International Journal of Applied Mathematics

Volume 29 No. 1 2016, 105-126
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v29i1.9

A MOVING MESH METHOD FOR THE NONLINEAR SHRÖDINGER EQUATION

Boureima Sangaré

Department of Mathematics, UFR / ST
Polytechnic University of Bobo Dioulasso
01 BP 1091 Bobo-Dioulasso 01, BURKINA FASO

Abstract: In this paper we apply a moving mesh method to a numerically challenging problem namely, the nonlinear Schrödinger (NLS) equation. This problem arises from applications where it has been used to describe the motion of a vortex filament in incompressible fluids and also to model fractal fibre architecture of aortic heart valve lea ets in the study of the human heart. Our aims are to show the existence and the properties of the numerical solution of a nonlinear shrödinger's equation. This method is used to investigate with highly resolved numerics the solution's behavior for small dispersion parameters. Numerical simulations has shown that this scheme is very efficient to solve this problem.

AMS Subject Classification: 65L12, 65M20, 65N40, 65M50
Key Words: moving mesh, nonlinear Shrödinger's equation (NLS), finite difference methods (FDMs), monitor functions

1. Introduction

We are interested to the numerical scheme that we developed to solve some nonlinear Shrödinger's equation. The problem under consideration is as follows: determine a function $u = u(z, t)$, to be a solution of:

Received: November 24, 2015

- the partial differential equation (PDE)

$$iu_t + u_{zz} + qu|u^2| = 0, \quad -\infty \leq z \leq +\infty, \quad t \geq 0, \tag{1}$$

- verifying the initial condition (IC)

$$u(z,0) = u^0(z), \quad z \in \mathbb{R}, \tag{2}$$

- and artificial conditions (BCs) boundary (z_L, z_R) type homogeneous Dirichlet

$$u(z_L,t) = u(z_R,t) = 0, \quad t \geq 0, \tag{3}$$

where q is a real constant and u^0 a given function with complex values.

This equation arises in a number of physical situations including (see e.g. [13] and the references therein) the propagation of a laser beam in a medium whose index of refraction is sensitive to the wave amplitude, the modulational stability of water waves, helical motion of a very thin vortex filament, the propagation of heat pulses in anharmonic crystals, nonlinear modulation of plasma waves and self-trapping of a light beam in a colour-dispersive system.

When the function u is split into its real parts v and w complex, the form $u(z,t) = v(z,t) + iw(z,t)$, the problem amounts to finding two v and w functions, real-valued, such as

$$v_t + w_{zz} + qw(v^2 + w^2) = 0, \tag{4}$$
$$w_t - v_{zz} - qv(v^2 + w^2) = 0, \tag{5}$$
$$v(z,0) = v^0(z), \tag{6}$$
$$w(z,0) = w^0(z), \tag{7}$$
$$v(z_L,t) = v(z_R,t) = 0, \tag{8}$$
$$w(z_L,t) = w(z_R,t) = 0. \tag{9}$$

We will consider here two special cases: a soliton propagation and interaction of two solitons.

The nonlinear Schrödinger equation (SNL) appears in many phenomena Physical for describing nonlinear waves, [5, 6], such as:

- the propagation of a laser beam in a medium whose refractive index is responsive to the magnitude of the wave,

- the modulational instability aquatic waves,

- the propagation of vibrations in heat nonharmonic crystals,
- and the non-linear modulation of plasma waves.

The equation (NLS) has risen considerable interest in the last two decades [1, 3, 7]. The difficulties in solving such equations lie on the fact that classical numerical schemes often produce instabilities and do not point out the time where the solution blows out, for example Backward Euler method.

Since the computational domain has to be a large spatial interval to avoid boundary effects (we assume here that $u_0(x)$ decays rapidly to zero or that it has compact support) uniform meshes become inefficient.

The small length scale, the localization of the solution, and the need of a relatively large computational domain call for an adaptive method. Moving mesh methods are difficult to apply in the non-singular but highly dispersive one-dimensional NLS. In addition to the capability of concentrating sufficient points about regions of rapid variation of the solution, a satisfactory moving mesh method should be simple, easy to program, and reasonably insensitive to the choice of its adjustable parameters.

Here, we study a stable and cost efficient moving mesh method for (1)-(3). Our building block is the mesh generator proposed by [3, 9, 15] combined with a semi-implicit second order time discretization and a fourth order approximation of the mesh advection term.

The organization of this paper is as follows. The nonlinear Schrödinger equation is presented in Section 2. In Section 3 the numerical method is described in detail and the issues of time and spatial discretizations are discussed as they affect stability and performance. The numerical results are presented in Section 4 and some concluding remarks are given in Section 5.

2. The Nonlinear Schrödinger Equation

2.1. Analytical Considerations

The analytical properties of (1)-(3) are well known and are mentioned here for completeness. The linear Schrödinger equation

$$i\Phi_t + \Phi_{zz} = 0, \tag{10}$$

provides a model for the propagation of dispersive waves. The general solution of (10) is given by

$$\Phi(x,t) = \int_{-\infty}^{+\infty} G(k)\, \exp[i(kx - k^2 t)].$$

The problem has been shown to possess an infinite number of conservation laws the most common of which are the charge P, and the energy R, given by

$$P = \int_{-\infty}^{\infty} |u(x,t)|^2\, dx, \quad R = \int_{-\infty}^{\infty} \left(|u(x,t)|^2 - \frac{q}{2}|u(x,t)|^4 \right) dx, \tag{11}$$

where P is just the square of the L_2 norm and its conservation can be deduced by multiplying both sides of (1)-(3) by $\bar{u}$ and equating the imaginary parts, while the conservation is obtained by taking the real part of the product of $\bar{u}_t$ [10, 15].

2.2. Propagation of Single Soliton: SNL1

In the case of the propagation of single soliton, the initial condition is given by

$$u^0(z) = \sqrt{\frac{2a}{q}}\, \exp[i0,5s(z - z_0)]\, \mathrm{sech}\Big[\sqrt{a}(z - z_0)\Big] \tag{12}$$

obtained from the exact solution given by:

$$u(z,t) = \sqrt{\frac{2a}{q}}\, \exp[i0,5s(z - z_0) - (0,25s^2 - a)t]\, \mathrm{sech}[\sqrt{a}(z - z_0 - st)]. \tag{13}$$

The artificial boundary of the spatial domain were fixed to $z_L = -30$ and $z_R = -70$ and the time interval $[0, 30]$ with a time step $\triangle t = 5$. The module $|u(z,t)|$ represents a wave initially located at point $z = z_0$ and which moves at the speed s in the positive direction of z. Its amplitude $\sqrt{\frac{2a}{q}}$ is determined by the real parameter a. Here we take $q = 1$, $a = 1$, $s = 1$ and $z_0 = 0$ as the problem parameter [11, 13, 14].

2.3. Interaction of Two Solitons: SNL2

We now consider an initial condition (IC) given by

$$u^0(z) = \sqrt{\frac{2a_1}{q}}\, \exp\Big[i0,5s_1(z - z_{01})\Big]\, \mathrm{sech}\Big[\sqrt{a_1}(z - z_{01})\Big]$$
$$+ \sqrt{\frac{2a_2}{q}}\, \exp\Big[i0,5s_2(z - z_{02})\Big] \mathrm{sech}\Big[\sqrt{a_2}(z - z_{02})\Big]$$

which is a superposition of two solitons with respective amplitudes a_1 and a_2, initially located at the points of x-axis z_{01} and z_{01} and which move in the opposite direction, with respective speeds s_1 and s_2. The following numerical values were taken: $a_1 = 0,2$, $a_2 = 0,5$, $z_{01} = 0$, $z_{02} = 25$, $s_1 = 1$, $s_2 = -0,2$. The two solitons interact as particles and run into elastic collision after which each soliton continues its path. The time interval study is $[0, 45]$ and artificial spatial boundary set to $z_L = -20$ and $z_R = -80$. The collision occurs at about $t = 20$, [11, 13, 14].

3. The Numerical Method

In the numerical experiments, we shall consider three popular cases of (1)-(3) obtained using different initial conditions and values of q. First, we describe numerical considerations common to all cases.

We begin by re-defining the pure initial value problem (1)-(3) as an initial boundary value problem in $z_L \leq x \leq z_R$, since over the time interval $[0, T]$ under consideration, the solutions of (1)-(3) of interest to us are negligibly small outside $[z_L; z_R]$. At these boundaries, it is convenient to pose homogeneous Dirichlet or Neumann boundary conditions, [10, 14, 13].

There are three mains components in our numerical method that determine its stability and overall efficacy (resolution vs computational cost): the dynamic mesh generator (or moving mesh PDE) and the time and spatial discretizations. We address separately these components in this section.

3.1. Lagrangian Form

The complex function u is decomposed into its real and imaginary parts w and v respectively (see e.g. Section 1). The resulting system of PDEs is then given by

$$v_t + w_{zz} + qw(v^2 + w^2) = 0, \tag{14}$$

$$w_t - v_{zz} - qv(v^2 + w^2) = 0, \tag{15}$$

$$v(z, 0) = v^0(z), \tag{16}$$

$$w(z, 0) = w^0(z), \tag{17}$$

$$v(z_L, t) = v(z_R, t) = 0, \tag{18}$$

$$v(z_L, t) = v(z_R, t) = 0, \tag{19}$$

where $z \in [z_L, z_R]$ and $t \in [0, T]$. After converting equation (14)-(19) into Lagrangian form [7, 8], central differencing is used for the spatial derivatives and the resulting semi-discrete system on a moving grid is

$$\dot{U}_i = \dot{x}_i \frac{U_{i+1} - U_{i-1}}{h_{i+1} + h_i} - \frac{2}{h_{i+1} + h_i} \times \left(\frac{V_{i+1} - V_i}{h_{i+1}} - \frac{V_i - V_{i-1}}{h_i} \right) - q(U_i^2 + V_i^2)V_i$$

$$\dot{V}_i = \dot{x}_i \frac{V_{i+1} - V_{i-1}}{h_{i+1} + h_i} - \frac{2}{h_{i+1} + h_i} \times \left(\frac{U_{i+1} - U_i}{h_{i+1}} - \frac{U_i - U_{i-1}}{h_i} \right) + q(U_i^2 + V_i^2)U_i$$

for $i = 1, \ldots, N$, where $h_j = x_j - x_{j-1}$, $j = 1, \ldots, N$ and the dot denotes differentiation with respect to time with ξ held constant and $U_j(t)$, $V_j(t)$ are the approximations to $u(x_j, t)$ and $v(x_j, t)$. We apply zero Dirichlet boundary conditions yielding $U_0 = 0$, $U_N = 0$, $V_0 = 0$ and $V_N = 0$.

If we define $U = (U_0,\ V_0,\ U_1,\ V1, ;U_N, V_N)^T$ then we approximate the L_2 norm of U^n on the moving grid by:

$$P = ||U^n||_{L_2}^2 = \sum_{i=1}^{N-1} \left(\frac{h_i^n + h_{i+1}^n}{2} \right) \left((U_i^n)^2 + (V_i^n)^2 \right)$$

which is the discrete analogue of (11), while (12) is approximated by:

$$R = \sum_{i=1}^{N-1} \left(\frac{h_i^n + h_{i+1}^n}{2} \right) \left[\left(\frac{U_{i+1}^n - U_{i-1}^n}{h_i^n + h_{i+1}^n} \right)^2 + \left(\frac{V_{i+1}^n - V_{i-1}^n}{h_i^n + h_{i+1}^n} \right)^2 - \frac{q}{2}((U_i^n)^2 + (V_i^n)^2)^2 \right] + \frac{(U_i^n)^2 + (V_i^n)^2}{2h_1^n} + \frac{(U_{N-1}^n)^2 + (V_{N-1}^n)^2}{2h_N^n}.$$

3.2. The Moving Mesh

The key to developing moving mesh methods lies in formulating a satisfactory mesh equation. It has been recommended that such an equation should be simple, easy to program and reasonably sensitive to the choice of its adjustable parameters, [2, 8, 10]. Huang and his collaborators, [4, 3, 13] have carried out an intensive study of these mesh equations or so-called moving mesh partial differential equations (MMPDEs) based on the equidistribution principle.

Of all the resulting MMPDEs, the most widely used are MMPDE5

$$\dot{u} = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right). \tag{20}$$

The MMPDE7, given by

$$\frac{\partial}{\partial\xi}\left(M\frac{\partial x}{\partial\xi}\right) - 2\frac{\frac{\partial}{\partial\xi}\left(M\frac{\partial x}{\partial\xi}\right)\frac{\partial\dot{u}}{\partial\xi}}{\frac{\partial\dot{u}}{\partial\xi}} = -\frac{1}{\tau}\frac{\partial}{\partial\xi}\left(M\frac{\partial x}{\partial\xi}\right)$$

is said to correspond to the discrete mesh equation used in the method of Dorfi and Drury [3, 12, 13]. We now have an equation which we can use to compute the node speeds and hence the grid position. However, in order to use this, we must first rewrite the PDE in a so-called Lagrangian form.

3.3. Time Discretization

The coupled moving mesh-FNLS system is given by:

$$\dot{x} = (wx_\xi)_\xi, \tag{21}$$

$$\dot{\phi} = i\frac{1}{2x_\xi}\Big(\frac{1}{2x_\xi}\phi_\xi\Big)_\xi + i|\phi|^2\phi + \dot{x}\phi_\xi. \tag{22}$$

This is a very stiff system due to the mesh equation and the second derivative (dispersive) term in the NLS. One numerical approach [4, 6, 13] is to solve (22) and (22) alternately in time; first, (22) is solved for one time step to obtain a new mesh and then this new mesh is turn used to solve (23) also for one time step. This procedure is repeated every time step [9]. In [9], the following discretization was used to compute the moving mesh equation (21) at every time step:

$$\frac{x^{n+1}-x^n}{\triangle t} = a^n x^{n+1}_{\xi\xi} + \Big(w^n x^n_\xi\Big)_\xi - a^n x^n_{\xi\xi},$$

where $a^n = max\ w^n$ and $\triangle t$ is the time step size.

Due to the dispersive nature of the NLS plus the introduction of the mesh advection term $\dot{x}\frac{1}{x_\xi}\phi_\xi$ it is difficult to obtain accurate and stable semi-implicit discretizations for (23) and fully implicit discretizations would too costly. To find a stable and cost efficient scheme we use as a guide the multi-step implicit/explicit (IMEX) methods studied in [2, 3, 5]. We consider two second order IMEX methods applied to (13). The semi backward difference formula (SBDF) scheme, also called extrapolated Gear:

$$\frac{1}{2\triangle t}\Big[3\phi^{n+1} - 4\phi^n + \phi^{n-1}\Big] = \frac{i}{2}\Big(\frac{1}{x_\xi}\Big)^{n+1} \times \Big(\Big(\frac{1}{x_\xi}\Big)^{n+1}\phi^{n+1}_\xi\Big)_\xi$$
$$+ 2\Big[i|\phi^n|^2\phi^n + \dot{x}^n\Big(\frac{1}{x_\xi}\Big)^{n+1}\phi^n_\xi\Big]. \tag{23}$$

When integrating from $t = t^n$ to $t = t^{n+1}$, (23) is considered a system of equations for the approximation of u where the node locations are available at $t = t^n$ and $t = t^{n+1}$. The vector x_ξ is replaced by $(x^{n+1} - x^n) = \triangle x^n$ and x is evaluated in $\left[t^n, t^{n+1}\right]$ using the linear interpolant $x(t) = x_n + x(t - t_n)$, where x_n is the approximation to x at $t = t_n$. The solution at $t = t_{n+1}$ is computed using a second-order singly diagonally implicit Runge-Kutta method which possesses excellent stability properties [6, 11, 15].

3.4. Spatial Discretization

Our spatial discretization scheme is based on simple finite difference schemes on nonuniform grids. This choice has the advantage of simplicity and flexibility. Indeed, finite difference schemes can accommodate any spatial differential operators, whereas the nonlinear Galerkin discretization used in MOVGRD is dedicated to convection-diffusion problems. However, finite difference schemes can be less efficient on some of these problems, and this is why the alternative use of slope limiters can be consider.

Any particular choice of spatial discretization for the NLS (13) affects the stability, accuracy, and cost of the overall adaptive method. Pseudo-spectral approximations have been common in NLS computations. With a uniform mesh, semi-implicit discretizations such as SBDF or CNLF can be inverted at a cost of $O(log_2 N)$ operations using the Fast Fourier Transform (FFT) (see e.g [6, 7, 9]). However, FFT cannot be used for the variable coefficient system produced by a non-uniform mesh. In this case one would have to employ an iterative method at each time-step which would increase the computational cost and may introduce some numerical instability.

We use the so-called method of lines (MOL) approach (see, e.g., [10, 12, 13]) in which the spatial derivatives are first approximated using, for instance, finite difference or finite element techniques so as to convert the PDE problem into a system of (usually stiff) ODEs. The resulting system of ODEs is then integrated in time numerically.

Equation (1) is discretised on the physical domain using central finite differences for the spatial derivatives to yield the semi-discrete system

$$\dot{u}_i = \dot{x}_i \frac{u_{i+1} - u_{i-1}}{h_{i+1} + h_i} - f_i, \;\; i = 1, \ldots, N-1,$$

where $h_i = x_i - x_{i-1}$ and f_i is a suitable spatial discretisation of f in (1). The MMPDE is similarly discretised to obtain the semi-discrete system

$$\dot{x}_i = \frac{4}{\tau}\Big(\tilde{M}_i(h_{i+1}+h_i)\Big)^{-2}\Big(\tilde{M}_{i+\frac{1}{2}}h_{i+1} - \tilde{M}_{i-\frac{1}{2}}h_i\Big) \tag{24}$$

for $i = 1, 2, \ldots, N-1$, with $z_0 = z_L$ and $z_N = z_R$. Here, $\tilde{M}_{i+\frac{1}{2}}$ is a smoothed monitor function, with smoothing carried out as described, with $\tilde{M}_k$ replaced by $\tilde{M}_{k+\frac{1}{2}}$.

The term $\tilde{M}_i$ in (25) is given by

$$\tilde{M}_i = \frac{\tilde{M}_{i-\frac{1}{2}}(z_{i+\frac{1}{2}} - z_i) + \tilde{M}_{i+\frac{1}{2}}(z_i - z_{i-\frac{1}{2}})}{z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}}} \quad \text{where} \quad z_{i+\frac{1}{2}} = \frac{z_{i+1}+z_i}{2}.$$

The integration from $t = t^n$ to $t = t^{n+1}$ is accomplished following the method of Beckett et al. [2, 3, 6]. Equation (25) plays a similar role for $\{x_i\}_{i=0}^N \equiv x$ when $\tilde{M}_{i\pm\frac{1}{2}}$ and $(\tilde{M}_i(h_{i+1}+h_i))^{-2}$ are know for $i = 1, 2, ..., N-1$. The $(\nu+1)th$ iterate is given by

$$x^{[n+1,\nu+1]} = (1+\omega)x_*^{[n+1,\nu+1]} + \omega x^{[n+1,\nu+]}.$$

This equation is discretised in time using the first-order backward Euler method wich possesses excellent stability properties.

It is instructive to consider the FE scheme for (24), which takes the form

$$(x_\xi^h u)_j^{n+1} = (x_\xi^h)_j^n + \frac{\Delta t}{2\Delta\xi}\left[(\kappa(D_+ - D_-)u)_j^n + (\dot{x}^h_{j+\frac{1}{2}} - a)(\delta u)^n_{j+\frac{1}{2}} - (\dot{x}^h_{j-\frac{1}{2}} - a)(\delta u)^n_{j-\frac{1}{2}}\right].$$

Multiplying (25) by u_j^n and following the same analysis as for the BE discretization, we find that

$$\|u^{n+1}\|^2_{n+1} = \|u^n\|^2_n + \|u^{n+1} - u^n\|^2_{n+1} - 2\kappa\Delta t\|D^+u^n\|^2_{\bar{n}}.$$

We now have an anti-diffusive term on the right-hand side caused by the mesh movement

$$\|u^{n+1} - u^n\|^2_{n+1},$$

and hence the scheme will be conditionally stable on a moving mesh. The question arises if it is possible to combine the BE and FE schemes to create a method that is unconditionally stable and second-order accurate in time.

3.5. Monitor Function

Except the classical monitor function, other monitor functions can be used, including

$$m(u) = \sqrt{\alpha + \left\| \frac{d^m h(u)}{dz^m} \right\|_2^k},$$

where the function $h(u) = u$ or $f(u)$ (the flux function), the order of differentiation $m = 1, 2$ and $k = 2, 1$.

Note that for systems of partial differential equations, the computational of (26) implies the calculation of the norm of a vector. In order to compute a discrete approximant of the monitor function M_i in the grid interval $[z_i, z_{i+1}]$, various finite difference schemes can be used, the simplest choice for a first-order derivative being $u_{z,i} = \dfrac{u_{i+1} - u_i}{z_{i+1} - z_i}$.

The choice of the monitor function is very important. The monitor function (26) which is based on the derivative of the solution, has appeared to be the most robust in the method of lines approach [7, 9, 12].

In the description of the method of moving grid in [8], u_z was approximated by finite differences progressive, two points, and the standard used was calculated by taking the average of N_{pde} components, giving the following formula:

$$M_i = \sqrt{\alpha + \frac{1}{N_{pde}} \sum_{j=1}^{N_{pde}} \frac{(U_{i+1}^j - U_i^j)^2}{(Z_{i+1} - Z_i)^2}}, \tag{25}$$

$$M_i = \sqrt{\alpha + \max_j \frac{(U_{i+1}^j - U_i^j)^2}{(Z_{i+1} - Z_i)^2}}, \qquad 0 \le i \le N.$$

We can, for approximating derivatives of order 1, choose a derivative operator digital D_1 calculating u_z ie $(u_z = D_1 U)$ and use again the average or maximum as follows:

$$M_i = \sqrt{\alpha + \frac{1}{N_{pde}} \sum_{j=1}^{N_{pde}} \frac{(U_{z,\,i+1}^j - U_{z,\,i}^j)}{2}}, \qquad \text{or} \tag{26}$$

$$M_i = \sqrt{\alpha + \max_j \frac{(U_{z,\,i+1}^j - U_{z,\,i}^j)}{2}}.$$

The choice of the operator D_1 is enough large if we recall the different stencils: centered, decentered, decentered biased. In the case of hyperbolic PDEs with a flux term $f(u)$, we may consider in the monitor function, the term $f(u)$ rather than the term u and approach the derivative $f(u)_z$ by different approximation formulas of the first derivatives or slopes limiters or flux.

For the monitor function m_2, the options are the following:

$$M_i = \sqrt{\alpha + \frac{1}{N_{pde}} \sum_{j=1}^{N_{pde}} \left| \frac{(U^j_{zz,\,i+1} - U^j_{zz,\,i})}{2} \right|},$$

$$\text{and} \quad M_i = \sqrt{\alpha + \max_j \left| \frac{(U^j_{zz,\,i+1} - U^j_{zz,\,i})}{2} \right|}.$$

In these formulas the second derivative u_{zz} is calculated by the equation $U_{zz} = D_2 U$ where D_2 denotes an operator of numerical derivation for the second derivatives. We can use the operator of three bullet points or five bullet points.

3.6. The Method Algorithm

To provide some measure of global solution difficulty, let us define

$$\eta(t) = \left(\frac{1}{N} \int_{\Omega_p} \sqrt{\alpha + \left\| \frac{d^m h(u)}{dz^m} \right\|_2^k} dz \right)^2.$$

If RTOL is a user-prescribed tolerance and γ and κ are given such that $\gamma > 1$ and $0 < \kappa < 1$, then we would like to ensure that $\gamma RTOL \le \eta(t) \le \kappa RTOL$. Having obtained $\eta(t)$, we can now derive a formula to predict the suitable number of grid cells needed to keep this $'error'$ below a given tolerance, $RTOL$, [9, 13, 15].

We now possess the main ingredients needed to implement the moving mesh method for the nonlinear Shrödinger equation.

if γ RTOL $\le \eta(t) \le \kappa RTOL$ *then*

$$N^{n+1} = N^n$$

$$N^{n+1} = N^n \times \min\left(\max fac, \max\left[\min fac, \kappa \left(\frac{\eta(t)}{RTOL} \right)^{\frac{1}{2}} \right] \right),$$

elseif $\eta(t) > \kappa$RTOL or $\eta(t) < \gamma$ RTOL *then*

Compute N^{n+1} using (29).

end

The new mesh with the increased number of points is obtained by equidistribution of the solution at $t = t_n$ using the new number of points.

3.7. Numerical Simulations of the Algorithm

3.7.1. Solver ODE15s

In MATLAB there are many, ODE solver. There syntaxis is the following: $M(t,Y)Y' = f(t,Y)$; ODE15s is chosen for this specialization in resolution of $stiffs$ problem. This syntaxis is as follows:

$$[T,Y] = ODE15s(ODEFUN,\ TSAN,\ Y0,\ OPTIONS)$$

- vectors T et Y represent respectively instant t_i and solutions $Y(t_i)$ corresponding,
- ODEFUN means the function describing the ODE system is the time interval of integration and intermediate times at which the solution is desired Y_0 is the initial vector
- OPTIONS represents the different options selected for the solver.

The options are set using the function ODESET and we used the following form:

$$\begin{aligned} ODESETOPTIONS = &\big('reltol'valeurRelTol'AbsTol'valeurAbsTol,\\ &'Mass', MatricedeMass'MStateDependance',\\ &'strong','JPattern',\\ &JPat'MvPattern'MvPat'\\ &MassSingular','no','stats','on'\big) \end{aligned}$$

where *reltol* and *AbsTol* denote the relative and absolute error which we can find justification in MATLAB, and for the other options tolerances. We insist on the $MvPattern$ options and $JPattern$ because they play a very important role and must be programmed. The introduction of these options in the code is a trick that allows the ODE integrator to reduce the number of operations and gain computation time, [7, 11].

3.7.2. Mass Matrix Implementation

To use the solver ODE15s we wrote the algorithm in the following general form:

$$\mathbf{A}(t,Y)\dot{U} = \mathbf{b}(t,Y), \tag{27}$$

where

$$U = \left[U_1^1, U_1^2, \ldots U_1^{Npde}, Z_1, \ldots, U_i^1, U_i^2, \ldots U_i^{Npde}, Z_i, \ldots, U_N^1, U_N^2, \ldots U_N^{Npde}, Z_N\right]^T \tag{28}$$

is the global unknown vector, $\mathbf{A}$ the mass matrix of the global algorithm and $\mathbf{b}$ the second associate member.

The construction of the unknown vector U shows that the matrix $\mathbf{A}$ is square and order $(NPDE+1)\times N$ times and has a penta-diagonal-block. The blocks, order $(NPDE+1)$, are of the form

$$A_{ii} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & -D_{i1} \\ 0 & 1 & 0 & \ldots & 0 & -D_{i2} \\ 0 & 1 & 0 & \ldots & 0 & -D_{i3} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & -D_{iNpde} \\ 0 & 0 & 0 & \ldots & 0 & -\tau B_{i,i} \end{bmatrix}, \; A_{ij} = \begin{bmatrix} 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & \tau\mathbf{B}_{i,j} \end{bmatrix},$$

$$A_{ij} = 0, \; 1 \le i \le N, \quad 1 \le j \le N, \quad j \ne i, \quad \mid i-j \mid > 2.$$

The above formulas allow us to write a first program called *mass* and calculates the global matrix $\mathbf{A}$ of the system. Sparse matrix $\mathbf{As}$ associated also square order $(N_{pde}+1)\times N$, penta-diagonal-block is given by the following blocks of order $(N_{pde}+1)$, [7, 8, 12].

$$As_{ii} = \begin{bmatrix} 0 & . & . & . & 0 \\ 0 & . & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & 0 \\ 1 & . & . & . & 1 \end{bmatrix}, \quad As_{i,i+1} = \begin{bmatrix} 1 & 0 & . & . & 0 & 1 \\ 0 & 1 & 0 & . & 0 & 1 \\ . & . & . & . & . & \\ . & . & . & . & . & \\ . & . & . & . & . & \\ 0 & . & . & . & 1 & 1 \\ 1 & . & . & . & . & 1 \end{bmatrix},$$

$$As_{i,i+2} = \begin{bmatrix} 0 & . & . & . & 0 \\ 0 & 0 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & 0 & 0 \\ 0 & . & . & 0 & 1 \end{bmatrix},$$

$$3 \leq i \leq N-2$$

$$As_{i,j} = 0,\ 1 \leq i \leq N, \quad 1 \leq j \leq N, \quad j \neq i, \quad \mid i-j \mid > 2.$$

4. Numerical Results

4.1. Propagation of Single Soliton: SNL1

4.1.1. Reference Solution

The problem defined with the initial condition (12) has the exact solution given by (13).

Figure 1 shows the graphical representation of the reference solution obtained using uniform fixed grid with $N = 1000$ nodes. Table 1 shows the corresponding statistics. The time study interval is equal to $[0, 30]$ with a time step $\triangle t = 5$.

4.1.2. Numerical Solution

The best results were obtained with the following choice of moving grid parameters and numerical derivation operators: $\alpha = 0,5$, $\kappa = 10^{-1}$; $\tau = 2$; $D_2 = five - point - centered$.

The diffusion term u_{zz} is approached by finite difference centered at 5 points. Numerical results obtained with the choice are given in Figure 2. To better appreciate these numerical results we have shown in the same figure the reference and numerical solutions.

The numerical results obtained with a moving grid $N = 100$ points, compared to those of a uniform fixed grid $N = 1000$ nodes, show at Figure 3 and Table 1, a poor accuracy of numerical solutions. Indeed a reduction of the amplitude of the soliton is observed.

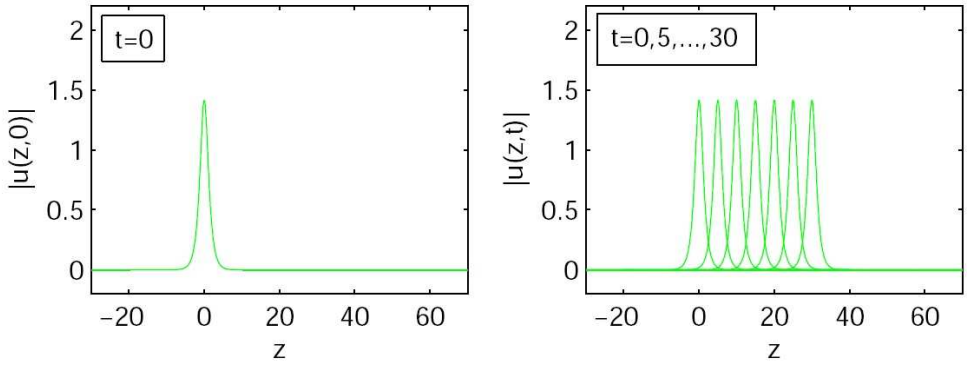

Figure 1: Nonlinear Schrödinger equation, soliton propagation (SNL1): Initial soliton and soliton at times t = 0, 5, ..., 30; reference solution with a fixed uniform grid to N = 1000 nodes.

solutions	STEPS	JACS	FNS	LU	LIN	CPU.t
reference Solution	30	4	8068	12	63	165.1248
numerical Solution	364	183	2924	269	1094	93.2548

Table 1: Nonlinear Schrödinger equation, soliton propagation (SNL1), numerical statistics: reference solution, fixed uniform grid of $N = 2000$ nodes and moving grid $N = 100$ nodes

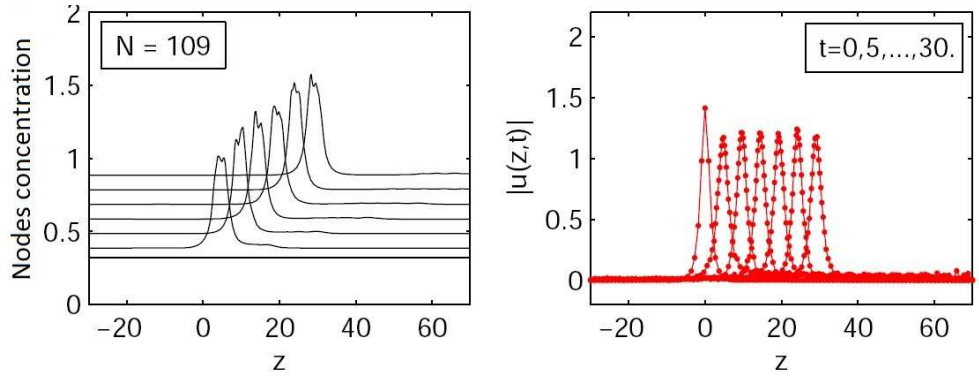

Figure 2: Nonlinear Schrödinger equation, soliton propagation (SNL1): Initial soliton and soliton at times t = 0, 5,..., 30; numerical solution using a moving grid with N = 109.

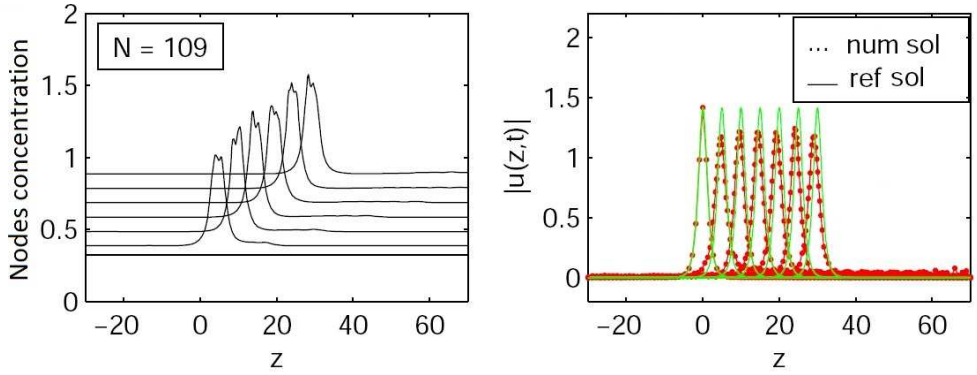

Figure 3: Nonlinear Schrödinger equation, soliton propagation (SNL1): Initial soliton and soliton at times t = 0, 5,..., 30; comparison between reference solution with N = 1000 nodes and numerical solution using a moving grid with N = 109 nodes.

To plotting accuracy, the method has done a good job of capturing the interacting solitons.

Note that with a well adapted initial grid [7], the best results will be obtained.

4.1.3. Adapting the Initial Grid

As the initial conditions (ICs) play a major role in the precision of numerical solutions, it may be judicious to have the best possible approximation, from the departure. Thus, when the initial condition depends on the spatial variable z, we apply the grid adapting process explained in [7].

We applied this option to the nonlinear Schrödinger equation (SNL1) (propagation of a soliton), the graphical results have improved after adjustment of the initial grid, as shown in Figure 4. We have prepared a chart to compare the computing times in both cases: initial grid adapted and non adapted initial grid.

Table 2 shows that in general (2 cases out of 3) one gains in computing time by adapting the initial grid when the initial condition of the problem depends on the independent variable z.

As for the impact on the quality of the graphic results, it is observed that the problem of propagation of a soliton (SNL1) in the non-linear Schrödinger

solutions	STEPS	JACS	FNS	LU	LIN	CPU.t
reference Solution	45	41	9068	13	67	134.1248
numerical Solution	435	179	3980	367	2098	68.2548

Table 2: Nonlinear Schrödinger equation, soliton propagation (SNL1), numerical statistics: reference solution, fixed uniform grid of $N = 1000$ nodes and moving grid $N = 100$ nodes.

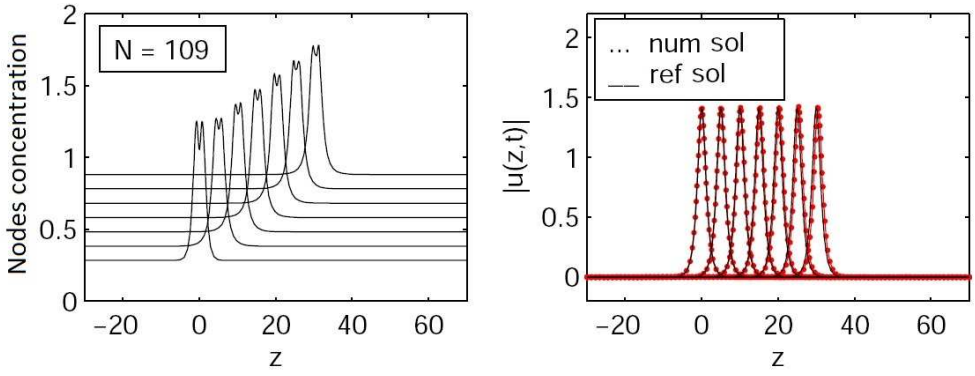

Figure 4: Nonlinear Schrödinger equation; soliton propagation (SNL1): Initial soliton and soliton at times $t = 0, 5, ..., 30$; comparison between the reference solution with $N = 1000$ nodes and the numerical solution using a moving grid, N = 109 nodes, with adaptation of the grid to the initial condition.

equation find an improved results. We can conclude that it is always judicious to adapt the grid to the initial condition of the problem when it depends on the variable z.

4.2. Interaction of Two Solitons: SNL2

4.2.1. Reference Solution

The problem of NLS (14) posed with the initial condition has no analytical solution. The reference solutions were built using a uniform fixed grid with $N = 1000$ nodes. The results obtained are given in Figure 5 and Table 3 statistics.

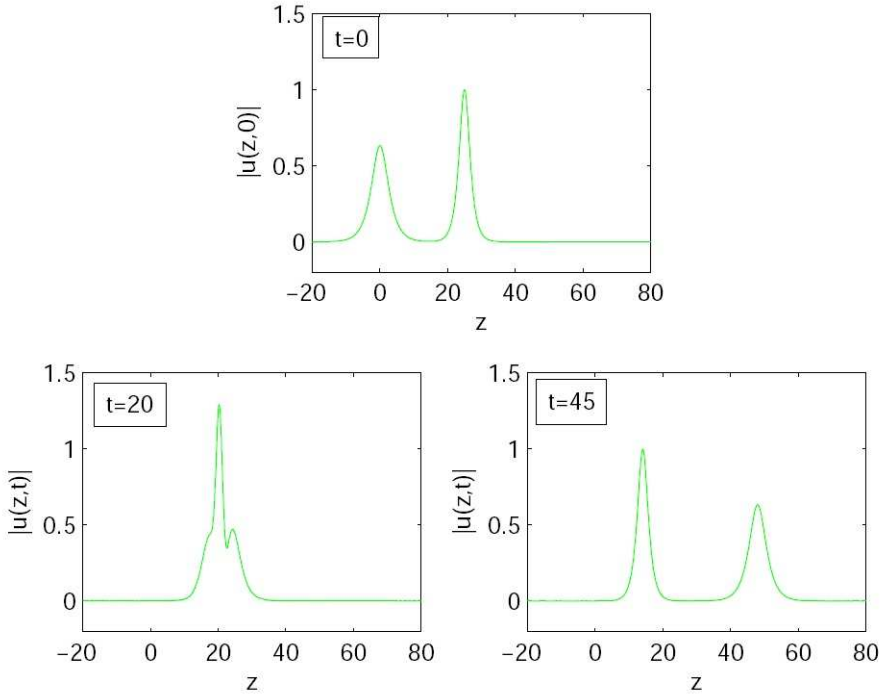

Figure 5: Nonlinear Schrödinger equation; interaction of two solitons (SNL2): solitons at times t = 0, 20, 45; reference solution with a uniform fixed grid N = 1000 nodes.

4.2.2. Numerical Solution

The best results were obtained with the following choice of moving grid parameters and numerical differential operators: $\alpha = 0,8$, $\kappa = 10^{-2}$; $\tau = 1$; $D_2 = three - point - centered$. The diffusion term u_{zz} is approached by finite difference centered at 3 points. Numerical results obtained are given in Figure 6.

To better appreciate these numerical results, we have shown in the same figure the reference solution and numerical solution.

The numerical results obtained with a moving grid $N = 200$ points compared to those of a uniform fixed grid $N = 1000$ nodes, show a good precision of numerical solutions, Figure 7. We note a slight gain in computational time (1914.6 s against 2502.5 s), a gain of around 4/3.

To plotting accuracy, the solution is very good and one can see the periodic-

solutions	STEPS	JACS	FNS	LU	LIN	CPU.t
reference Solution	34	46	9068	17	77	174.1248
numerical Solution	456	189	3456	423	31125	95.2548

Table 3: Nonlinear Schrödinger equation, interaction of two solitons (SNL2), numerical statistics: reference solution, fixed uniform grid of $N = 2000$ nodes and moving grid $N = 100$ nodes.

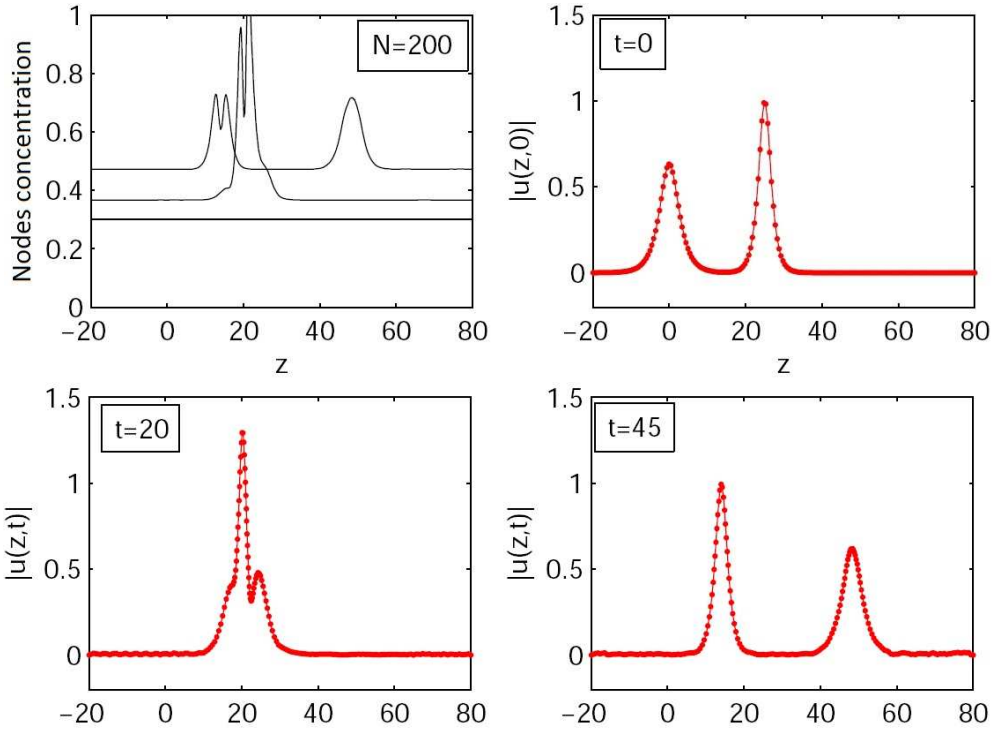

Figure 6: Nonlinear Schrödinger equation, interaction of two solitons (SNL2): grid movement and solitons at times $t = 0, 20, 45$; Numerical solutions using a moving grid with $N = 200$ nodes.

ity expected. For this case, about 200 nodes are effectively used in the region of high solution activity. This suggests that one should be able to capture the solution accurately if an adaptive grid is used which has a comparable resolution in the active region.

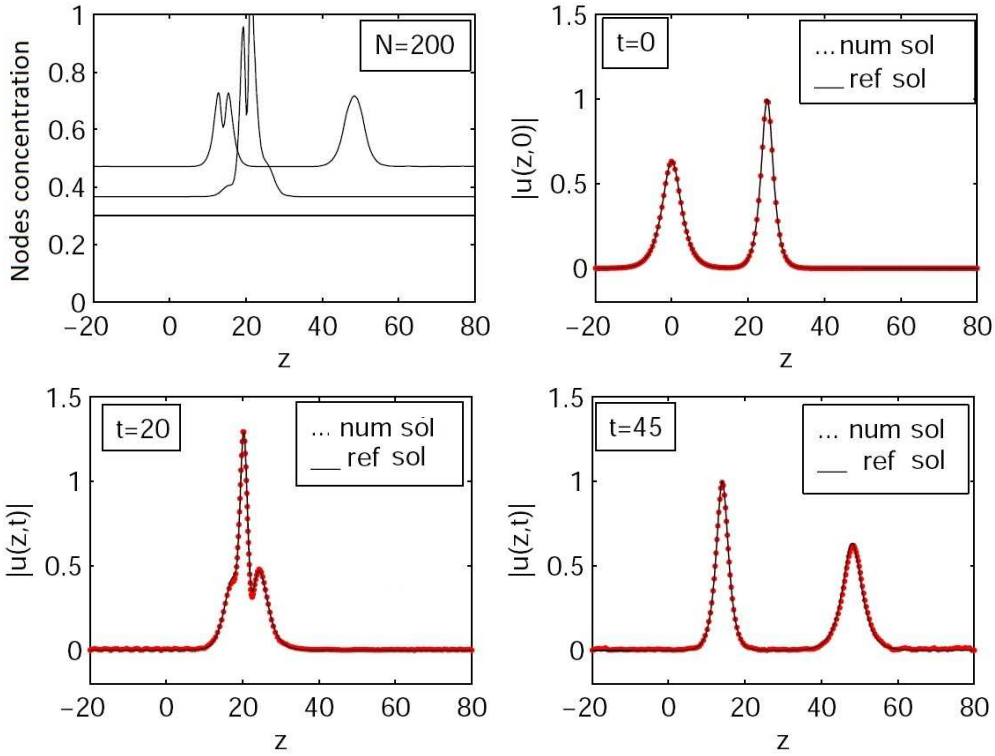

Figure 7: Nonlinear Schrödinger equation, interaction of two solitons (SNL2): grid movement and solitons at times $t = 0, 20, 45$; comparison between reference solutions with $N = 1000$ nodes and digital solutions using a moving grid $N = 200$ nodes.

5. Concluding Remarks

The nonlinear Schrödinger equation presents a great computational challenge. Not only does the method have to resolve accurately the solution self-focusing but also the subsequent highly oscillatory regions of solitons with wavelengths. The experiments with NLS have led us to consider the effect on accuracy due to interpolation which is used when the method is performed. We have applied interpolation as it seems to help solution accuracy. We have shown that the developed method does a good job of resolving the solution of the NLS for the cases we have considered.

We highlight the fact that the adaptive strategy developed here is simple to implement and we have not had to resort to a high order spatial discretization.

The algorithm is both robust and efficient as these experiments have shown. One of the main difficulties in applying adaptive moving mesh methods to this problem originates from the introduction of the mesh advection term into the underlying equation which otherwise has no physical advection at all.

We have shown here that with the right semi-implicit scheme and a high order discretization for the mesh advection term it is possible to obtain accurate, cost-efficient, and stable moving mesh methods. We believe that the method presented here is a very valuable tool and can be used to learn more about the structure of the solutions limiting behavior for non-analytic initial conditions.

References

[1] M. Bakker, Software for semi-discretization of time-dependent partial differential equations in one space variable, *Rep. NW 52/77*, Mathematisch Centrum, Amsterdam (1977).

[2] G. Beckett, J.A. Mackenzie, A. Ramage, D.M. Sloan, On the numerical solution of one-dimensional pdes using adaptive methods based on equidistribution, *J. of Computational Physics*, **167** (2001), 372-392.

[3] E.A. Dorfi, L.O. Drury, Simple adaptive grids for 1-D initial value problems, *Journal of Computational Physics*, **69** (1987), 175-195.

[4] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, *J. of Computational Physics* **171** (2001), 753-775.

[5] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J. of Computational Physics*, **160** (2000), 241-282.

[6] N. Maman, Algorithmes d'adaptation dynamique de maillages en éments finis. Application des coulements ractifs instationnaires, *PhD Thesis*, Université Paul Sabatier de Toulouse (1992).

[7] B. Sangaré, Finite differences method and adaptive grids in the method of lines for partial differential equation, *Int. Journal of R. and Reviews in Appl. Science*, **23**, No 2 (2015), 81-99.

[8] B. Sangaré, O. Diallo and L. Somé, A new MATLAB implementation and analysis of a moving grid method for systems of one-dimensional time-dependent partial differential equations based on the equidistribution principle, *Int. J. Appl. Math. Stat.*, **25**, No 1 (2012), 66-85.

[9] B. Sangaré, B. Somé and L. Somé, A numerical study of two moving mesh method for system of one-dimensional time-dependent partial differential equations which are based on the equidistribution principle, *Int. J. Appl. Math. Stat.*, **33**, No 3 (2013), 28-46.

[10] J.M. Sanz-Serna, I. Christie, A simple adaptive technique for nonlinear wave problems. *J. of Computational Physics*, **67** (1986), 348-360.

[11] P. Saucez, L. Some, A. Vande Wouwer, Matlab implementation of a moving grid method based on the equidistribution principle, *Appl. Math. and Computation*, **215**, No 5 (2009), 1821-1829.

[12] A. Vande Wouwer, P. Saucez, W.E. Schiesser, S. Thompson, A MATLAB implementation of upwind finite differences and adaptive grids in the method of lines, *J. of Computational and Applied Mathematics*, **183** (2005), 245-258.

[13] W.E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press, San Diego (1991).

[14] J.G. Verwer, J.G. Blom, R.M. Furzeland, P.A. Zegeling, A moving grid method for one-dimensional PDEs based on the method of lines. In: *Adaptive Methods for Partial Differential Equations* (J.E. Flaherty, P.J. Paslow, M.S. Shepard, J.D. Vasilakis, Eds.), SIAM, Philadelphia, (1989), 160-175.

[15] Wankere Rinwi Mekwi, *Adaptive Solution of Partial Differential Equations Using Mesh Movement and Refinement*, Dept. of Math., University of Strathclyde, Glasgow (2006).