

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.Sciencedirect.com)

# Results in Physics

journal homepage: [www.journals.elsevier.com/results-in-physics](http://www.journals.elsevier.com/results-in-physics)

## Integration of the hyperbolic telegraph equation in (1+1) dimensions via the generalized differential quadrature method



A. Zerarka\*, S. Guergueb

Labo of Applied Mathematics, University Med Khider, BP145, 07000 Biskra, Algeria

### ARTICLE INFO

#### Article history:

Received 23 November 2012

Accepted 25 January 2013

Available online 9 February 2013

#### Keywords:

Quadrature

Interpolating polynomial

Hyperbolic equation

### ABSTRACT

The 2D generalized differential quadrature method (hereafter called ((1+1)-GDQ) is introduced within the context of dynamical system for solving the hyperbolic telegraph equation in (1+1) dimensions. Best efficiency is obtained with a low-degree polynomial ( $n \leq 8$ ) for both time variable and  $x$ -direction. From realistic examples, some models are presented to illustrate an excellent performance of the proposed method, compared with the exact results.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/3.0/).

### Contents

1. Introduction	20
2. The GDQ method formulation	21
2.1. Optimization of weighting coefficients	21
2.2. Simulation algorithm	21
2.3. Some remarks	22
3. Illustrative examples	22
3.1. First test example	22
3.2. Second test example	22
4. Conclusion	22
Acknowledgments	23
References	23

### 1. Introduction

A class of many methods [1–10] has been proposed to describe a wide variety of linear and non-linear dynamical equations. Among these dynamical equations there is that of the hyperbolic telegraph equation (HT) [11–13]. Because of its importance to the field of engineering and many physical systems, a great deal of approximate schemas and numerical calculations have appeared

in recent years to calculate the hyperbolic and parabolic equations for numerous physical situations [14–20].

The first part of this paper serves to introduce the use of a (1,1)-GDQ method [21–28] in order to explore the structure of the solutions of the hyperbolic telegraph equation. In our description, the solutions are evaluated in a polynomial form. Based on the properties of the Lagrange's polynomial, this method improves of a considerable manner the efficiency and accuracy of results compared with the other usual methods much argued in the literature.

The remainder of this paper is organized as follows. In “The GDQ method formulation”, we present the theoretical bases which underlie the formulation of the GDQ method in connection with the problems of interest. In “Illustrative examples”, some applications are proposed to concretize the method and conclusion and trends for future work are made in “Conclusion”.

\* Corresponding author. Tel.: +213 0662887540.

E-mail address: [abzerarka@yahoo.fr](mailto:abzerarka@yahoo.fr) (A. Zerarka).

## 2. The GDQ method formulation

In this section our effort is concentrated essentially on the connection of the (1+1) GDQ method to some important physical problems in which the hyperbolic telegraph equation is considered

$$hw = g(x, t), \quad x \in [A, B], \quad t \geq 0, \quad (1)$$

where the linear differential operator  $h$  is defined by:

$$h = a \frac{\partial}{\partial t} + \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} + b, \quad (2)$$

in which  $a$  and  $b$  are free parameters.

The term  $[hw]$  can be expressed as a constant coefficient eigenfunction combination at all discrete points in the domain of variables  $x$  and  $t$  as

$$[hw]_{ik} = \sum_{j=1}^M \tau_{kj} w(x_i, t_j) + \sum_{j=1}^N \chi_{ij} w(x_j, t_k) \quad (3)$$

for  $i = 1, \dots, N$  and  $k = 1, \dots, M$ ,

the relation (3) can be rewritten as

$$[hw]_{ik} = (\tau_{kk} + \chi_{ii}) w(x_i, t_k) + \sum_{j=1, j \neq k}^M \tau_{kj} w(x_i, t_j) + \sum_{j=1, j \neq i}^N \chi_{ij} w(x_j, t_k) = g(x_i, t_k), \quad \text{for } i = 1, \dots, N \text{ and } k = 1, \dots, M, \quad (4)$$

where  $\{x_j\}$ ,  $1 \leq j \leq N$  and  $\{t_j\}$ ,  $1 \leq j \leq M$  are the sequences of the  $x$  and  $t$ -variables called the collocation points, and the function  $w(x, t)$  is a family of polynomials and specifically taken as the Lagrange interpolated polynomial as

$$w(x, t) = \sum_{j=1}^M \sum_{i=1}^N w(x_i, t_j) P_{N,i}(x) P_{M,j}(t), \quad (5)$$

and

$$P_{l,k}(z) = \frac{G_l(z)}{(z - z_k) G_l^{(1)}(z_k)}, \quad k = 1, \dots, l, \quad (6)$$

where  $G_l(z) = \prod_{j=1}^l (z - z_j)$  and  $G_l^{(1)}(z_i) = \frac{\partial G_l(z)}{\partial z} \Big|_{z=z_i} = \prod_{j=1, j \neq i}^l (z_i - z_j)$

Using the different derivatives of  $w(x, t)$  at  $x_i$  and  $t_k$  from (5), it follows that from the properties of  $P_{l,k}(z)$ , we can establish the unknown weighting coefficients  $\tau_{ij}$  and  $\chi_{ij}$  as

$$\begin{cases} \tau_{kj} = a\alpha_{kj}^{(1)} + \alpha_{kj}^{(2)}, \\ \text{for } k = 1, \dots, M \text{ and } j = 1, \dots, M \text{ and} \\ \tau_{kk} = - \sum_{j=1, j \neq k}^M \tau_{kj}, \end{cases} \quad (7)$$

and

$$\begin{cases} \chi_{ij} = -\beta_{ij}^{(2)}, \quad \text{for } j \neq i \text{ and} \\ \chi_{ii} = b - \sum_{j=1, j \neq i}^N \chi_{ij}, \quad \text{for } i = 1, \dots, N. \end{cases} \quad (8)$$

The superscripts 1 and 2 in parentheses do not indicate powers, but merely identify the derivatives of the Lagrange's polynomial with which the quantities  $\alpha_{kj}^{(n)}$  and  $\beta_{ij}^{(2)}$  are associated.

$$\alpha_{kj}^{(n)} = \frac{d^n P_j(t_k)}{dt^n}, \quad n = 1, 2, \quad (9)$$

$$\beta_{ij}^{(n)} = \frac{d^n P_j(x_i)}{dx^n}, \quad n = 1, 2. \quad (10)$$

### 2.1. Optimization of weighting coefficients

Moreover, it is readily to obtain from relations (6), (8) and (10), the coefficients  $\tau_{kj}$  and  $\chi_{ij}$  in a reduced form

$$\begin{cases} \tau_{kj} = \alpha_{kj}^{(1)} \left[ a + 2 \left( \alpha_{kk}^{(1)} - \frac{1}{(t_k - t_j)} \right) \right] \\ \text{for } j \neq k, \quad j = 1, \dots, M \text{ and } k = 1, \dots, M, \text{ and} \\ \tau_{kk} = - \sum_{j=1, j \neq k}^M \tau_{kj} \quad \text{for } k = 1, \dots, M \end{cases} \quad (11)$$

and

$$\begin{cases} \chi_{ij} = -2\beta_{ij}^{(1)} \left( \beta_{ii}^{(1)} - \frac{1}{(x_i - x_j)} \right) \\ \text{for } j \neq i, \quad j = 1, \dots, N \text{ and } i = 1, \dots, N, \text{ and} \\ \chi_{ii} = b - \sum_{j=1, j \neq i}^N \chi_{ij} \quad \text{for } i = 1, \dots, N \end{cases} \quad (12)$$

and the fundamental terms  $\alpha_{ik}^{(1)}$  and  $\beta_{ik}^{(1)}$  in the relations (11) and (12) are given respectively as follows

$$\begin{cases} \alpha_{ik}^{(1)} = \frac{G_M^{(1)}(t_i)}{(t_i - t_k) G_M^{(1)}(t_k)}, \quad \text{for } i \neq k, \quad i = 1, \dots, M \text{ and } k = 1, \dots, M, \text{ and} \\ \alpha_{ii}^{(1)} = - \sum_{j=1, j \neq i}^M \alpha_{ij}^{(1)} \end{cases} \quad (13)$$

and

$$\begin{cases} \beta_{ik}^{(1)} = \frac{G_N^{(1)}(x_i)}{(x_i - x_k) G_N^{(1)}(x_k)}, \quad \text{for } i \neq k, \quad i = 1, \dots, N \text{ and } k = 1, \dots, N, \text{ and} \\ \beta_{ii}^{(1)} = - \sum_{j=1, j \neq i}^N \beta_{ij}^{(1)} \end{cases} \quad (14)$$

So, the coefficients  $\alpha_{ij}^{(1)}$  and  $\beta_{ij}^{(1)}$  are known through the relations (13) and (14). Technically, the other coefficients  $\tau_{ij}$  and  $\chi_{ij}$  are computed via (11) and (12) by using systematically the results obtained from (13) and (14). Having found the weighting coefficients  $\tau_{ij}$  and  $\chi_{ij}$ , one can accurately solve the following matrix equation and therefore the original problem (1)

$$[\gamma]W = G \quad (15)$$

in the above expression,  $[\gamma]$  is a  $N \times M$  matrix with elements  $\tau_{ij}$  and  $\chi_{ij}$ ,  $W$  is a column vector with components  $(w(x_1, t_1), w(x_1, t_2), \dots, w(x_N, t_M))$ , and  $G$  is a column vector with components  $(g(x_1, t_1), g(x_1, t_2), \dots, g(x_N, t_M))$ .

At this stage, we have to solve this linear system of equations that arises from the main relation (4). Once the linear system (15) is solved and the components  $w(x_i, t_j)$  are known, the corresponding solution  $w(x, t)$  of Eq. (1) is then determined completely.

A more complete description will be given later on with two specific examples.

### 2.2. Simulation algorithm

The system of algebraic Eq. (15) is an explicit prescription that gives the solution in  $O(N \times M)$  operations. Eq. (15) must also be understood as a vector equation. We have to solve the  $N \times M$  set of linear algebraic equations. Hence, we summarize the main steps for this algorithm as,

Step 1: Input the  $x_i$  and  $t_j$  adequate interpolation points, and construct the interpolation polynomials  $P_{N,k}(x)$  and  $P_{M,l}(t)$  with (6). The interpolation points can be evaluated as  $x_i = i\Delta x$ ,  $1 \leq i \leq N$ , and at  $t_j = j\Delta t$ ,  $1 \leq j \leq M$  where  $\Delta x$  and  $\Delta t$  are the space and time steps respectively.

Step 2: Calculate the  $\tau_{ij}$  and  $\chi_{kl}$  elements by relations (11) and (12), for  $i = 1, \dots, M$ ,  $j = 1, \dots, M$  and  $k = 1, \dots, N$ ,  $l = 1, \dots, N$ , together with (13) and (14). To simulate the weighting coefficients, a specific code is also written in Mathematica to generate the

matrix elements  $\tau_{ij}$  and  $\chi_{kl}$  at any order, once the set of interpolation points is given.

Step 3: We evaluate the algebraic system (15).

Step 4: Once the function values at all grid points are obtained, it is then easy to determine the function values in the overall domain in terms of polynomial approximation, such that  $w(x, t) = \sum_{j=1}^M \sum_{i=1}^N w(x_i, t_j) P_{N,i}(x) P_{M,j}(t)$ .

2.3. Some remarks

- (1) The trouble with the standard Lagrangian polynomial technique is that we do not know which degree of polynomial to use. If the degree is too low the interpolating polynomial does not give good estimates of function. If the degree is too high, undesirable oscillations in polynomial values can occur. For this, we can adopt a strategy on choice of points used in construction of interpolating polynomials.
- (2) The difficulty in the applications of quadrature method is how to find of manner accurate and efficient the weighting coefficients involved in the strategy of interpolation techniques. It can be seen that (11)–(14) are very simple algebraic expressions for  $\tau_{ij}$  and  $\chi_{kl}$ . Note also that, the choice of grid coordinates is not restricted and can be taken freely.
- (3) If we know a priori that the function is smooth, a low-degree polynomial should work satisfactorily.

3. Illustrative examples

To illustrate how this method works, and see whether it is robust, a good test may be made by examining two examples. To be more precise and for the benefit of comparison, we use therefore the problems with known analytical solutions.

3.1. First test example

More interesting is the case of the hyperbolic telegraph equation of the form Eq. (1) with the following parameters:  $a = 1, b = 1, A = 0, B = 1$  and the right hand side function is  $g(x, t) = x^2 + t - 1$ . The exact solution is written as [11]

$$w(x, t) = x^2 + t. \tag{16}$$

The space-time graph of approximated solution  $w(x, t)$  is given in Fig. 1.

The accuracy of the  $w(x, t)$  estimation was evaluated using the absolute error. On the other hand, good agreement was found in the solution  $w(x, t)$ . The results were essentially the same. The accuracy of this work has been examined using the absolute error as

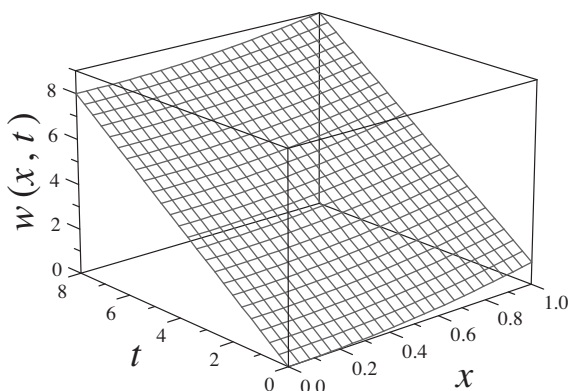


Fig. 1. Representation of the estimated solution  $w(x, t)$  for the first example.

Table 1 Absolute error at times  $t = 5$ , and  $t = 10$ , for  $w(x, t)$ .

$x$	Absolute error, $t = 5$ .	Absolute error, $t = 10$ .
0.	$1.46795 \times 10^{-10}$	$2.23556 \times 10^{-10}$
0.0669873	$1.60378 \times 10^{-10}$	$1.97907 \times 10^{-10}$
0.5	$2.01696 \times 10^{-10}$	$1.16964 \times 10^{-10}$
0.933013	$1.66925 \times 10^{-10}$	$1.76694 \times 10^{-10}$
1.	$1.54341 \times 10^{-10}$	$1.98916 \times 10^{-10}$

Table 2 Absolute error at times  $t = 1.5$  and  $t = 2.25$  for  $w(x, t)$ .

$x$	Absolute error, $t = 1.5$	Absolute error, $t = 2.25$
0.785398	$8.32667 \times 10^{-16}$	$3.19189 \times 10^{-16}$
1.5708	$1.19349 \times 10^{-15}$	$5.13478 \times 10^{-16}$
2.35619	$1.38778 \times 10^{-16}$	$2.08167 \times 10^{-16}$
3.14159	$1.10891 \times 10^{-15}$	$1.70925 \times 10^{-16}$
3.92699	$5.27356 \times 10^{-16}$	$1.94289 \times 10^{-16}$
4.71239	$6.66134 \times 10^{-16}$	$5.55112 \times 10^{-17}$
5.49779	$1.11022 \times 10^{-16}$	$8.32667 \times 10^{-17}$

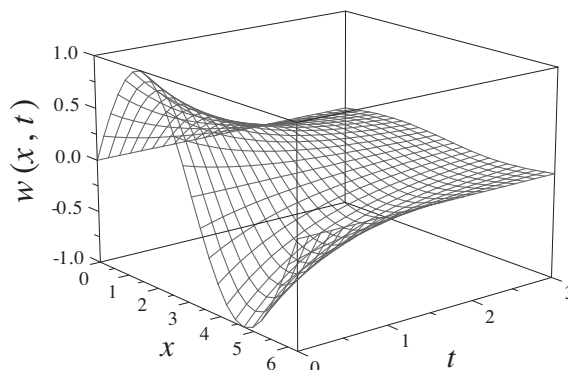


Fig. 2. Representation of the estimated solution  $w(x, t)$  for the second example.

$$\text{Absolute error} = \|\text{Estimated} - \text{Analytical}\| \tag{17}$$

We report in Table 1, the absolute errors at times  $t = 5$ , and  $t = 10$ .

3.2. Second test example

Another example that has analytically known solutions and provides a further application of the method presented in this work is the hyperbolic telegraph Eq. (1) with the following data [15]:  $a = 8, b = 4, A = 0, B = 2\pi, t \in [0, 3]$  and  $g(x, t) = (2 - a + b) \sin(x) \exp(-t)$ . The exact solution is given by:

$$w(x, t) = \sin(x) \exp(-t) \tag{18}$$

Following a similar fashion to that used for the first example, this case, may be numerically tested by using the results of Section 2. The absolute errors at times  $t = 1.5$  and  $t = 2.25$  are shown in Table 2.

The plot of the space-time graph of estimated solution  $w(x, t)$  is given in Fig. 2. All the computations were performed by using Mathematica.

4. Conclusion

In this work a (1+1)-GDQ method is investigated that determines the structures of the hyperbolic telegraph equation in

(1+1) dimensions. A family of compactly supported orthogonal polynomial is constructed to find the appropriate solutions.

We obtained, that the numerical examples given in this work have shown the efficiency of this procedure. Moreover, the present method provides a reliable technique that requires less work if compared with the difficulties arising from computational aspect. In view of its simplicity and elegance with which it may be implemented, this method is also more desirable to work with and the corresponding solutions are commonly in good agreement with exact results.

Alternatively, this method can be always qualitatively extended to multi-dimensional problems and coupled systems where the above developments are always applicable. So, would be interesting, to show the application of this method by examining the case where the solution cannot be handled analytically with the usual techniques. This direction of inquiry is under consideration and is envisioned for a sequel to this paper.

### Acknowledgments

The authors want to thank Prof. Hans and Dr S. Brend for interesting discussions and for their valuable comments. AZ would like to thank the M.E.R.S (Ministère de l'Enseignement et de la Recherche Scientifique) for financial support: PNR contract No. 30/15/2011.

### References

- [1] Mohyud-Din ST, Yildirim A, Sariaydin S. Numerical soliton solution of the Kaup-Kupershmidt equation. *Int J Numer Methods Heat Fluid Flow*, Emerald 2011;21(3):272–81.
- [2] Mohyud-Din ST, Yildirim A, Demirli G. Analytical solution of wave system in with coupling controllers. *Int J Numer Methods Heat Fluid Flow*, Emerald 2011;21(2):198–205.
- [3] Mohyud-Din ST, Noor MA. Homotopy perturbation method for solving partial differential equations. *Zeitschrift fr Naturforschung A – A J Phys Sci* 2009;64a:157–70.
- [4] Mohyud-Din ST, Yildirim A, Sariaydin S. Numerical soliton solutions of the improved Boussinesq equation. *Int J Numer Methods Heat Fluid Flow* 2011;21(7):822–7.
- [5] Abdusalam HA. Analytic and approximate solutions for Nagumo telegraph reaction diffusion equation. *Appl Math Comput* 2004;157:515–22.
- [6] Amaratunga K, Williams JR, Qian S, Weiss J. Wavelet –Galerkin solution for one-dimensional partial differential equations. *Int J Numer Methods Eng* 1994;37:705–2716.
- [7] Daubechies I. Ten lectures on wavelets. Philadelphia: SIAM; 1992.
- [8] Glowinski R, Lawton WM, Ravachol M, Tenenbaum E. Wavelet solutions of linear and nonlinear elliptic, parabolic, and hyperbolic problems in one space dimension. *Comput Methods Appl Sci Eng* 1990:5–120.
- [9] Ho SL, Yang SY. Wavelet–Galerkin method for solving parabolic equations in finite domains. *Finite Element Anal Des* 2001;37:023–1037.
- [10] Mohanty RK, Jain MK, George K. On the use of high order difference methods for the system of one space second order non-linear hyperbolic equations with variable coefficients. *J Comput Appl Math* 1996;72:421–31.
- [11] El-Azab MS, El-Gamel M. A numerical algorithm for the solution of telegraph equations. *Appl Math Comput* 2007;190:757–64.
- [12] Dehghan M, Shokri A. A numerical method for solving the hyperbolic telegraph equation. *Numer Methods Partial Differ Equat* 2008;24:1080–93.
- [13] Mohebbi A, Dehghan M. High order compact solution of the one-space-dimensional linear hyperbolic equation. *Numer Methods Partial Differ Equat* 2008;24:1222–35.
- [14] Saadatmandi A, Dehghan M. Numerical solution of the one-dimensional wave equation with an integral condition. *Numer Methods Partial Differ Equat* 2007;23:282–92.
- [15] Gao F, Chi C. Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation. *Appl Math Comput* 2007;187:1272–6.
- [16] Aloy R, Casaban MC, Caudillo-Mata IA, Jodar L. Computing the variable coefficient telegraph equation using a discrete eigenfunctions method. *Comput Math Appl* 2007;54:448–58.
- [17] Mohanty RK, Jain MK. An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation. *Numer Methods Partial Differ Equat* 2001;17:684–8.
- [18] Mohanty RK, Jain MK, Arora U. An unconditionally stable ADI method for the linear hyperbolic equation in three space dimensions. *Int J Comput Math* 2002;79:133–42.
- [19] Guo BY. The state of art in spectral methods. Hong Kong: City University of Hong Kong; 1996.
- [20] Braun M, Sofianos SA, Papageorgiou DG, Lagaris IE. An efficient Chebyshev–Lanczos method for obtaining eigensolutions of the Schrödinger equation on a grid. *J Comput Phys* 1996;126(315).
- [21] Shu C, Richards BE. High resolution of natural convection in a square cavity by generalized differential quadrature. *Proc 3rd Int Conf Adv Numer Methods Eng: Theory Appl* 1990;11:978–85.
- [22] Shu C, Richards BE. Application of generalized differential quadrature to solve two-dimensional incompressible Navier–Stokes equations. *Int J Numer Methods Fluids* 1992;15:791–8.
- [23] Zerarka A, Hassouni S, Saidi H, Boumedjane Y. Energy spectra of the Schrödinger equation and the differential quadrature method. *Commun Nonlinear Sci Numer Simulat* 2005;10.
- [24] Zerarka A, Khelil N, Saidi H. A generalised integral quadratic method: improvement of the solution for one dimensional Volterra integral equation using particle swarm optimisation. *Int J Simulat Process Model* 2006;2(1/2).
- [25] Zerarka A, Saidi H, Hassouni S, Bensalah N. Evaluation of the bound states of a quantum system via the differential quadrature method: extended to coupled differential equations. *Appl Math Comput* 2006;182:665–71.
- [26] Saidi H, Khelil N, Hassouni S, Zerarka A. Energy spectra of the Schrödinger equation and the differential quadrature method: Improvement of the solution using particle swarm optimization. *Appl Math Comput* 2006;182:559–66.
- [27] Boumedjane Y, Saidi H, Hassouni S, Zerarka A. Some first excited energy levels for the generalized Killingbeck potential with the differential quadratic method. *Appl Math Comput* 2007;194:43–249.
- [28] Zerarka A, Soukeur A, Khelil N. The particle swarm optimization against the Runge's phenomenon: Application to the generalized integral quadrature method. *Int J Comput Math Sci* 2009;3(7).