3 = b_3$ and as a consequence $b'_2 = b_2$. This means that the third order method we are looking for would actually coincide with the fourth order method it is embedded in and therefore it has no use for local error estimate. We must therefore impose $a_{32} = 0$. But with this choice, the fourth order conditions are satisfied by the 2 RK method (63) and (71) and therefore again the 2 methods can not be used for local error estimate purposes. We conclude

that we can not find a third order method defined by Butcher tableau (71) when the method defined by Butcher tableau (63) is of order 4.

Let us now consider the existence of a third order method defined by Butcher tableau (72). The conditions for the method to be at least of order 2 read

$$\begin{aligned} b_1'' + b_2'' + b_3'' &= 1 \\ \frac{1}{2}b_2'' + \frac{1}{2}b_3'' &= \frac{1}{2} \end{aligned} \tag{76}$$

There are 2 conditions for the method to be of the third order

$$\begin{aligned} \frac{1}{4}b_2'' + \frac{1}{4}b_3'' &= \frac{1}{3} \\ \frac{1}{2}a_{32}b_3'' &= \frac{1}{6} \end{aligned} \tag{77}$$

It is clear that the second condition in (76) is incompatible with the first condition in (77). It means that there is no third order method defined by Butcher tableau (72) but only a second order method. Second order methods with Butcher tableau (72) are given by the conditions $b_1'' = 0$ and $b_2'' + b_3'' = 1$ whatever are the coefficients a_{ij}.

Therefore low order RK methods embedded in the standard RK4 method are necessarily of maximum order 2 and a common choice is the "standard" second order RK method given by the following Butcher tableau

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ \hline & 0 & 0 & 1 \end{array} \tag{78}$$

Note that other fourth order RK methods where the coefficients satisfy conditions (68) and (69) could also be used with almost the same advantages. Actually it is usual to choose the value of the free coefficients in order to minimize the leading terms of the local error asymptotic expansion of either the lower or the higher order RK formula.

We conclude that the only 4th order RK formula with 4 stages that embeds a lower RK formula for local error estimation purposes is the RK4(2) formula given by Butcher tableaux (22) and (78) summarized as follows

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ \hline 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{79}$$

4.4.4. A 5 stages embedded RK4 method It follows from the previous study that the quest for a 3rd order RK method to be used in conjunction with the standard RK4 method defined by Butcher tableau (22) necessarily implies a 5 stages method for the 3rd order RK method. In order to have the 4th order 4 stages RK method embedded in the 3rd order 5 stages RK method, we look for a 3rd order RK method defined by a Butcher tableau of the following form:

$$\begin{array}{c|ccccc} 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & & \\ 1 & 0 & 0 & 1 & & \\ c_5 & a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 \end{array} \tag{80}$$

where the free coefficients $a_{5,j}$, $j = 1, \ldots, 4$ and b_j, $j = 1, \ldots, 4$ have to be determined in order to improve the computational efficiency of the method. Note that we necessarily have $c_5 = \sum_{j=1}^{4} a_{5,j}$ and therefore the value of c_5 is imposed by the value of the other coefficients.

The computational sequence for one step of the embedded RK method becomes (see (35) for comparison)

$$\begin{aligned} u_k^{\mathrm{ip}}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot u_k^{[4]}(t) \\ \alpha_1 &= \mathcal{G}_k(z_k, t, u_k^{\mathrm{ip}}(z_k, t)) = \exp(\tfrac{h_k}{2}\mathcal{D})\, \mathcal{N}(u_k^{[4]}(t)) \\ \alpha_2 &= \mathcal{G}_k(z_k + \tfrac{h_k}{2}, t, u_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_1) = \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_1) \\ \alpha_3 &= \mathcal{G}_k(z_k + \tfrac{h_k}{2}, t, u_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_2) = \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_2) \\ \alpha_4 &= \mathcal{G}_k(z_k + h_k, t, u_k^{\mathrm{ip}}(z_k, t) + h_k\alpha_3) \\ &= \exp(-\tfrac{h_k}{2}\mathcal{D}) \cdot \mathcal{N}\left(\exp(\tfrac{h_k}{2}\mathcal{D}) \cdot [u_k^{\mathrm{ip}}(z_k, t) + h_k\alpha_3]\right) \\ \alpha_5 &= \mathcal{G}_k(z_k + c_5 h_k, t, u_k^{\mathrm{ip}}(z_k, t) + h_k \sum_{j=1}^{4} \alpha_j a_{5,j}) \\ &= \exp(-(c_5 - \tfrac{1}{2})h_k\mathcal{D}) \cdot \mathcal{N}\Big(\exp((c_5 - \tfrac{1}{2})h_k\mathcal{D}) \cdot [u_k^{\mathrm{ip}}(z_k, t) + h_k \sum_{j=1}^{4} \alpha_j a_{5,j}]\Big) \\ u_{k+1}^{[3]}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \big[u_k^{\mathrm{ip}}(t) + h_k \sum_{j=1}^{5} b_j \alpha_j\big] \\ u_{k+1}^{[4]}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \big[u_k^{\mathrm{ip}}(t) + \tfrac{h_k}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4\right)\big] \end{aligned} \tag{81}$$

where $t \mapsto u_k^{[4]}(t)$ (resp. $t \mapsto u_k^{[3]}(t)$) denotes the RK4 (resp. RK3) approximation of $t \mapsto A(z_k, t)$ the solution at grid point z_k.

By choosing $a_{51} = \frac{1}{6}$, $a_{52} = \frac{1}{3}$, $a_{53} = \frac{1}{3}$ and $a_{54} = \frac{1}{6}$ and therefore $c_5 = \sum_{j=1}^{4} a_{5,j} = 1$, the same term appears in the expression of α_5 as well as in the expression of $u_{k+1}^{[4]}$ and this can be capitalized in order to reduce the computations. Embedded RK methods having this feature are said to satisfy the FSAL property (First Step At Last). Thus it remains to determine the values of b_j, $j = 1, \ldots, 5$ in order that the second RK formula has order 3 (but not order 4). There are 4 condition equations for the method to be of order 3 [7, 21]:

$$\begin{cases} b_1 + b_2 + b_3 + b_4 + b_5 = 1 \\ \dfrac{1}{2}b_2 + \dfrac{1}{2}b_3 + b_4 + b_5 = \dfrac{1}{2} \\ \dfrac{1}{4}b_2 + \dfrac{1}{4}b_3 + b_4 + b_5 = \dfrac{1}{3} \\ \dfrac{1}{4}b_3 + \dfrac{1}{2}b_4 + \dfrac{1}{2}b_5 = \dfrac{1}{6} \end{cases}. \tag{82}$$

The solution $(b_1, b_2, b_3, b_4, b_5)$ to this under-determined linear system is

$$\big(\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6} - b_5, b_5\big) \quad \text{where } b_5 \text{ is arbitrarily chosen} \tag{83}$$

One can check that whatever is the value of b_5 there are at least 2 out of 4 condition equations for the fourth order not satisfied. Therefore, we obtain a family of ERK4(3) methods indexed by a free parameter λ defined by the following Butcher tableau

$$\begin{array}{c|ccccc}
0 & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & & \\
1 & 0 & 0 & 1 & & \\
\hline
1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} - \lambda & \lambda
\end{array} \tag{84}$$

This ERK4(3) method actually coincides with Dormand and Prince *Runge-Kutta 4(3) T formula* [22]. Suitable value for λ suggested in [22] is $\lambda = \frac{1}{10}$.

The computational sequence (81) for one step of the ERK4(3) method can be improved. As before one can save the computation of the $\exp(-\frac{h_k}{2}\mathcal{D})$ term involved in the expression of α_4 and α_5 since a cancellation happens with the $\exp(\frac{h_k}{2}\mathcal{D})$ term in the expression of $u_{k+1}^{[3]}$ and $u_{k+1}^{[4]}$. Moreover, although the ERK4(3) method appears as a 5 stages method, is effective cost is very similar to a 4 stages method since the computation of the first coefficient α_1 at step $k+1$ shares the evaluation of the nonlinear operator $\mathcal{N}$ in common with the coefficient α_5 computed at step k. Namely the computational procedure (81) can be recast as follows

$$\begin{aligned}
u_k^{\mathrm{ip}}(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot u_k^{[4]}(t) \\
\alpha_1 &= \exp(\tfrac{h_k}{2}\mathcal{D})\ \alpha'_{5,k} \\
\alpha_2 &= \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_1) \\
\alpha_3 &= \mathcal{N}(A_k^{\mathrm{ip}}(z_k, t) + \tfrac{h_k}{2}\alpha_2) \\
\alpha'_4 &= \mathcal{N}\left(\exp(\tfrac{h_k}{2}\mathcal{D}) \cdot [A_k^{\mathrm{ip}}(z_k, t) + h_k\alpha_3]\right) \\
r(t) &= \exp(\tfrac{h_k}{2}\mathcal{D}) \cdot \left[u_k^{\mathrm{ip}}(t) + \tfrac{h_k}{6}\left(\alpha_1 + 2\alpha_2 + 2\alpha_3\right)\right] \\
u_{k+1}^{[4]}(t) &= r(t) + \tfrac{h_k}{6}\alpha'_4 \\
\alpha'_{5,k+1} &= \mathcal{N}(u_{k+1}^{[4]}) \\
u_{k+1}^{[3]}(t) &= r(t) + \tfrac{h_k}{30}(2\alpha'_4 + 3\alpha'_{5,k+1})
\end{aligned} \tag{85}$$

We will detail in the next section the way the local error can be estimated from the values $u_{k+1}^{[3]}$ and $u_{k+1}^{[4]}$. Compared to the computational procedure (37) of the standard RK4 scheme for the IP method, the propound computational procedure (85) has a very similar computational cost even if the embedded RK method has 5 stages: the number of evaluations of the nonlinear operator $\mathcal{N}$ is 4 in both case. The extra cost is limited to 2 additions and 3 multiplications and the need to keep in memory 2 intermediate results.

4.4.5. Local error estimate for the RK4-IP method We consider the embedded Runge-Kutta method given by Butcher tableaux (84). 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), we denote by $u_k^{[3]}$ (resp. $u_k^{[4]}$) the approximate solution computed at the current grid point z_k by the third order (resp. the fourth order) RK method. The local errors at grid point z_{k+1} and time t for each of the 2 methods are respectively given by

$$\begin{aligned} \ell_{k+1}^{[3]}(t) &= A(z_{k+1},t) - u_{k+1}^{[3]}(t) = \psi_3(t,z_k,u_k^{[3]})\, h_k^4 + \mathcal{O}(h_k^5) \\ \ell_{k+1}^{[4]}(t) &= A(z_{k+1},t) - u_{k+1}^{[4]}(t) = \psi_4(t,z_k,u_k^{[4]})\, h_k^5 + \mathcal{O}(h_k^6) \end{aligned} \tag{86}$$

and by difference of these 2 relations we obtain

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

Thus the local error for the third 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

$$\ell_{k+1}^{[3]}(t) \approx \psi_3(t,z_k,u_k^{[3]})\, h_k^4 + \mathcal{O}(h_k^5) \approx u_{k+1}^{[4]}(t) - u_{k+1}^{[3]}(t). \tag{87}$$

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

$$\begin{aligned} L_{k+1}^{[3]} = \|\ell_{k+1}^{[3]}\|_{\mathbb{L}^2(\mathbb{R},\mathbb{C})} = \|\widehat{\ell_{k+1}^{[3]}}\|_{\mathbb{L}^2(\mathbb{R},\mathbb{C})} &\approx \left(\int_{\mathbb{R}} \left| \widehat{u}_{k+1}^{[4]}(t) - \widehat{u}_{k+1}^{[3]}(t) \right|^2 \, \mathrm{d}t \right)^{\frac{1}{2}} \\ &\approx \left(h_t \sum_{j=0}^{J-1} \left| \widehat{u}_{k+1}^{[4]}(t_j) - \widehat{u}_{k+1}^{[3]}(t_j) \right|^2 \right)^{\frac{1}{2}} \end{aligned} \tag{88}$$

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

As mentioned before, even if the local error estimate (87) holds only for the third order method, in practise we use the value given by the fourth order method as the approximation of the solution at grid point z_{k+1}. In general, this approach overestimates the actual local error, which is safe but not of optimal efficiency.

5. Algorithms and experimental comparison

5.1. The basic RK4-IP algorithm for solving the GNLSE

We present in this section the basic RK4-IP algorithm for solving the GNLSE, *i.e.* the RK4-IP algorithm without step-size control. First, as shown in the previous sections, the computation of the $\exp(\frac{h}{2}\mathcal{D}_t)$ and nonlinear terms requires several uses of the Fourier transform and inverse Fourier transform. It is therefore important to recast the computational procedure given in (37) in order to reduce the number of Fourier transforms to achieve.

For convenience, we introduce the following notations. For a mapping $f : \mathbb{R} \to \mathbb{C}$, we denote by $\widehat{\mathcal{N}}(\widehat{f}(\nu))$ the Fourier transform $\mathcal{F}[t \mapsto \mathcal{N}(f(t))](\nu)$, by $t \mapsto u_k(t)$ the approximation of the electric pulse amplitude A at grid point z_k and by $\widehat{u}_k$ its Fourier transform. Taking into account relation (15) we transform the computational

sequence (37) in the following way:

$$\begin{aligned}
\widehat{u}_k^{\mathrm{ip}}(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h}{2}} \times \widehat{u}_k(\nu) \\
\widehat{\alpha}_1(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h}{2}} \times \widehat{\mathcal{N}}(\widehat{u}_k(\nu)) \\
\widehat{\alpha}_2(\nu) &= \widehat{\mathcal{N}}\big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h}{2}\widehat{\alpha}_1(\nu)\big) \\
\widehat{\alpha}_3(\nu) &= \widehat{\mathcal{N}}\big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h}{2}\widehat{\alpha}_2(\nu)\big) \\
\widehat{\alpha}_4'(\nu) &= \widehat{\mathcal{N}}\big(\mathrm{e}^{\widehat{d}_\nu \frac{h}{2}} \times \widehat{u}_k^{\mathrm{ip}}(\nu) + h\widehat{\alpha}_3(\nu)\big)
\end{aligned} \tag{89}$$

and

$$\begin{aligned}
\widehat{u}_{k+1}(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h}{2}} \times \big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h}{6}\left(\widehat{\alpha}_1(\nu) + 2\widehat{\alpha}_2(\nu) + 2\widehat{\alpha}_3(\nu)\right)\big) + \tfrac{h}{6}\widehat{\alpha}_4'(\nu) \\
u_{k+1}(t) &= \mathcal{F}^{-1}\big(\nu \mapsto \widehat{u}_{k+1}(\nu)\big)(t)
\end{aligned} \tag{90}$$

This reformulation of the computational procedure gives rise to the following algorithm for solving the GNLSE (2) by the Interaction Picture method in conjunction with the RK4 method.

RK4-IP algorithm

Input: Array u containing the sampling of the signal amplitude at the fibre entrance
Array $[\nu_j]_{j=1,\ldots,N}$ containing the frequency sampling
Array $[z_k]_{k=0,\ldots,K}$ containing the spatial grid points
Array $\widehat{h}_R$ containing the sampling of the Fourier transform of the Raman response function

Output: Array u containing the sampling of the signal amplitude at the fibre end

{Initialisation}
for $j = 1, \ldots, N$ **do**
  $\widehat{d}[j] \leftarrow -\frac{1}{2}\alpha + \mathrm{i}\sum_{n=2}^{n_{max}} \frac{\beta_n}{n!}(2\pi\nu_j)^n$
  $tfexpd[j] \leftarrow \exp(\frac{h}{2}\widehat{d}[j])$
end for
$\widehat{u}_1 \leftarrow \mathrm{FFT}(u, forward)$
{Loop over the propagation sub-interval}
for $k = 1, \ldots, K$ **do**
  for $j = 1, \ldots, N$ **do**
    $\widehat{u}_{ip}[j] \leftarrow tfexpd[j] \times \widehat{u}_1[j]$
  end for
  $\widehat{\alpha}_1 \leftarrow \mathrm{COMPUTE\_TFN}(u_1)$
  for $j = 1, \ldots, N$ **do**
    $\widehat{\alpha}_1[j] \leftarrow tfexpd[j] \times \widehat{\alpha}_1[j]$
    $\widehat{u}_2[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_1[j]$
  end for
  $u_2 \leftarrow \mathrm{FFT}(\widehat{u}_2, backward)$
  $\widehat{\alpha}_2 \leftarrow \mathrm{COMPUTE\_TFN}(u_2)$
  for $j = 1, \ldots, N$ **do**
    $\widehat{u}_3[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_2[j]$
  end for
  $u_3 \leftarrow \mathrm{FFT}(\widehat{u}_3, backward)$
  $\widehat{\alpha}_3 \leftarrow \mathrm{COMPUTE\_TFN}(u_3)$

for $j = 1, \ldots, N$ **do**
$\widehat{u}_4[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + h\widehat{\alpha}_3[j])$
end for
$u_4 \leftarrow \mathrm{FFT}(\widehat{u}_4, backward)$
$\widehat{\alpha}_4 \leftarrow$ COMPUTE_TFN(u_4)
for $j = 1, \ldots, N$ **do**
$\widehat{u}_1[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + \frac{h}{6}\widehat{\alpha}_1[j] + \frac{h}{3}\widehat{\alpha}_2[j] + \frac{h}{3}\widehat{\alpha}_3[j]) + \frac{h}{6}\widehat{\alpha}_4[j]$
end for
$u \leftarrow \mathrm{FFT}(\widehat{u}_1, backward)$ {Array u contains $[A_k(z_{k+1}, t_j)]_{j=1,\ldots,N}$ the sampling of the signal amplitude at step z_k}
end for

FUNCTION $\widehat{g}$ = COMPUTE_TFN(f)
{Compute the Fourier transform of $g : t \mapsto \mathcal{N}(f)(z,t)$ for a given z}
Input: Array f contains the time sampling of function f for the given z
Array $\widehat{h}_R$ contains the sampled Fourier transform of the Raman response function
Array $[\nu_j]_{j=1,\ldots,J}$ contains the frequency sampling points
Output: Array $\widehat{g}$ contains the sampled Fourier transform of g
for $j = 1, \ldots, J$ **do**
$op_1[j] \leftarrow |f[j]|^2$
end for
$\widehat{op}_1 \leftarrow \mathrm{FFT}(op_1, forward)$
for $j = 1, \ldots, J$ **do**
$\widehat{op}_2[j] \leftarrow \widehat{op}_1[j] \times \widehat{h}_R[j]$
end for
$op_2 \leftarrow \mathrm{FFT}(\widehat{op}_2, backward)$ {Array op_2 contains the convolution product $h_R \star_t |f(t)|^2$}
for $j = 1, \ldots, J$ **do**
$op_3[j] \leftarrow f[j] \times \big((1 - f_R)op_1[j] + f_R op_2[j]\big)$
end for
$\widehat{op}_3 \leftarrow \mathrm{FFT}(op_3, forward)$
for $j = 1, \ldots, J$ **do**
$\widehat{g}[j] \leftarrow \mathrm{i}\gamma(1 + \frac{\nu[j]}{\nu_0})h_t \times \widehat{op}_3[j]$
end for

In this algorithm, the computational cost mainly lies in the computation of the Fourier transforms. Over one spatial step, the number of Fourier transforms to be computed is 16. The C++ program we have developed to solve the GNLSE (2) by the RK4-IP method according to the above algorithm uses the FFTW library for computing the Fourier transforms [23]. FFTW, for "Fastest Fourier Transform in the West", is a software library for computing discrete Fourier transforms (DFTs) developed by Matteo Frigo and Steven G. Johnson at the Massachusetts Institute of Technology. It supports a variety of algorithms and can choose the one it estimates or measures to be preferable in the particular circumstances. FFTW is known as the fastest free software implementation of the Fast Fourier transform (FFT) algorithm at present time. It can compute transforms of real-valued and complex-valued arrays of arbitrary size n with a complexity in $\mathcal{O}(n\log(n))$.

5.2. Step-size control

For step-size control, a tolerance "tol" is given as bound on the local error estimate. A step-size control strategy [7] consists in rejecting the current step-size if it gives an estimated local error higher than the specified tolerance and 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 such 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 such that the local error is lower than $\mathrm{tol} \times h_k$. It is clear that for sufficiently small tolerance value EPUS criterion selects a smaller step-size than EPS criterion. 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. In both case, the new step-size has to be estimated using the available information on the previous step computations. Here, we consider the ERK4(3) method defined by (84) and we assume that the leading term in the asymptotic expansion (87) of the local error dominates. However a similar development could be done with the other methods presented in section 4. From (87) and (88) there exists $C \in \mathbb{R}_+^*$ 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.* $C\ h_{\mathrm{opt}}^4 = \mathrm{tol}$. By eliminating the constant C from these 2 relations we obtain

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

For robustness the step-size control has to be designed in order to respond as smoothly as possible with real or apparent abrupt changes in behaviour. This means that the step-size should not vary from one step to the other 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. As well, in order to avoid situations where the specified tolerance is ever exceeded resulting in rejecting too many steps, a safety factor is sometimes introduced. 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 $0.9h_{\mathrm{opt}}$ for instance is used instead. Here our approach of evaluating the local error for the 3rd order RK method but propagating the solution computed from the 4th order RK method is known to overestimate the actual local error. As a consequence no safety factor is needed.

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

$$h_{\mathrm{new}} = \max\left(0.5\ ,\ \min\left(2.0\ ,\ \sqrt[4]{\frac{\mathrm{tol}}{\mathrm{err}}}\right)\right) h_k \tag{91}$$

where "tol" denotes the tolerance value specified by the user as a bound on the local error and "err" denotes the estimation of the local error for the current step as given by (88) for the ERK4(3) method. The 2 constants with values 0.5 and 2.0 are somewhat arbitrary and have to be regarded as design parameters.

In figure 1 we have figure the behavior of the step-size control formula (91) by drawing the variation of the ratio $h_{\text{new}}/h_{\text{old}}$ against the variation of the ratio tol/est.

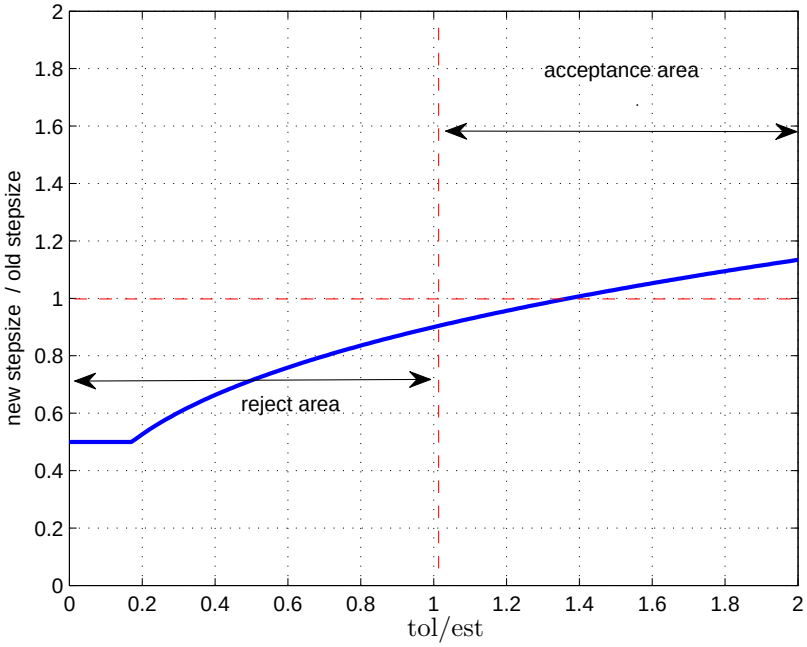

Figure 1. Illustration of the behaviour of the step-size control formula.

Following a similar reasoning, we use the following step-size control formula for the Conservation Quantity Error method and for the step doubling method

$$h_{\text{new}} = \max\left(0.5\ , 0.9\ \min\left(2.0\ , \sqrt[5]{\frac{\text{tol}}{\text{err}}}\right)\right) h_k \tag{92}$$

where the estimation of the local error for the current step is given by (52) for the CQE method and by (43) for the SD method. In (92), the constant with value 0.9 is to be regarded as a safety factor.

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

We present in this section the algorithm for the fourth order Embedded Runge-Kutta in the Interaction Picture (ERK4(3)-IP) method. Taking into account the step-size control strategy based on the ERK4(3) method presented in section 5.2, we transform the computational sequence (89) in the following way:

$$
\begin{aligned}
\widehat{u}_k^{\mathrm{ip}}(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}} \times \widehat{u}_k(\nu) \\
\widehat{\alpha}_1(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}} \times \widehat{\alpha}'_{5,k} \\
\widehat{\alpha}_2(\nu) &= \widehat{\mathcal{N}}\big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h_k}{2}\widehat{\alpha}_1(\nu)\big) \\
\widehat{\alpha}_3(\nu) &= \widehat{\mathcal{N}}\big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h_k}{2}\widehat{\alpha}_2(\nu)\big) \\
\widehat{\alpha}'_4(\nu) &= \widehat{\mathcal{N}}\big(\mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}} \times \widehat{u}_k^{\mathrm{ip}}(\nu) + h_k\widehat{\alpha}_3(\nu)\big) \\
\widehat{r}(\nu) &= \mathrm{e}^{\widehat{d}_\nu \frac{h_k}{2}} \times \big(\widehat{u}_k^{\mathrm{ip}}(\nu) + \tfrac{h_k}{6}\left(\widehat{\alpha}_1(\nu) + 2\widehat{\alpha}_2(\nu) + 2\widehat{\alpha}_3(\nu)\right)\big) \\
\widehat{u}_{k+1}^{[4]}(\nu) &= \widehat{r}(\nu) + \tfrac{h_k}{6}\widehat{\alpha}'_4(\nu) \\
\widehat{\alpha}'_{5,k+1}(\nu) &= \widehat{\mathcal{N}}\big(\widehat{u}_{k+1}^{[4]}(\nu)\big) \\
\widehat{u}_{k+1}^{[3]}(\nu) &= \widehat{r}(\nu) + \tfrac{h_k}{30}(2\widehat{\alpha}'_4 + 3\widehat{\alpha}'_5) \\
L_{k+1}^{[3]} &= \left(h_t \sum_{j=0}^{J-1} \left|\widehat{u}_{k+1}^{[4]}(t_j) - \widehat{u}_{k+1}^{[3]}(t_j)\right|^2\right)^{\frac{1}{2}} \\
h_{\mathrm{opt}} &= \max\left(0.5\ ,\ \min\left(2.0\ ,\ \sqrt[4]{\frac{\mathrm{tol}}{\mathrm{L}_{k+1}^{[3]}}}\right)\right) h_k
\end{aligned}
\tag{93}
$$

If $L_{k+1}^{[3]} \leqslant \mathrm{tol}$ where tol denotes the prescribed tolerance for the local error then the current computational step is accepted and we record the mapping $u_{k+1} : t \mapsto \mathcal{F}^{-1}\big(\nu \mapsto \widehat{u}_{k+1}(\nu)\big)(t)$ as the approximate solution at grid point z_{k+1}. For computations from the new grid point $z_{k+1} = z_k + h_k$, the step-size is set to $h_{k+1} = \min(h_{\mathrm{opt}}, L - z_{k+1})$ where L denotes the fiber length. If the local error does not match the tolerance then computations from grid point z_k given by (93) are repeated with the smaller step-size $h_k = h_{\mathrm{opt}}$.

This computational procedure gives rise to the following algorithm derived from the basic RK4-IP algorithm presented in section 5.1:

ERK4(3)-IP algorithm

Input: Array u contains the input signal amplitude sampled in time
Array $[\nu_j]_{j=1,\ldots,J}$ contains the frequency sampling points
Array $\widehat{h}_R$ contains the Fourier transform of the Raman response function
tol the tolerance value for the local error
{Initialisation}
for $j = 1, \ldots, J$ **do**
$\widehat{d}[j] \leftarrow -\frac{1}{2}\alpha + \mathrm{i}\sum_{n=2}^{n_{\max}} \frac{\beta_n}{n!}(2\pi\nu_j)^n$
end for
$\widehat{u} \leftarrow \mathrm{FFT}(u, forward)$
$\widehat{Nu} \leftarrow$ COMPUTE_TFN(u)
$z_k = 0$, $h = h_{\mathrm{initial}}$
{Loop over the propagation sub-interval}
while $z_k < L$ **do**
for $j = 1, \ldots, J$ **do**
$tfexpd[j] \leftarrow \exp(\frac{h}{2}\widehat{d}[j])$

$\widehat{u}_{ip}[j] \leftarrow tfexpd[j] \times \widehat{u}[j]$
end for
for $j = 1, \ldots, J$ **do**
$\widehat{\alpha}_1[j] \leftarrow tfexpd[j] \times \widehat{Nu}[j]$
$\widehat{u}_2[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_1[j]$
end for
$u_2 \leftarrow \mathrm{FFT}(\widehat{u}_2, backward)$
$\widehat{\alpha}_2 \leftarrow \mathrm{COMPUTE\_TFN}(u_2)$
for $j = 1, \ldots, J$ **do**
$\widehat{u}_3[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_2[j]$
end for
$u_3 \leftarrow \mathrm{FFT}(\widehat{u}_3, backward)$
$\widehat{\alpha}_3 \leftarrow \mathrm{COMPUTE\_TFN}(u_3)$
for $j = 1, \ldots, J$ **do**
$\widehat{u}_4[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + h\widehat{\alpha}_3[j])$
end for
$u_4 \leftarrow \mathrm{FFT}(\widehat{u}_4, backward)$
$\widehat{\alpha}_4 \leftarrow \mathrm{COMPUTE\_TFN}(u_4)$
for $j = 1, \ldots, J$ **do**
$\widehat{r}[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + \frac{h}{6}\widehat{\alpha}_1[j] + \frac{h}{3}\widehat{\alpha}_2[j] + \frac{h}{3}\widehat{\alpha}_3[j])$
$\widehat{u}_1[j] \leftarrow \widehat{r}[j] + \frac{h}{6}\widehat{\alpha}_4[j]$ {RK4 solution}
end for
$u_1 \leftarrow \mathrm{FFT}(\widehat{u}_1, backward)$
$\widehat{\alpha}_5 \leftarrow \mathrm{COMPUTE\_TFN}(u_1)$
for $j = 1, \ldots, J$ **do**
$\widehat{v}_1[j] \leftarrow \widehat{r}[j] + \frac{h}{30}(2\widehat{\alpha}_4[j] + 3\widehat{\alpha}_5[j])$ {RK3 solution}
end for
{Step-size control}
$\mathrm{err} \leftarrow 0$
for $j = 1, \ldots, J-1$ **do**
$\mathrm{err} \leftarrow \mathrm{err} + |\widehat{u}_1[j] - \widehat{v}_1[j]|^2$
end for
$\mathrm{err} \leftarrow \sqrt{h_t\,\mathrm{err}}$
$h_{\mathrm{opt}} = \max\left(0.5\ ,\ \min\left(2.0\ ,\ \sqrt[4]{\frac{\mathrm{tol}}{\mathrm{err}}}\right)\right) h$ {Optimal step-size for the given prescribed tolerance}
if $\mathrm{err} \leqslant \mathrm{tol}$ **then**
{the current local error matches the tolerance}
$z_k = z_k + h$ {New grid point is confirmed}
$h = \min(h_{\mathrm{opt}}, L - z_k)$ {New step-size value}
$u \leftarrow u_1$ {Array u contains the time sampled values $[A_k(z_k, t_j)]_{j=1,\ldots,J}$ of the signal amplitude at grid point z_k}
$\widehat{u} \leftarrow \widehat{u}_1$
$\widehat{Nu} \leftarrow \widehat{\alpha}_5$
else
{The current local error does not match the tolerance}
$h = h_{\mathrm{opt}}$ {New computation from z_k with smaller step-size h_{opt} is necessary}
end if
end while

A comparison of this algorithm to the basic RK4-IP algorithm presented in section 5.1 shows that it has a computational cost very similar. The number of evaluations of the non-linear operator $\mathcal{N}$ is 4 in both case and in both case we have 4 evaluations of the $\exp(\frac{h_k}{2}\mathcal{D})$ operator. The extra cost is limited to 2 additions and 3 multiplications and the need to keep in memory 2 intermediates results (not including the computation of the new optimal step-size). Of course, this is only true when the current step-size is not rejected since otherwise all the computations for the current step are lost when we restart from the previous grid point with a smaller step-size. The over-cost is limited to the computation of the OPN and to the computation of the new optimal step-size.

5.4. Algorithm for the RK4-IP method with step-size control by the CQE method

We now present a variant algorithm for the RK4-IP method where the step-size control is done according to the modified CQE method described in section 4.3. This algorithm is derived from the basic RK4-IP algorithm presented in section 5.1 and in particular the function COMPUTE_TFN(f) is identical in both cases.

RK4-IP algorithm with MCQE adaptive step-size strategy

Input: Array u contains the sampling of the signal amplitude at the fibre entrance
Array $[\nu_j]_{j=1,\ldots,J}$ contains the frequency sampling points
Array $\widehat{h}_R$ contains the Fourier transform of the Raman response function

Output: Array u contains the sampling of the signal amplitude at the fibre end

{Initialisation}
for $j = 1, \ldots, J$ **do**
$\widehat{d}[j] \leftarrow -\frac{1}{2}\alpha + \mathrm{i}\sum_{n=2}^{n_{\max}} \frac{\beta_n}{n!}(2\pi\nu_j)^n$
end for
$\widehat{u} \leftarrow \mathrm{FFT}(u, forward)$
$z_k = 0$, $h = h_{\mathrm{initial}}$
{Loop over the propagation sub-interval}
while $z_k < L$ **do**
$\mathrm{OPN}_k \leftarrow 0$ {Optical Photon Number at the current grid point}
for $j = 1, \ldots, J$ **do**
$\mathrm{OPN}_k \leftarrow \mathrm{OPN}_k + |\widehat{u}[j]|^2/(2\pi\nu[j])$
$tfexpd[j] \leftarrow \exp(\frac{h}{2}\widehat{d}[j])$
$\widehat{u}_{ip}[j] \leftarrow tfexpd[j] \times \widehat{u}[j]$
end for
$\widehat{\alpha}_1 \leftarrow \mathrm{COMPUTE\_TFN}(u)$
for $j = 1, \ldots, J$ **do**
$\widehat{\alpha}_1[j] \leftarrow tfexpd[j] \times \widehat{\alpha}_1[j]$
$\widehat{u}_2[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_1[j]$
end for
$u_2 \leftarrow \mathrm{FFT}(\widehat{u}_2, backward)$
$\widehat{\alpha}_2 \leftarrow \mathrm{COMPUTE\_TFN}(u_2)$
for $j = 1, \ldots, J$ **do**
$\widehat{u}_3[j] \leftarrow \widehat{u}_{ip}[j] + \frac{h}{2}\widehat{\alpha}_2[j]$
end for
$u_3 \leftarrow \mathrm{FFT}(\widehat{u}_3, backward)$

$\widehat{\alpha}_3 \leftarrow \text{COMPUTE\_TFN}(u_3)$
for $j = 1, \ldots, J$ **do**
$\widehat{u}_4[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + h\widehat{\alpha}_3[j])$
end for
$u_4 \leftarrow \text{FFT}(\widehat{u}_4, backward)$
$\widehat{\alpha}_4 \leftarrow \text{COMPUTE\_TFN}(u_4)$
for $j = 1, \ldots, J$ **do**
$\widehat{u}_{\text{tmp}}[j] \leftarrow tfexpd[j] \times (\widehat{u}_{ip}[j] + \frac{h}{6}\widehat{\alpha}_1[j] + \frac{h}{3}\widehat{\alpha}_2[j] + \frac{h}{3}\widehat{\alpha}_3[j]) + \frac{h}{6}\widehat{\alpha}_4[j]$
end for
{Step-size control}
$\text{OPN}_{k+1} \leftarrow 0$ {Optical Photon Number at new grid point}
for $j = 1, \ldots, J$ **do**
$\text{OPN}_{k+1} \leftarrow \text{OPN}_{k+1} + |\widehat{u}_{\text{tmp}}[j]|^2/(2\pi\nu[j])$
end for
$\text{err} \leftarrow |\mathrm{e}^{-\alpha h}\text{OPN}_k - \text{OPN}_{k+1}|$
$h_{\text{opt}} = \max\left(0.5\ ,\ \min\left(2.0\ , 0.9\sqrt[5]{\frac{\text{tol}}{\text{err}}}\right)\right) h$ {Optimal step-size for the given prescribed tolerance}
if $\text{err} \leqslant \text{tol}$ **then**
{the current local error matches the tolerance}
$z_k = z_k + h$ {New grid point is confirmed}
$h = \min(h_{\text{opt}}, L - z_k)$ {New step-size value}
$\widehat{u} \leftarrow \widehat{u}_{\text{tmp}}$
$u \leftarrow \text{FFT}(\widehat{u}, backward)$ {Array u contains the values $[A_k(z_{k+1}, t_j)]_{j=1,\ldots,J}$ the sampling of the signal amplitude at step z_k}
else
{The current local error does not match the tolerance}
$h = h_{\text{opt}}$ {New computation from z_k with smaller step-size h_{opt} is necessary}
end if
end while

A comparison of this algorithm to the basic RK4-IP algorithm presented in section 5.1 and to the ERK4(3)-IP algorithm shows that it has a computational cost very similar. Indeed, the MCQE adaptive step-size strategy do not modify the structure of the basic RK4-IP algorithm but only add the computation of the OPN at the entrance and at the end of the current step section. Here again, the number of evaluations of the non-linear operator $\mathcal{N}$ is 4 in both case and in both case we have 4 evaluations of the $\exp(\frac{h_k}{2}\mathcal{D})$ operator.

5.5. Numerical experiments

5.5.1. Soliton solution to the NLSE in optics We first consider the case of the non-linear Schrödinger equation (NLSE) in optics, a simplified version of the GNLSE (2) where $\alpha = 0$, $f_R = 0$, $n_{\max} = 2$. The linear operator is $\mathcal{D} : A \mapsto \mathrm{i}\beta_2\partial_{tt}A$ and the non-linear operator is $\mathcal{N} : A \mapsto \mathrm{i}\gamma A(z,t)\left|A(z,t)\right|^2$. 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

$$\forall t \in \mathbb{R} \quad a_0(t) = \frac{N}{\sqrt{\gamma L_D}} \frac{1}{\text{ch}(t/T_0)} \tag{94}$$

where $N = 1$ 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 reads $\forall z \in [0, L]$

$$\forall t \in \mathbb{R} \quad A(z,t) = \frac{N}{\sqrt{\gamma L_D}} \frac{\mathrm{e}^{\mathrm{i}z/2L_D}}{\mathrm{ch}(t/T_0)}. \tag{95}$$

Furthermore, for $N \in \mathbb{N}, N \geqslant 2$, relation (95) gives the solution to the NLSE in position z multiple of $\frac{\pi}{2} L_D$.

Fundamental soliton ($N = 1$) doesn't provide a well suited example for exploring the features of the ERK4(3) method and for comparison purposes since its shape doesn't change on propagation. We therefore consider in the following a 3rd order soliton ($N = 3$). In Fig. 2 we show for the 3rd order soliton the adjustment of the step-size when using the ERK4(3) method for evaluating the local error with a tolerance set to $\mathrm{tol} = 10^{-6}$ and an initial step-size of $h = 1\,\mathrm{m}$. The other physical parameters of the numerical experiment are $L = 637.21\,\mathrm{m}$, $\gamma = 4.3\,\mathrm{W}^{-1}\,\mathrm{km}^{-1}$, $\beta_2 = -19.83\,\mathrm{ps}^2\,\mathrm{km}^{-1}$, $T_0 = 2.8365\,\mathrm{ps}$. The number of discretisation steps along the fiber is found to be 605 and the computation time is 69 s on a Intel Core 2 Quad Q6600. At the fiber end ($z = L$), the relative global error measured with the quadratic norm is $1.12\,10^{-4}$ whereas the maximum relative error is $1.89\,10^{-4}$.

The same accuracy with a constant step-size computation would have required a step size of 0.01 m for a total number of step of 63722 and a computation CPU time of 5490 s.

For comparison, when using an adaptive step-size strategy based on the SD approach with the same values of tolerance and initial step-size, we obtain that the number of discretisation steps along the fiber is 396 (or 792 if we consider that it is the accurate solution computed over the fine grid of step-size $h_k/2$ that is propagated) and the computation time is 148 s. At the fiber end ($z = L$), the relative quadratic error is $8.83\,10^{-6}$ whereas the maximum relative error is $1.48\,10^{-5}$. The evolution of the step-size along the fiber is depicted in Fig. 2 for a comparison with the ERK4(3) method. Now if we impose to find with the ERK4(3) method a quadratic error at the fiber end of approx. $8.83\,10^{-6}$ (for a comparison with the accuracy obtained with the SD method) we obtain the result by setting a tolerance of 10^{-7} and the number of step is 1209 whereas the CPU time is 128 s.

This simple example illustrates the fact that the ERK4(3) method overestimates the local error as mentioned before. The consequence is that the size of the steps are a little smaller than the one obtained with the SD method on the coarse grid and therefore that a larger number of steps is required. However, since in the ERK4(3) method the local error for each step is computed faster than in the SD method the total CPU time of the computation is much lower. The global error at the fiber end ($z = L$) is smaller when the IP method is used in conjunction with the SD method since it is not the solution computed with the RK4 method that is propagated but the more accurate one computed with the half step-size in the SD method [11].

Since $\alpha = 0$ in the simplified situation considered here, the two step-size control approaches based on the CQE method and on the MCQE method coincide. With the same values of tolerance and initial step-size as before ($\mathrm{tol} = 10^{-6}$ and $h = 1$ m), we obtain that the number of discretisation steps along the fiber is 226 and the computation time is 34 s. At the fiber end ($z = L$), the relative quadratic error is $5.19\,10^{-3}$ whereas the maximum relative error is $9.17\,10^{-3}$. The evolution of the step-size along the fiber is depicted in Fig. 3 for a comparison with the ERK4(3) method.

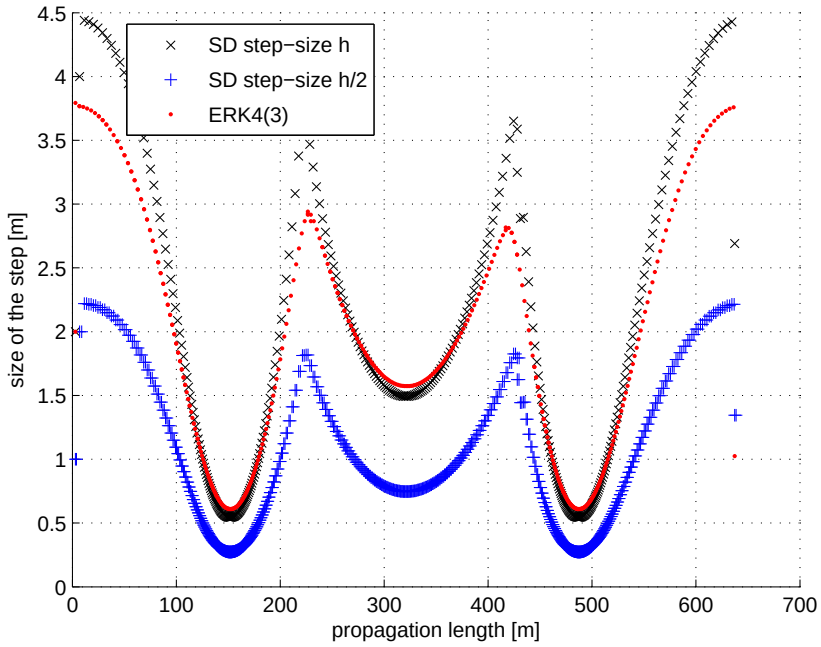

Figure 2. Evolution of the step-size along the fiber length for the ERK4(3) and the SD methods (considered over the coarse and fine grids) when solving the NLSE for a 3rd order soliton.

We can observe that the size of the steps obtained by the CQE method are much larger than the ones obtained by the ERK4(3) method. Moreover we can see that the general shape of the curve obtained with the ERK4(3) method is smooth whereas the one corresponding to the CQE method is rougher. It's means that there is more estimated step-sizes rejected with the CQE method than with the ERK4(3) method. This observation has to put facing the globally good agreement in the general shape of the curve for the ERK4(3) and the SD method shown in Fig. 2. It appears that the CQE method selects for a given tolerance larger step-sizes than the 2 other methods. The CQE method is therefore fast, but it causes a larger number of rejected step-sizes and a rather bad accuracy of the computed solution at the fiber end for a given tolerance compared to the SD and ERK4(3) method. When compared to the ERK4(3) method, we can see that the CQE method computes a solution 50 times less accurate for a CPU time just twice smaller.

To be comprehensive, when using the same parameters as before (a tolerance set to 10^{-6} and an initial step-size set to $h = 1\,\mathrm{m}$) with a step-size control strategy based on the ERK4(2) method defined in (79) rather than on the ERK4(3) method, we obtain at the fiber end a quadratic relative error of $2.96\,10^{-8}$ and a maximum relative error of $2.77\,10^{-8}$. The number of discretisation steps is 4035 and the computation time is 367 s. In this case, the local error is estimated from the solution computed with a 2nd order RK scheme and therefore the local error is largely overestimated (since it is the more accurate solution computed with a 4th order RK scheme that is propagated) resulting in a underestimation of the optimal step-size.

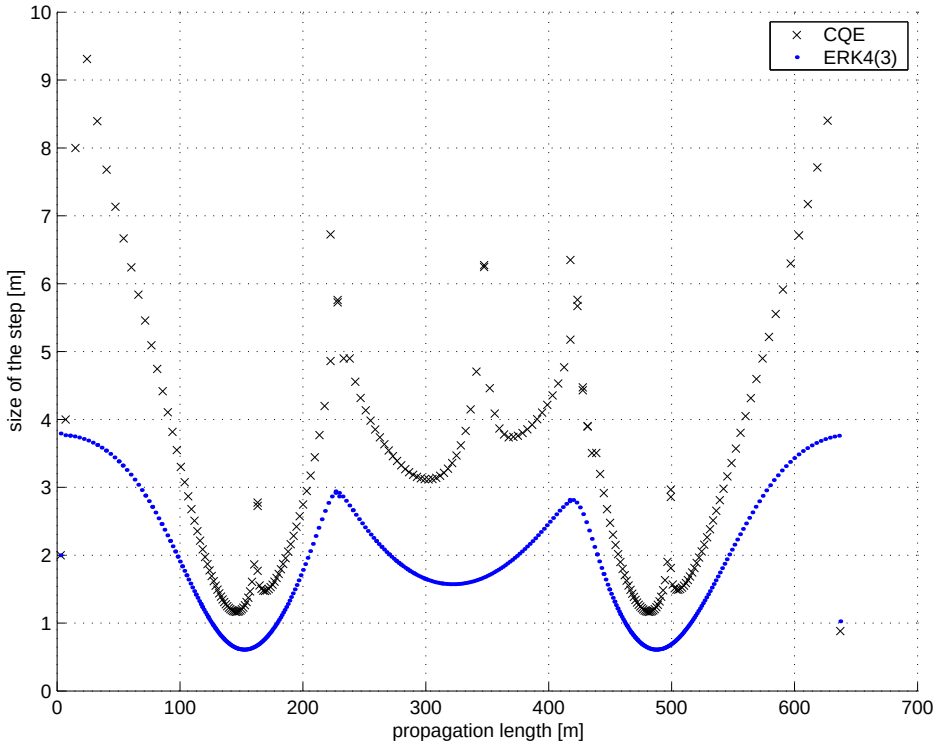

Figure 3. Evolution of the step-size along the fiber length for the ERK4(3) and the CQE methods when solving the NLSE for a 3rd order soliton.

We have summarized in table 1 the features of the various step-size control approaches given above. In table 2 we give the features of the step-size control approaches for solving the NLSE with a tolerance of 10^{-9} and an initial step-size of 0.1 m. The same comments as for a tolerance of 10^{-6} can be done regarding the comparison between the ERK4(3) and CQE methods. For a similar CPU time, the accuracy of the computed solution at the fiber end is 100 time better with the ERK4(3) method than with the CQE method. Surprisingly, it is the SD method that provides the better result. The reason could be that the low value chosen for the tolerance implies a large number of steps and since the accurate local error estimate provided by the SD method enables to design a quite optimal step-size, the total number of steps is smaller (as observed in the numerical experiment). Globally even if each step

Method	number of steps	CPU time (s.)	quadratic error	maximal error
ERK4(3)	605	69	$1.12\,10^{-4}$	$1.89\,10^{-4}$
SD	792	148	$8.83\,10^{-6}$	$1.48\,10^{-5}$
CQE	226	34	$5.19\,10^{-3}$	$9.17\,10^{-3}$
ERK4(2)	4035	367	$2.96\,10^{-8}$	$2.77\,10^{-8}$

Table 1. Comparison of the efficiency of the various step-size control approaches for solving the NLSE with a tolerance of 10^{-6} and an initial step-size of 1 m.

requires more time, the whole execution time is good enough and the result accurate.

Method	number of steps	CPU time (s.)	quadratic error	maximal error
ERK4(3)	5052	554	$4.49\,10^{-8}$	$4.35\,10^{-8}$
SD	3188	529	$1.46\,10^{-8}$	$1.79\,10^{-8}$
CQE	6367	662	$5.73\,10^{-6}$	$5.04\,10^{-6}$

Table 2. Comparison of the efficiency of the various step-size control approaches for solving the NLSE with a tolerance of 10^{-9} and an initial step-size of 0.1 m.

5.5.2. Solving the GNLSE in optics We now consider the case of the GNLSE (2) with the following set of physical parameters : $\omega_0 = 1770\,\text{Thz}$, $\gamma = 4.3\,\text{W}^{-1}\text{km}^{-1}$, $\beta_2 = 19.83\,\text{ps}^2\text{km}^{-1}$, $\beta_3 = 0.031\,\text{ps}^3\text{km}^{-1}$ and $\beta_n = 0$ for $n \geqslant 4$, $\alpha = 0.046\,\text{km}^{-1}$, $L = 96,77\,\text{m}$, $f_R = 0.245$. An expression for the Raman time response function for silica core fiber is given in [24]. The Gaussian pulse at the fiber entrance ($z = 0$) is expressed as

$$\forall t \in \mathbb{R} \quad a_0(t) = \sqrt{P_0}\,\text{e}^{-\frac{1}{2}(t/T_0)^2} \tag{96}$$

where $T_0 = 2.8365\,\text{ps}$ is the pulse half-width and $P_0 = 100\,\text{W}$ is the pulse peak power.

In Fig. 4 we show the adjustment of the step-size when using the ERK4(3) method for evaluating the local error with a tolerance set to $\text{tol} = 10^{-6}$ and an initial step size of $h = 0.1\text{m}$. The number of discretisation steps along the fiber is found to be 279 and the computation time is 50 s. The quadratic norm of the solution at the fiber end ($z = L$) is 23.018853566539611 whereas the norm of the maximum is 5.1082812862836695.

When using the SD method for determining the step-size in the IP method in the same circumstances we find that the number of discretisation steps along the fiber is 232 and the computation time is 124 s. The quadratic norm of the solution at the fiber end ($z = L$) is 23.018876949765658 whereas the norm of the maximum is 5.1082806571962491. Here again the same comments as for the soliton case can be done when comparing the 2 adaptive step-size approaches.

In the present situation since the fiber suffers losses, the 2 methods CQE and MCQE take into account the losses in a slightly different way and therefore give very slightly different results for the control of the step-size. With a tolerance set to $\text{tol} = 10^{-6}$ and an initial step size of $h = 0.1\,\text{m}$ as before, we obtain 123 steps along the fiber and a CPU time of 28.54 s for the CQE method and 123 steps along the fiber and a CPU time of 30.79 s for the MCQE method. The quadratic norm of the solution at the fiber end ($z = L$) is 23.0181610088442560 whereas the norm of the maximum is 5.1083035516257808 for the CQE method and it is respectively 23.018161878786020 and 5.1083036530739773 for the MCQE method. The evolution of the step-size along the fiber for the CQE and MCQE methods is depicted in Fig. 4 for a comparison with the ERK4(3) and SD methods. Here again one can observe that the size of the steps obtained by the CQE and MCQE methods are much larger than the ones obtained by the ERK4(3) and SD methods.

We have summarized in table 3 the features of the various step-size control approaches when solving the GNLSE with a tolerance of 10^{-6} and an initial step-size of 0.1 m.

When the tolerance is set to $\text{tol} = 10^{-9}$ with an initial step size of $h = 0.1\,\text{m}$, the number of steps with the ERK4(3) method is 1545 and the computational time is 221 s

Method	steps numb.	CPU time (s.)	quadratic norm	norm of the max.
ERK4(3)	279	50	23.018853566539611	5.1082812862836695
SD	232	124	23.018876949765658	5.1082806571962491
CQE	123	28.54	23.018161008844256	5.1083035516257808
MCQE	123	30.79	23.018161878786020	5.1083036530739773

Table 3. Comparison of the efficiency of the various step-size control approaches for solving the GNLSE with a tolerance of 10^{-6} and an initial step-size of 0.1 m.

whereas 906 steps are required by the SD method for a computational time of 645 s. The quadratic norm of the solution at the fiber end ($z = L$) is 23.018880606441541 whereas the norm of the maximum is 5.1082806056123137 for the ERK4(3) method and it is respectively 23.018880620838782 and 5.1082806054745253 for the SD method. We obtain 457 steps along the fiber and a CPU time of 76 s for the CQE method and 463 steps along the fiber and a CPU time of 70 s for the MCQE method. The quadratic norm of the solution at the fiber end ($z = L$) is 23.018877436289240 whereas the norm of the maximum is 5.1082219724449098 for the CQE method and it is respectively 23.018877663305577 and 5.1082220123430686 for the MCQE method.

The evolution of the step-size along the fiber length is very similar to the one presented in Fig. 3.

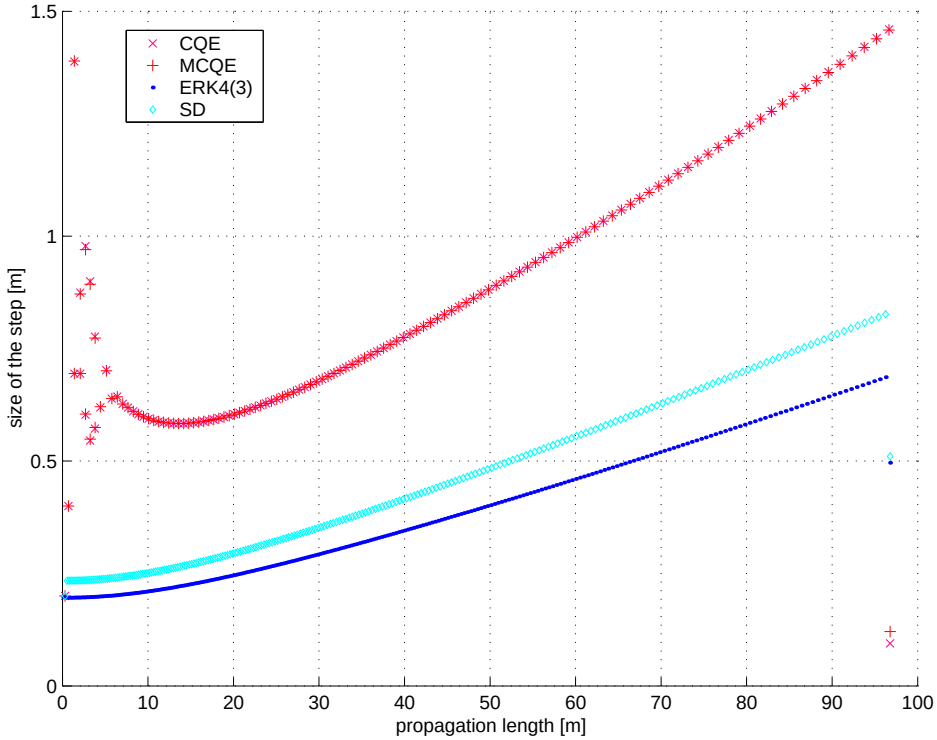

Figure 4. Evolution of the step-size along the fiber length for the ERK4(3) and the SD methods when solving the GNLSE.

6. Conclusion

In this document we have studied the possibility to use an ERK method to solve the nonlinear problem in the IP or S3F methods applied to the GNLSE and to deliver an estimation of the local error for step-size control purposes. We have shown that it doesn't exist a 4 stages (or less) RK3 method embedded in the standard RK4 method. Only a RK2 method is obtained. However we have presented a 5 stages RK3 method in which the standard RK4 method is embedded. Since in practise the solution computed with the RK4 method is more accurate, it is the one that is propagated. This approach usually overestimates the actual local error which is safe even though not optimal for step-size control purposes. This drawback is offset by the fact the ERK method delivers an estimation of the local error at no extra cost compared to the standard RK4 method. We have also compare this approach to other standard methods for estimating the local error such as the step doubling method and the CQE method both theoretically and on numerical experiments. In the 2 situations we have investigated numerically, we have found that the ERK4(3) method provides step-sizes in a very similar way to the SD method. However, as the ERK4(3) method overestimates the local error the step-sizes are a little smaller than the ones provided by the SD method. The ERK4(3) method is however much faster than the SD method. When compared to the ERK4(3) method, we have observed that the CQE and MCQE methods give larger step-sizes and less accurate solutions for a comparable computation time. Moreover we have seen that the control step-size strategy based on the CQE and MCQE methods provides a rougher curve for the selected step-sizes than the one obtained with the ERK4(3) method. A recommendation on the basis of this study would be to prefer a control step-size strategy based on the ERK4(3) method rather than on the CQE method.

On the contrary to the Symmetric Split-Step method the IP method uses a splitting based on a change of unknown rather than to the second order accurate Strang formula and therefore is exact. It follows that it could be interesting to look for higher order ERK methods to be used in conjunction with the IP method. One motivation for such a study is that so as to attain a certain accuracy of the results less computational steps are required with high order RK schemes and therefore they are likely to reduce the accumulation of round-off errors. There exists in the literature a lot of high order ERK schemes [7]. However each of these schemes has been constructed in order to satisfy one given criterion and none of them preserve the advantageous position of the internal quadrature nodes of the RK4 formula liable for the efficiency of the RK4-IP method. Therefore a next stage would be the construction of a ERK5(4) method well suited to be used in conjunction with the IP method and that preserves the ease of implementation and the advantageous position of the internal quadrature nodes of the RK4 formula so far as one can.

Acknowledgments

This work has been undertaken under the framework of the Green-Laser project and was partially supported by Conseil Régional de Bretagne, France. The authors would like to thank F. Mahé and R. Texier-Picard from the Institute of Mathematics in Rennes, France (IRMAR CNRS UMR 6625) for the helpfull advices on Runge-Kutta methods and for their contributions to the Green-Laser project. The authors also would like to thank Pr. O. Boffard and S. Le Maguer from IRISA (CNRS UMR

6074) Project team CORDIAL for the technical support in computer science they have kindly provided.

References

[1] G. Agrawal. *Nonlinear fiber optics*. Academic Press, 3rd edition, 2001.
[2] O.V. Sinkin, R. Holzlöhner, J. Zweck, and C.R. Menyuk. Optimization of the Split-Step Fourier method in modeling optical-fiber communications systems. *J. Lightwave Technol.*, 21(1):61, 2003.
[3] T.N. Nguyen, M. Gay, L. Bramerie, T. Chartier, and J.C. Simon. Noise reduction in 2R-regeneration technique utilizing self-phase modulation and filtering. *Opt. Exp.*, 14(5):1737–1747, 2006.
[4] J.A.C. Weideman and B.M. Herbst. Split-step methods for the solution of the nonlinear Schrodinger equation. *SIAM J. Numer. Anal.*, 23(3):485–507, 1986.
[5] R. I. McLachlan and G. R. W. Quispel. Splitting methods. *Acta Numer.*, 11:341–434, 2002.
[6] J. Hult. A fourth-order Runge–Kutta in the Interaction Picture method for simulating supercontinuum generation in optical fibers. *J. Lightwave Technol.*, 25(12):3770–3775, 2007.
[7] J.C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley and Sons, 2008.
[8] E. Hairer, S. P. Norsett, and G. Wanner. *Solving ordinary differential equations I: nonstiff problems*. Springer-Verlag, 1993.
[9] S. Balac, A. Fernandez, F. Mahé, and R. Texier-Picard. The interaction picture method for solving the generalised nonlinear Schrodinger equation for wave propagation in optical fibres: theoretical and experimental comparison with split-step methods. Technical report, CNRS UMR 6082 FOTON, 2012.
[10] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5(3):506–517, 1968.
[11] L. Shampine. Local error estimation by doubling. *Computing*, 34:179–190, 1985.
[12] A. Heidt. Efficient adaptive step size method for the simulation of supercontinuum generation in optical fibers. *J. Lightwave Technol.*, 27(18):3984–3991, 2009.
[13] A. Fernandez, S. Balac, A. Mugnier, F. Mahé, R. Texier-Picard, T. Chartier, and D. Pureur. Numerical simulation of incoherent optical wave propagation in nonlinear fibers. *Submitted to Eur. Phys. J. - Appl. Phys.*, 2012.
[14] A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Number vol. 44 in Applied Mathematical Sciences. Springer, 1992.
[15] J.M. Sanz-Serna and P. Calvo. *Numerical Hamiltonian problems*. Applied mathematics and mathematical computation. Chapman & Hall, 1994.
[16] J.H.E. Cartwright and O. Piro. The dynamics of Runge-Kutta methods. *Int. J. Bifurcation and Chaos*, 2(3):427–449, 1992.
[17] M. Crouzeix and A. Mignot. *Analyse numérique des équations différentielles*. Masson, Paris, 1984.
[18] J. C. Butcher. On Runge-Kutta processes of high order. *Journal of the Australian Mathematical Society*, 4(02):179–194, 1964.
[19] K.J. Blow and D. Wood. Theoretical description of transient stimulated Raman scattering in optical fibers. *IEEE J. Quantum Electron.*, 25(12):2665–2673, 1989.
[20] E. Fehlberg. Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control. Technical report, National Aeronautics and Space Administration, 1968.
[21] J.R. Dormand and P.J. Prince. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, 6:19–26, 1980.
[22] J. R. Dormand and P. J. Prince. New Runge-Kutta algorithms for numerical simulation in dynamical astronomy. *Celestial Mech.*, 18:223–232, 1978.
[23] M. Frigo and S.G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 2(93):216–231, 2005.
[24] D. Hollenbeck and C. D. Cantrell. Multiple-vibrational-mode model for fiber-optic Raman gain spectrum and response function. *Journal Of The Optical Society Of America B-Optical Physics*, 19(12):2886–2892, 2002.