Bulletin of Mathematical Analysis and Applications
ISSN: 1821-1291, URL: http://www.bmathaa.org
Volume 14 Issue 1(2022), Pages 31-45.
https://doi.org/10.54671/bmaa-2022-1-4

GALERKIN APPROXIMATION FOR ONE-DIMENSIONAL WAVE EQUATION BY QUADRATIC B-SPLINES

NOURIA ARAR

Abstract. This work is devoted to the development of a Galerkin-type approximation of the solution of a wave equation, using quadratic B-Spline functions and a 2-centred finite difference scheme. Two examples are used to validate the proposed approximation. The numerical results obtained show the effectiveness of the procedure for very small times. This makes it attractive for the approximation of PDEs with not known explicit solution.

1. Introduction

Many physical phenomena can be described using the properties of wave propagation. We can cite the waves propagating on the surface of the water following the fall of an object, the waves moving on the surface of the sea, the seismic waves moving on the ground and the sound waves, they are called acoustic waves. Another type of waves are electromagnetic waves such as light and radio waves. In this article, we are mainly interested in the acoustic wave equation [8, 10, 11, 15, 17, 20], which is the simplest model (scalar model) but which is already very rich because it allows to address the main concepts common to all these models. This is one of the best known equations of mathematical physics, owing to Jean Le Rond d'Alembert (1747). It is of the form

$$\frac{\partial^2 u}{\partial t^2} - \alpha^2 \frac{\partial^2 u}{\partial x^2} = f(t, x).$$

Much work has already been devoted to the development of effective numerical methods to solve this type of problem. In the 1930s, finite differences, were the first methods for discretizing wave equations [19]. Despite their great simplicity in terms of writing and low calculation cost, these remain limited to simple geometries and the Courant, Friedrichs and Lewy (CFL) stability condition was proved to be necessary for convergence. Another difficulty with finite difference methods lies in the implementation of the boundary condition with the same numerical precision

2000 *Mathematics Subject Classification.* 65D07 , 65N30 , 65N22 , 65N06.
Key words and phrases. Wave equation; Finite differences; Galerkin method; Finite elements; Quadratic B-splines.

Submitted September 25, 2021. Published March 10, 2022.
Communicated by A. Zafer.
N. Arar was supported by Directorate General for Scientific Research and Technological Development (DGRSDT), Algeria.

as inside the domain. We also mention the method of the finite volumes, which can treat complex geometries and deal well with the boundary conditions of Neumann type. However, few theoretical results of convergence can be established. On the other hand, the finite elements method is worth to mention since many theoretical convergence results are possible. Nevertheless, in the classical finite elements method, the approximate solution is the only continuous function. Now, in many applications, as in computer graphics, it is preferable to use functions having at least one continuous derivative. This property will be satisfied in the finite elements method by B-Splines [6, 7, 14, 16, 18].
Bases consisting of B-splines are well-conditioned, at least for orders less than 20. These are functions with compact support; in the sense that at every point only a fixed number (equal to the order) of B-Splines is nonzero. They are defined piecewise by a polynomial on each interval between nodes and having at least the first derivative continuous to the right and to the left in each node.
Several works have been established using B-Splines. These include, for example, data fitting, function approximation, numerical quadrature, and the numerical solution of operator equations such as those associated with ordinary and partial differential equations as in [1, 2, 3, 13].
When used alone, any numerical method has its own problems and limitations. However, it is more beneficial to combine two methods and profit from their advantages while minimising their respective disadvantages. An example is a combination of finite differences and finite elements methods using B-Splines.
In this paper, a homogeneous one-dimensional wave equation with initial and boundary conditions is transformed into a set of linear ordinary differential equations with deduced boundary conditions, by applying a 2-centred finite difference scheme. Then, the numerical solution of every one of the ODEs is built up with a Galerkin approximation using a quadratic B-spline basis by virtue of their simplicity of calculation and implementation. The performance of the displayed method has been tested on two problems with known analytic solutions.

2. The model problem

Consider the problem of a wave equation given by

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - \alpha^2 \dfrac{\partial^2 u}{\partial x^2} = 0, & (x,t) \in [a,b] \times [0,T] \\ u(a,t) = u(b,t) = 0, & 0 < t < T \\ u(x,0) = g(x), & a < x < b \\ \dfrac{\partial u}{\partial t}(x,0) = h(x), & a < x < b \end{cases} \tag{2.1}$$

where α is a positive parameter. The solution is given by the d'Alembert formula in [17]

$$u(x,t) = \frac{1}{2}\left(g(x+\alpha t) + g(x-\alpha t)\right) + \frac{1}{2\alpha}\int_{x-\alpha t}^{x+\alpha t} h(\xi)\, d\xi, \tag{2.2}$$

if $g \in \mathscr{C}^2(\mathbb{R})$ and $h \in \mathscr{C}^1(\mathbb{R})$.

3. Construction of numerical model

3.1. **Time discretization of (2.1).** First, let us start with time discretization where the interval $[0, T]$ is subdivided into p subintervals $I_j = [t_{j-1}, t_j]$, of the same lenght $\Delta t = t_j - t_{j-1}$ for $j = \overline{1, p}$ and $t_0 = 0$.
Considering a scheme with the centred difference of order 2 to approach $\frac{\partial^2 u}{\partial t^2}$, the method of discretization in time consists in finding successively for $j = \overline{1, p}$ the functions $U_j = U_j(x)$ which are solutions of the problems

$$\begin{cases} \dfrac{1}{\Delta t^2} U_{j+1} = \dfrac{2}{\Delta t^2} U_j - \dfrac{1}{\Delta t^2} U_{j-1} + \alpha^2 \dfrac{d^2 U_{j+1}}{dx^2}, & x \in [a, b] \\ U_0(x) = g(x), & a < x < b \\ U_1(x) = g(x) + \Delta t \cdot h(x), & a < x < b \end{cases} \tag{3.1}$$

where U_0 is the initial condition whereas U_1 is obtained using an implicit sheme for approximating the initial condition

$$\frac{\partial u}{\partial t}(x, 0) = h(x), \quad a < x < b$$

Note that U_1 can be obtained by using Taylor expansion at time Δt :

$$u(x, \Delta t) = u(x, 0) + \Delta t \frac{\partial u}{\partial t}(x, 0) + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2}(x, 0)$$

and from

$$\frac{\partial^2 u}{\partial t^2} = \alpha^2 \frac{\partial^2 u}{\partial x^2} \quad \text{and} \quad \frac{\partial u}{\partial t}(x, 0) = h(x), \quad a < x < b$$

we have

$$U_1(x) = g(x) + \Delta t \cdot h(x) + \frac{\Delta t^2}{2} \alpha^2 g''(x). \tag{3.2}$$

3.2. **Spatial discretization of (3.1).** This is accomplished on two steps

Step 1: Solution of the problem (3.1) starts with the construction of a weak formulation needed in Galerkin approximation.
Let v be a test function. The weak formulation of (3.1) is given by

$$\int_a^b v(x) \left(\frac{1}{\Delta t^2} U_{j+1} - \frac{2}{\Delta t^2} U_j + \frac{1}{\Delta t^2} U_{j-1} - \alpha^2 \frac{d^2 U_{j+1}}{dx^2} \right) dx = 0$$

which can be written, after integration by parts, for all $j = \overline{1, p}$

$$\begin{gathered} \frac{1}{\Delta t^2} \int_a^b v(x) U_{j+1}(x) dx + \alpha^2 \int_a^b v'(x) U'_{j+1}(x) dx \\ = \\ \frac{2}{\Delta t^2} \int_a^b v(x) U_j(x) dx - \frac{1}{\Delta t^2} \int_a^b v(x) U_{j-1}(x) dx + \alpha^2 \left[v(x) U'_{j+1}(x) \right]_a^b. \end{gathered} \tag{3.3}$$

Step 2: Next, let us subdivide the interval $[a,b]$ into N subintervals of length $h=\frac{b-a}{N}$ by the nodes x_i given by $x_i=a+ih,\ (i=0,1,\dots,N)$. Then as a basis approximation space take the set of quadratic B-splines $\{B_{-1},B_0,\cdots,B_N\}$ defined by

$$B_m(x)=\frac{1}{h^2}\begin{cases}(x_{m+2}-x)^2-3(x_{m+1}-x)^2+3(x_m-x)^2 & [x_{m-1},x_m]\\ (x_{m+2}-x)^2-3(x_{m+1}-x)^2 & [x_m,x_{m+1}]\\ (x_{m+2}-x)^2 & [x_{m+1},x_{m+2}]\\ 0 & \text{otherwise}\end{cases}$$

where $h=x_{m+1}-x_m;\ \ m=\overline{-1,N}$, (for more details see [9, 12]).
Note that each quadratic spline $B_m(x)$ and its first derivative vanish outside the interval $[x_{m-1},x_{m+2}]$. The values of $B_m(x)$ and $B'_m(x)$ at the nodes are given in the following table

x	x_{m-1}	x_m	x_{m+1}	x_{m+2}
$B_m(x)$	0	1	1	0
$B'_m(x)$	0	$2/h$	$-2/h$	0

The approximation of the solution $U_{j+1}(x)$ can be expressed in the basis $\{B_m\}$ by

$$U_{j+1}^N(x)=\sum_{n=-1}^{N}c_n^{j+1}B_n(x) \tag{3.4}$$

where c_n^{j+1} are (unique) coefficients to be determined.
As each spline covers three intervals $x_{m-1}\le x_{m+2}$ so that three B_{m-1}, B_m, and B_{m+1} cover each finite element $[x_m,x_{m+1}]$, all other splines are zero in this region. Then from the equation (3.4), the values $U_{j+1,m}$ of the solution and the values $U'_{j+1,m}$ of the derivative at the nodes x_m are given by

$$\begin{aligned}U_{j+1,m}&=U_{j+1}(x_m)\\ &=c_{m-1}^{j+1}+c_m^{j+1}\end{aligned} \tag{3.5}$$

and

$$\begin{aligned}U'_{j+1,m}&=U'_{j+1}(x_m)\\ &=\frac{2}{h}(c_m^{j+1}-c_{m-1}^{j+1}).\end{aligned} \tag{3.6}$$

From boundary conditions, we get

$$c_{-1}^{j+1}=-c_0^{j+1}\qquad\text{and}\qquad c_N^{j+1}=-c_{N-1}^{j+1}.$$

Hence, approximation (3.4) takes the following form

$$U_{j+1}^N(x)=\sum_{n=0}^{N-1}c_n^{j+1}Q_n(x) \tag{3.7}$$

where, (as defined in [4])

$$\begin{aligned}Q_0(x)&=B_0(x)-B_{-1}(x)\\ Q_m(x)&=B_m(x); \qquad\qquad m=\overline{1,N-2}\\ Q_{N-1}(x)&=B_{N-1}(x)-B_N(x).\end{aligned}$$

Thus, there remains N unknowns c_m^{j+1} $(m=\overline{0,N-1})$ to be determined.
Galerkin's approximation starts of replacing in the weak formulation (3.3) $v(x)$ by

$Q_m(x)$ $(m = \overline{0, N-1})$. Then, substituting (3.5), (3.6) and (3.7) in the obtained equation yields

$$\frac{1}{\Delta t^2}\sum_{n=0}^{N-1} c_n^{j+1}\int_a^b v(x)Q_n(x)dx + \alpha^2\sum_{n=0}^{N-1} c_n^{j+1}\int_a^b v'(x)Q_n'(x)dx$$
$$=$$
$$\frac{2}{\Delta t^2}\int_a^b v(x)U_j(x)dx - \frac{1}{\Delta t^2}\int_a^b v(x)U_{j-1}(x)dx + \alpha^2\big[v(x)U_{j+1}'(x)\big]_a^b.$$

But, $Q_m(x_0) = Q_m(x_N) = 0$ for all $m = \overline{0, N-1}$. So, for $v(x) = Q_m(x)$, we have

$$\big[v(x)U_{j+1}'(x)\big]_a^b = 0.$$

The latter equation becomes for all $m = \overline{0, N-1}$ and $j = \overline{1, p}$

$$\frac{1}{\Delta t^2}\sum_{n=0}^{N-1}\left(\int_a^b Q_m(x)Q_n(x)dx\right)c_n^{j+1} + \alpha^2\sum_{n=0}^{N-1}\left(\int_a^b Q_m'(x)Q_n'(x)dx\right)c_n^{j+1}$$
$$=$$
$$\frac{2}{\Delta t^2}\int_a^b Q_m(x)U_j(x)dx - \frac{1}{\Delta t^2}\int_a^b Q_m(x)U_{j-1}(x)dx. \tag{3.8}$$

which can be written in the matrix form:

$$\left(\frac{1}{\Delta t^2}\mathcal{A} + \alpha^2\mathcal{B}\right)c^{j+1} = \frac{1}{\Delta t^2}(2D^j - E^{j-1}) \tag{3.9}$$

where

$$c^{j+1} = (c_0^{j+1}, c_1^{j+1}, c_2^{j+1}, \cdots, c_{N-1}^{j+1})^T; \quad j = \overline{1, p}$$
$$D^j = (D_0^j, D_1^j, \cdots, D_{N-1}^j)^T$$
$$\text{with} \quad D_m^j = \int_a^b Q_m(x)U_j(x)dx; \quad m = 0, \dots, N-1.$$

and

$$E^{j-1} = (E_0^{j-1}, E_1^{j-1}, \cdots, E_{N-1}^{j-1})^T$$
$$\text{with} \quad E_m^{j-1} = \int_a^b Q_m(x)U_{j-1}(x)dx; \quad m = 0, \dots, N-1.$$

The matrices $\mathcal{A}$, $\mathcal{B}$ are $N \times N$ penta-diagonal matrices defined by

$$\mathcal{A}_{mn} = \int_a^b Q_m(x)Q_n(x)dx, \quad \text{and} \quad \mathcal{B}_{mn} = \int_a^b Q_m'(x)Q_n'(x)dx.$$

Solving the system (3.9) is done in three steps

Step 1: $j = 1$: A starting vector c^2 must first be determined from the initial conditions in (2.1) by solving the following system

$$\left(\frac{1}{\Delta t^2}\mathcal{A} + \alpha^2\mathcal{B}\right)c^2 = \frac{1}{\Delta t^2}(2D^1 - E^0) \tag{3.10}$$

where

$$c^2 = (c_0^2, c_1^2, c_2^2, \cdots, c_{N-1}^2)^T; \quad j = \overline{1, p}$$
$$E^0 = (E_0^0, E_1^0, \cdots, E_{N-1}^0)^T$$
$$\text{with} \quad E_m^0 = \int_{x_{m-1}}^{x_{m+2}} Q_m(x)g(x)dx; \quad m = 0, \dots, N-1.$$

and

$$D^1 = (D_0^1, D_1^1, \cdots, D_{N-1}^1)^T$$

$$\text{with} \quad D_m^1 = \int_{x_{m-1}}^{x_{m+2}} Q_m(x)(g(x) + \Delta t \cdot h(x))dx; \quad m = 0, \dots, N-1$$

$$\text{or} \quad D_m^1 = \int_{x_{m-1}}^{x_{m+2}} Q_m(x) \left(g(x) + \Delta t \cdot h(x) + \frac{\Delta t^2}{2} \alpha^2 g''(x) \right) dx.$$

Step 2: $j = 2$: The vector c^3 is determined from the initial condition concerning the derivative and the vector c^2 calculated in the first step, by solving the following system

$$\left(\frac{1}{\Delta t^2} \mathcal{A} + \alpha^2 \mathcal{B} \right) c^3 = \frac{1}{\Delta t^2} (2D^2 - E^1) \tag{3.11}$$

where

$$E^1 = (E_0^1, E_1^1, \cdots, E_{N-1}^1)^T$$

$$\text{with} \quad E_m^1 = \int_{x_{m-1}}^{x_{m+2}} Q_m(x) U_1(x) dx = D_m^1; \quad m = 0, \dots, N-1.$$

Then

$$E^1 = D^1 \quad \text{and then} \quad E^j = D^j, \ \forall j \geq 1.$$

On the other hand

$$D^2 = (D_0^2, D_1^2, \cdots, D_{N-1}^2)^T$$

$$\text{with} \quad D_m^2 = \int_{x_{m-1}}^{x_{m+2}} Q_m(x) U_2(x) dx; \quad m = 0, \dots, N-1.$$

or

$$U_2^N(x) = \sum_{n=0}^{N-1} c_n^2 Q_n(x)$$

then

$$D_m^2 = \sum_{n=0}^{N-1} \left(\int_{x_{m-1}}^{x_{m+2}} Q_m(x) Q_n(x) dx \right) c_n^2; \quad m = 0, \dots, N-1.$$

So

$$D^2 = \mathcal{A} c^2 \quad \text{and then} \quad D^j = \mathcal{A} c^j, \ \forall j \geq 2.$$

Step 3: $j \geq 3$: After having calculated c^2 and c^3, we solve system (3.9) by using a recurrence on the following system, for all $j \geq 3$

$$\left(\frac{1}{\Delta t^2} \mathcal{A} + \alpha^2 \mathcal{B} \right) c^{j+1} = \frac{1}{\Delta t^2} \mathcal{A} (2c^j - c^{j-1}) \tag{3.12}$$

4. Numerical tests

In this section, we present the numerical solutions in the following two examples whose respective analytic solution is known. The results obtained from the B-spline approximation method will be compared with those of the 2-centred finite difference scheme. The precision of the method will be measured at each moment with the L_∞ error norm defined by

$$ME = \|\text{Exact sol.} - U^N\|_\infty \simeq \max_j |\text{Exact sol}_j - U_j^N|$$

and compared with the similar error obtained with the second order centred finite difference method.

All computations were done using the PC with an hp Pavilion, Intel(R) Core(TM) i3-3217U CPU@ 1.80GHz. All the programming is implemented in MATLAB 9.0. The rate of convergence is computed approximately by the following formula

$$\text{Order} = \frac{\ln\left(\left|\text{Exact sol.} - U^N_{h_j}\right| / \left|\text{Exact sol.} - U^N_{h_{j+1}}\right|\right)}{\ln\left(h_j/h_{j+1}\right)}, \tag{4.1}$$

where h_j is the spatial discretization step Note that for the approximation of U_1 we used the formula in (3.1) instead of (3.2).

Example 1. *Set $\alpha = 0.5$ in (2.2) and taking the following boundary conditions:*

$$u(-3\pi, t) = u(3\pi, t) = 0; \quad \forall\ t \in [0, 10]$$

and the initial conditions

$$u(x, 0) = \cos(x/2), \quad \frac{\partial u}{\partial t}(x, 0) = 0,\ \forall\ x \in [-3\pi, 3\pi].$$

The exact solution of (2.1) is easily shown to be as follows

$$u(x, t) = \cos(x/2)\cos(t/4).$$

The results presented in Table 1 through Table 6 are obtained with time discretization steps $\Delta t = 0.1$, 0.05 and 0.01 at times $t = 4$ and $t = 7$ respectively. The behavior of the maximal error committed is observed by increasing the number of nodes for the method of quadratic B-splines (BS-M), whereas the error becomes very large with the second order centred finite difference method (FD-M) from $N \geq 512$ for $\Delta t = 0.1$, $N \geq 1024$ for $\Delta t = 0.05$, and $N \geq 4096$ for $\Delta t = 0.01$. Moreover, the error obtained for the approximation by the quadratic B-splines is of the order of 10^{-3} for $\Delta t = 0.05$ and is of 10^{-4} for $\Delta t = 0.01$ whatever the number of points considered. This implies that the approximation produced by the quadratic B-splines method, for a small number of nodes and in a short time may be sufficient to conjecture it as being the desired solution.

Table 1. Maximal error for Example 1 with $\Delta t = 0.1$ and $t = 4$.

N	FD-M	BS-M
128	1.068364×10^{-2}	6.536689×10^{-3}
256	1.054402×10^{-2}	6.536612×10^{-3}
512	5.990567×10^{11}	6.536607×10^{-3}

Table 2. Maximal error for Example 1 with $\Delta t = 0.1$ et $t = 7$.

N	FD-M	BS-M
128	1.264294×10^{-2}	4.206455×10^{-3}
256	1.235262×10^{-2}	4.206468×10^{-3}
512	1.191619×10^{33}	4.206468×10^{-3}

Figure 1 and Figure 2 show the effectiveness of the proposed approximation for only $N = 256$ and in a short time since the $\Delta t = 0.1$ as mentioned above. Figure 3 and Figure 4 illustrate the maximal error obtained for Example 1 when $\Delta t = 0.1$, $N = 256$ at $t = 4$ by BS-approximation method and FD-method respectively. It is

TABLE 3. Maximal error for Example 1 with $\Delta t = 0.05$ et $t = 4$.

N	FD-M	BS-M
128	5.441953×10^{-3}	3.322631×10^{-3}
256	5.300873×10^{-3}	3.322553×10^{-3}
512	5.265599×10^{-3}	3.322548×10^{-3}
1024	1.948620×10^{40}	3.322548×10^{-3}

TABLE 4. Maximal error for Example 1 with $\Delta t = 0.05$ et $t = 7$.

N	FD-M	BS-M
128	6.526843×10^{-3}	2.027483×10^{-3}
256	6.235820×10^{-3}	2.027495×10^{-3}
512	6.163062×10^{-3}	2.027496×10^{-3}
1024	1.104524×10^{83}	2.027496×10^{-3}

TABLE 5. Maximal error for Example 1 with $\Delta t = 0.01$ et $t = 4$.

N	FD-M	BS-M
128	1.241290×10^{-3}	6.732863×10^{-4}
256	1.099043×10^{-3}	6.732082×10^{-4}
512	1.063477×10^{-3}	6.732033×10^{-4}
1024	1.054585×10^{-3}	6.732032×10^{-4}
2048	1.052362×10^{-3}	6.732032×10^{-4}
4096	4.330312×10^{126}	6.732055×10^{-4}

TABLE 6. Maximal error for Example 1 with $\Delta t = 0.01$ et $t = 7$.

N	FD-M	BS-M
128	1.618322×10^{-3}	3.930522×10^{-4}
256	1.326734×10^{-3}	3.930646×10^{-4}
512	1.253835×10^{-3}	3.930654×10^{-4}
1024	1.235610×10^{-3}	3.930655×10^{-4}
2048	1.231054×10^{-3}	3.930656×10^{-4}
4096	1.494652×10^{23}	3.930641×10^{-4}

very clear that the BS-approximation is much better than the FD-approximation. In Figure 5 and Figure 6, we display comparison of the exact solution with BS-approximation, and the exact solution with FD-approximation respectively.
The converge rates computed, for example 1, by the present method for values of space size h_j and a fixed value of the time step Δt are recorded in Table 8. It is clearly seen that the scheme provides reductions in convergence rates for the smaller space sizes.

TABLE 7. Order of convergence at $t = 10$, $\Delta t = 0.001$

N	h_j	ME	Order
16	$3\pi/8$	5.303660×10^{-4}	–
32	$3\pi/16$	2.676937×10^{-4}	9.864051×10^{-1}
64	$3\pi/32$	2.514317×10^{-4}	9.041663×10^{-2}
128	$3\pi/64$	2.504165×10^{-4}	5.836928×10^{-3}
256	$3\pi/128$	2.503538×10^{-4}	3.612713×10^{-4}
512	$3\pi/256$	2.503503×10^{-4}	2.016933×10^{-5}
1024	$3\pi/512$	2.503490×10^{-4}	7.491537×10^{-6}

TABLE 8. Order of convergence as a function of Δt and N, at $t = 10$.

N	Δt_j	ME	Order
256	$1/10$	2.494313×10^{-2}	–
256	$1/20$	1.249634×10^{-2}	0.997137
256	$1/40$	6.253762×10^{-3}	0.998709
256	$1/80$	3.128203×10^{-3}	0.999390
256	$1/160$	1.564424×10^{-3}	0.999703
256	$1/320$	7.822930×10^{-4}	0.999851
256	$1/640$	3.911678×10^{-4}	0.999921

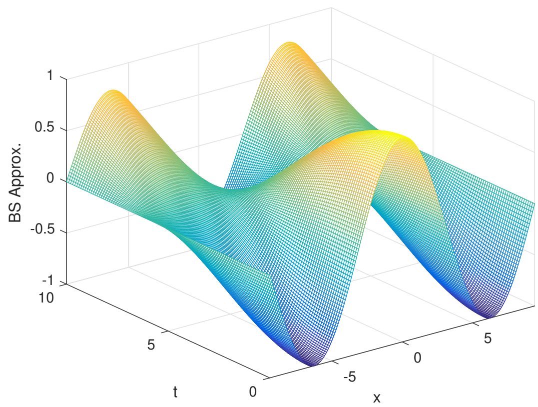

FIGURE 1. BS-approximation of Example 1 for $\Delta t = 0.1$ and $N = 256$.

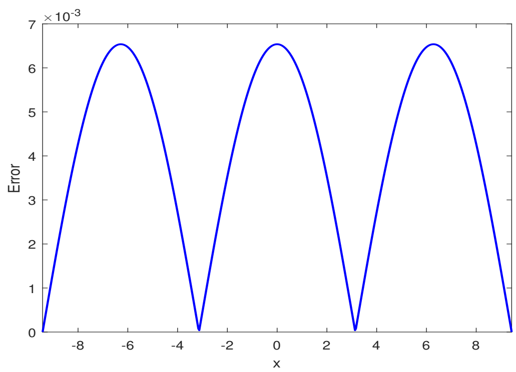

FIGURE 3. Maximal error of Example 1 at $t = 4$ for $\Delta t = 0.1$ and $N = 256$ by BS-M

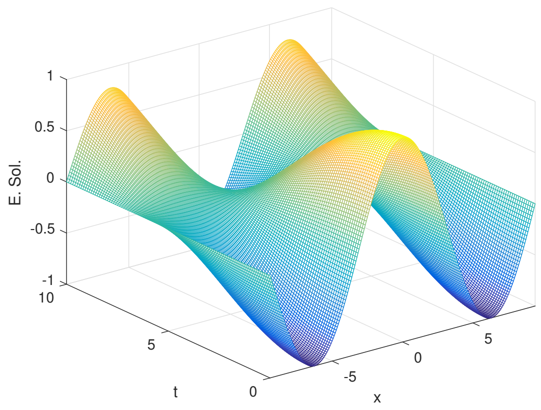

FIGURE 2. Exact solution of Example 1 for $\Delta t = 0.1$ and $N = 256$

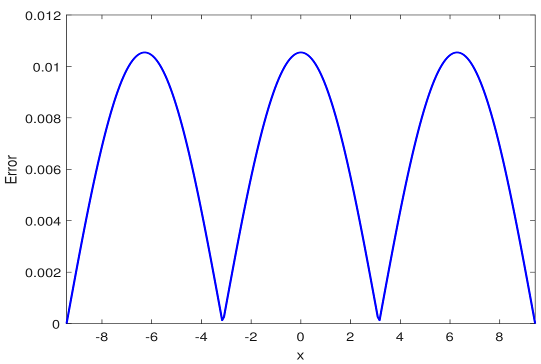

FIGURE 4. Maximal error of Example 1 at $t = 4$ for $\Delta t = 0.1$ and $N = 256$ by FD-M

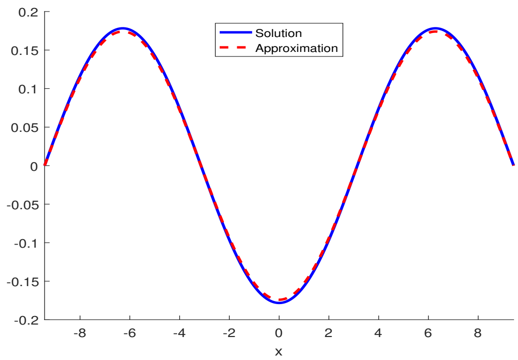

FIGURE 5. Comparison of the exact solution and its BS-M approximation for $\Delta t = 0.1$, $N = 256$ at $t = 7$.

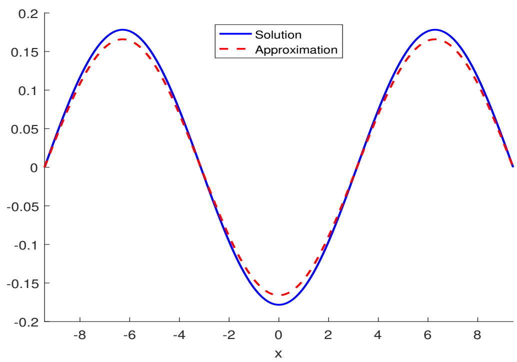

FIGURE 6. Comparison of the exact solution and its FD-M approximation for $\Delta t = 0.1$, $N = 256$ at $t = 7$.

Example 2. *Set $\alpha = 1$ in (2.2) and taking the following boundary conditions:*

$$u(0,t) = u(2,t) = 0; \quad \forall\, t \in [0,2]$$

and the initial conditions

$$u(x,0) = 0, \quad \frac{\partial u}{\partial t}(x,0) = \sin(\frac{3\pi}{2}x),\ \forall\, x \in [0,2].$$

The exact solution of (2.1) is shown to be as follows

$$u(x,t) = \frac{2}{3\pi}\sin(\frac{3\pi}{2}x)\sin(\frac{3\pi}{2}t), \quad \forall (x,t) \in [0,2]\times[0,2].$$

The results presented in Table 9 through Table 14 are obtained with time discretization steps $\Delta t = 0.1$, 0.05 and 0.01 at times $t = 0.6$ and $t = 1.4$ respectively. The behavior of the maximal error committed is observed by increasing the number of nodes for the method of approximation by the quadratic B-splines, whereas the error becomes very large when calculating the approximation by the second order centred finite difference method from $N \geq 128$. In this example again we find the

stability and convergence of the quadratic B-spline method for a small number of nodes and in a short time in comparison with the finite-centred difference method which seems unstable.
Figure 7 and Figure 8 show the effectiveness of the proposed approximation for $N = 1024$ and $\Delta t = 0.001$.

TABLE 9. Maximal error for Example 2 with $\Delta t = 0.1$ et $t = 0.6$.

N	FD-M	BS-M
128	3.546268×10^{-3}	5.800394×10^{-2}
256	4.748272×10^{-3}	5.800097×10^{-2}
512	2.794580	5.800060×10^{-2}
1024	5.627875×10^{3}	5.800055×10^{-2}

TABLE 10. Maximal error for Example 2 with $\Delta t = 0.1$ et $t = 1.4$.

N	FD-M	BS-M
128	8.173375×10^{11}	4.731669×10^{-2}
256	4.961335×10^{19}	4.731405×10^{-2}
512	3.976005×10^{27}	4.731371×10^{-2}

TABLE 11. Maximal error for Example 2 with $\Delta t = 0.05$ et $t = 0.6$.

N	FD-M	BS-M
128	2.585296	4.460768×10^{-2}
256	1.348315×10^{7}	4.460607×10^{-2}
512	7.199611×10^{13}	4.460587×10^{-2}

TABLE 12. Maximal error for Example 2 with $\Delta t = 0.05$ et $t = 1.4$.

N	FD-M	BS-M
128	3.364519×10^{25}	2.316473×10^{-2}
256	2.320939×10^{42}	2.316302×10^{-2}
512	5.006166×10^{58}	2.316280×10^{-2}

TABLE 13. Maximal error for Example 2 with $\Delta t = 0.01$ et $t = 0.6$.

N	FD-M	BS-M
128	1.091746×10^{-4}	1.271944×10^{-2}
256	2.292106×10^{19}	1.271943×10^{-2}
512	2.317129×10^{63}	1.271942×10^{-2}

TABLE 14. Maximal error for Example 2 with $\Delta t = 0.01$ et $t = 1.4$.

N	FD-M	BS-M
128	1.447045×10^{-4}	2.118835×10^{-3}
256	9.157291×10^{69}	2.118836×10^{-3}
512	7.554487×10^{173}	2.118835×10^{-3}

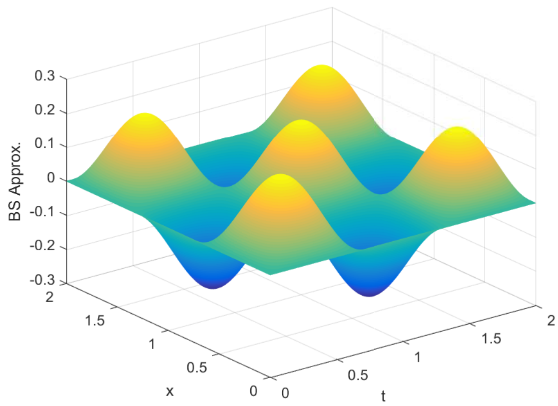

FIGURE 7. BS-approximation of Example 2 for $\Delta t = 0.001$ and $N = 1024$.

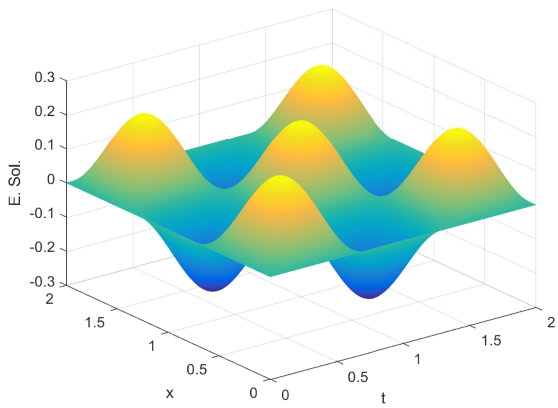

FIGURE 8. Exact solution of Example 2 for $\Delta t = 0.001$ and $N = 1024$

Conclusion. The use of quadratic B-spline for the approximation of the wave equation leads to a convergent method in a rather small and especially stable time. This fact is lost when using a finite difference scheme in some cases. Performance of method has been shown in terms of maximal error between exact solution and

BS-approximation in the Tables by studying two test problems. We conclude that it may be preferable to use a quadratic B-spline approximation to obtain the numerical solutions of the differential equations at short times particularly for the EDP whose explicit solution is not known.

Acknowledgments. The author would like to thank the anonymous referee for his/her comments that helped me improve this article.

References

[1] T. AK and S. B. G. Karakoc and A. Biswas, *A new approach for numerical solution of modified Korteweg-de Vries equation*, Iranian Journal of Science and Technology, Transactions A: Science 41 **4** (2017), 1109–1121. `https://doi.org/10.1007/s40995-017-0238-5`

[2] T. AK and S. B. G. Karakoc and A. Biswas, *Application of Petrov-Galerkin method to shallow water waves model: Modified Korteweg-de Vries equation*, Scientia Iranica B 24 **3** (2017), 1148–1159. `https://doi.org/10.24200/sci.2017.4096`

[3] T. Ak, *An application of Galerkin method to generalized Benjamin-Bona-Mahony-Burgers equation*, Adýyaman University Journal of Science 8 **2** (2018), 53–69. `https://dergipark.org.tr/en/download/article-file/620140`

[4] E. N. Aksan and A. Özdes, *Numerical solution of Korteweg-de Vries equation by Galerkin B-spline finite element method*, Applied Mathematics and Computation **175** (2006), 1256–1265. `https://doi.org/10.1016/j.amc.2005.08.038`

[5] G. Allaire, *Analyse numérique et optimisation*, Edition de l'école polytechnique, France, 2012. `https://www.editions.polytechnique.fr/files/pdf/EXT_1255_8.pdf`

[6] Carl de Boor, *On calculating with B-splines*, Journal of approximation theory **6** (1972), 50–62. `https://doi.org/10.1016/0021-9045(72)90080-9`

[7] M. G. Cox, *The Numerical Evaluation of a Spline from its B-Spline Representation*, IMA Journal of Applied Mathematics 21 **2** (1978), 135–143. `https://doi.org/10.1093/imamat/21.2.135`

[8] J. Chandezon and G. Cornet, *Application d'une nouvelle méthode de résolution des équations de Maxwell à l'étude de la propagation des ondes électromagnétiques*, dans les guides périodiques **36** (1981), 305–314. `https://doi.org/10.1007/BF03000609`

[9] M. Gheorghe and M. Sanda, *Handbook of Splines*, Kluwer Academic Publishers, London, 1999. `https://doi.org/10.1007/978-94-011-5338-6`

[10] V. Girault, *Résolution numérique d'une équation des ondes avec condition aux limites non linéaire par une méthode de caractéristiques*, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, Tome 1 **4** (1967), 51–62. `http://www.numdam.org/item/M2AN_1967_1_4_51_0/`

[11] J. S. Hesthaven and S. Gottlieb and D. Gottlieb, *Spectral methods for time dependent problems*, Cambridge Univ. Press., 2009. `https://doi.org/10.1017/CBO9780511618352.001`

[12] W. J. Kammerer and G. W. Reddien and R. S. Varga, *Quadratic Interpolating Splines : Theory and Applications* , Numerical Solution of Partial Differential Equations **4** (1973), 151–172. `https://doi.org/10.1007/978-94-010-2672-7_6`

[13] T. Özis and A. Esen and S. Kutluay, *Numerical solution of Burgers' equation by quadratic B-spline finite element*, Appl. Math. Comput. **165** (2005), 237–249. `https://doi.org/10.1016/j.amc.2004.04.101`

[14] P. M. Prenter, *Splines and Variational Methods*, New York (NY): John Wiley, 1975. `https://doi.org/10.1002/zamm.19770570718`

[15] A. Quarteroni and F. Saleri and P. Gervasio, *Calcul scientifique. Cours, exercices corrigés et illustrations en MATLAB et Octave*, Springer-Verlag, Italia, 2010. `https://doi.org/10.1007/978-88-470-1676-7`

[16] S. S. Rana, *Quadratic spline interpolation*, Journal of approximation theory **57**(3) (1989), 300–305. `https://doi.org/10.1016/0021-9045(89)90045-2`

[17] S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*, Springer-Verlag, Italia, 2016. `https://doi.org/10.1007/978-3-319-31238-5`

[18] R. A. Usmani, *On quadratic spline interpolation*, BIT Numerical Mathematics **27** (1987), 615–622. `https://doi.org/10.1007/BF01937280`

[19] Vidar Tommé, *From finite differences to finite elements A short history of numerical analysis of partial differential equations*, Journal of Computational and Applied Mathematics **128** (2001), 1–54. https://doi.org/10.1016/S0377-0427(00)00507-0

[20] G. B. Whitham, *Linear and nonlinear waves*, John Wiley & Sons, INC, Pure & Applied Mathematics, 1999. https://doi.org/10.1002/9781118032954

Nouria Arar
Laboratoire de mathématiques et sciences de la décision (LAMASD)
Département de mathématiques
Université Frères Mentouri, Constantine 1, Algeria
E-mail address: arar.nouria@umc.edu.dz