#### **Research Article**

# Rostamy\*

## Hybrid of differential quadrature and sub-gradients methods for solving the system of Eikonal equations

https://doi.org/10.1515/nleng-2021-0035 Received Jul 10, 2021; accepted Oct 25, 2021

Abstract: Many important natural phenomena of wave propagations are modeled by Eikonal equations and a variety of new methods are needed to solve them. The differential quadrature method (DQM) is an effective numerical method for solving the system of differential equations that can achieve accurate numerical results using fewer grid points and therefore requires relatively little computational effort. In this paper, we focus on the implementation of the non-smooth Eikonal optimization by using a hybrid of polynomial differential quadrature (PDQ) or Fourier differential quadrature (FDQ) method and sub-gradients idea. Our goal is to develop a new Eikonal equation system design for wave propagation equations, as well as the efficiency and accuracy of new grid points to reduce errors and compare errors in various physical equation problems, especially wave propagation equations, and achieve their convergence. We explore the accuracy and stability of the Eikonal equation system by two-dimensional and three-dimensional numerical examples and the use of three types of grid points in a comprehensive manner. This article aims to create a new and innovative solution to the non-smooth Eikonal equation system. This new method is much more efficient and effective than traditional numerical solution methods same as DQ.

**Keywords:** Non-smooth problem, optimization, wave propagation, Seismic wave, polynomial differential quadrature (PDQ), fourier differential quadrature (FDQ), 2D Eikonal equation system, 3D Eikonal equation system.

# 1 Introduction

Eikonal differential equations are the most important geometric equations that are widely used in many different fields of mathematical science and engineering. One of the most important applications of these equations is seismic waves. They are studied by geophysicists called seismologists. Seismic wave fields are recorded by a seismometer, hydrophone (in water). Also, seismic waves are energy waves that pass through the earth's layers. The result is earthquakes, volcanic eruptions, large-scale landslides, and large man-made explosions that produce lowfrequency sound energy. All of them are somehow modeled with the equations of this paper. Many phenomena are caused by low-amplitude waves, which are called ambient disturbances [1–4] and references therein.

Such as the propagation of a wave in a moving fluid that has been extensively investigated by 2D and 3D numerical samples in [1] and maximum stability of the solution of the equation for the geometric equation and their existence has been investigated in [2]. The method (FSM) for solving the factored Eikonal equation of three evaluation criteria to evaluate the accuracy and convergence of inversion velocity in [3] has been discussed. The time-dependent Eikonal equation used in various sections. So According to the initial and boundary conditions, it is considered by the meshless methods in [4].

In this paper, we consider the system of Eikonal equations in the form below

$$\begin{cases} \mathcal{F}_{1}(u,v) = \alpha_{1}|\nabla(u+v)| + \beta_{1}|\nabla v| - f_{1}(x) = 0, & x \in \Omega, \\ \mathcal{F}_{2}(u,v) = \alpha_{2}|\nabla u| + \beta_{2}|\nabla(v-u)| - f_{2}(x) = 0, & x \in \Omega, \\ u(x) = g_{1}(x), v(x) = g_{2}(x) & x \in \Gamma \subset \Omega \end{cases}$$
(1)

where  $\Omega$  is a domain in  $\mathbb{R}^d$  and  $\Gamma$  is the subset of  $\Omega$  domain where the boundary condition is located. In (1),  $f_1(x), f_2(x)$ ,  $g_1(x), g_2(x), \alpha_1, \alpha_2, \beta_1, \beta_2$  are known and  $u(x), v(x) \in C^n(\Omega)$  are unknown functions. The space  $C^n(\Omega), k \ge 1$ , consists of functions which are n-times continuously differentiable over  $\Gamma$ .

<sup>\*</sup>Corresponding Author: Davood Rostamy: Department of Mathematics, Imam Khomeini International University, Qazvin, Iran; Email: rostamy@khayam.ut.ac.ir, rostamyd@yahoo.com Mehrollah Meher: Department of Mathematics, Imam Khomeini International University, Qazvin, Iran

Let us consider another form of (1) described by the following minimization problem:

$$\inf_{\substack{u,v \in C^n(\Omega)}} J(u,v)$$
(2)  
subject to  
$$\begin{bmatrix} u(x) = g_1(x), & x \in \Gamma, \\ v(x) = g_2(x), & x \in \Gamma, \end{bmatrix}$$

where  $J(u, v) := ||\mathcal{F}(u, v)||_2^2 = \mathcal{F}_1^2(u, v) + \mathcal{F}_2^2(u, v)$  is the "energy" functional. Here, we consider that  $\mathcal{F}(u, v) =$  $(\mathcal{F}_1(u,v),\mathcal{F}_2(u,v))^T$ .

When dealing with polynomials, we use multi-index notation. Let  $\alpha = (\alpha_1, ..., \alpha_d)$  with all  $\alpha_i \ge 0$  being integers; and let  $|\alpha| = \alpha_1 + \alpha_d$ . The monomial  $\mathbf{x}^{\alpha}$  is defined by

$$\mathbf{x}^{\alpha} = x_1^{\alpha_1} \cdots x_d^{\alpha_d}.$$

Therefore, we define

$$\Pi_n^d = \left\{ \Sigma_{|\alpha| \le n} b_\alpha \mathbf{x}^\alpha | b_\alpha \in \mathbb{R} \right\}.$$

These are the polynomials in d variables that are of degree ≤ n**.** 

Moreover, from (2), we consider the following minimization problem:

$$\inf_{P_1, P_2 \in \Pi_n^d} J(P_1, P_2)$$
(3)  
subject to  
$$\begin{bmatrix} P_1(x) = g_1(x), x \in \Gamma, \\ P_2(x) = g_2(x), x \in \Gamma. \end{bmatrix}$$

We claim that if  $||u - P_1||_{\infty} \longrightarrow 0$  and  $||v - P_2||_{\infty} \longrightarrow 0$ then (2) and (3) are equivalent.

We consider that in (3), the objective function is not smooth. The problem (3) is called non-smooth Eikonal optimization (NSEO) [5, 6]. The subgradient method is one of the simplest methods for solving partial differential equations used to minimize a nondifferentiable convex function. It is well known that the subgradient method is slower, but it can be more successful than existing methods (nonsmooth optimization) by solving changes and creating algorithms and combining some other methods to solve different problems. It is well known that the subgradient method is slower, but with useful modifications and algorithms, and by combining some other methods, it may be more successful in solving various problems than the existing methods for (nonsmooth optimization).

In this paper we want to use a combination of two methods (subgradient and differential quadrature) to solve the system of Eikonal equations. Because these methods are easy to implement and have a relatively low computational

cost. As a result, we can quickly find solutions and develop methods for nonsmooth problems. Different versions of the subgradient method for various problems have been developed and presented by researchers. A new method for solving (non-smooth convex optimization problem) using the sub-degree method has been introduced and its convergence has been proved. Also, The Proximal bundle method is compared with the proposed algorithm [7]. Subgradient methods are a powerful tool that can quickly find near-optimal dual solutions. The use of dual subgradient methods for convex optimization of a nonlinear multicommodity flow problem has been investigated by researchers in [8] and their convergence has been generalized. Algorithms have a direct impact on convergence speed, computational cost and response accuracy, so we need a useful and efficient algorithm for our problem. A new algorithm has been implemented by researchers to minimize non-convex and nonsmooth functions and has been compared with two versions of the subgradient method [9]. Subgradient methods are mostly used to solve nonsmooth convex optimization problems, such as bifurcation/ buckling analysis of beams, which is a nonsmooth problem. Also, the nonsmooth benchmark problems are acceptable by this method [10]. The combination of the subgradient method with the sloping method of a nonlinear operator on a problem similar to fused lasso has been analyzed as (fixed-point subgradient splitting method) and its convergence has been proved [11]. Researchers have extensively researched Riemannian subgradient methods to minimize weak convex functions on the Stiefel manifold [12]. Various solution methods for (mixed integer linear optimization) such as the subgradient optimization problem, the use of Lagrangian duals, for mixed-binary optimization, the recovery of primary solutions and cutting-planes have been extensively analyzed by researchers in [13]. A nonsmooth Equilibrium Problem is presented with a subgradient method in finite dimensional space so, an inexact subgradient algorithm and its convergence have been analyzed [14]. A dual subgradient method has been proposed using subgradient methods to solve convex optimization problems, which has been investigated with the average for (optimal resource allocation) in a multifactorial environment [15]. On the other hand, subgradient methods can be used to generate initial-dual solutions and develop solutions for nonsmooth convex optimization problems [16]. To solve bilevel equilibrium problems, a computational method using the subgradient method has been investigated and its convergence theorem has been proved, thus an algorithm with low computational cost has been introduced in [17]. For equilibrium problems and nonexpansive mappings, the subgradient method is used and an algorithm that uses a projection does not require the Lipschitz condition for bifunctions [18]. From the combination of subgradient methods, two strongly convergent algorithms have beenproposed, their convergence speed has been compared and its numerical results have been analyzed [19]. To solve the nonlinear split feasibility problems, an algorithm has been investigated by researchers using a combination of subgradient methods [20]. Recent advances have shown that the subgradient method is a powerful, useful, and efficient numerical method in science and other physical and engineering phenomena that plays a vital role in real life. Such as estimation in sensor networks, distributed control of multifactor systems, resource allocation, game theory, equilibrium problem, cooperative control of multiagent systems, equilibrium problem in network systems and machine learning [21–23]. Which has been used extensively by researchers. Therefore, a combination of subgradient methods for different problems is proposed and their algorithms are developed [24]. Different models of mathematics in science and engineering lead to the system of equations (1). This type of nonlinear system model can be seen in mechanics, materials science, engineering, and other sciences. Where one type of these models (Eikonal equation system) with Dirichlet boundary conditions for the 2-D problem is considered using the FEM in [25]. A numerical method based on elliptical solvents is presented in [26] and the type of special systems that arise in optimal control of a random evolutionary process has been studied in [27]. Consideration (density, velocity, energy), velocity and pressure disturbance, energy flux, wave equation, Acoustic wave are among the applications of hyperbolic systems that have been extensively researched in [28]. In model (1), suppose we have two *u* and *v* wave propagations and are moving in a fluid environment by fluid, Therefore,  $f_1$  and  $f_2$  show the wave propagation velocity, respectively. In many applications, waves have very high frequencies, are equivalents, or have very low wavelengths. To describe such waves, we need a distinction that is significantly smaller than a single wavelength. So how can the problem of having two waves propagating simultaneously at different speeds be calculated? Although there are analytical methods for solving the system of equations (1), using them is not easy. For example, optical phenomena. They are described by a wave equation. Assume that we use 10 points along the wave to describe each wave, so in this case, storing data is difficult [29]. In addition to the complexity of solving system equations (1), its application is also important, so we want to develop the Eikonal equation system.

In this paper, we implement the differential quadrature (DQ) method [30] for numerical solution of Equation System (1). Therefore, we consider the methods of Polyno-

mial differential quadrature (PDQ) and Fourier differential quadrature (FDQ) on the system of equations (1). It should be noted that in this method, the effect of grid point on the accuracy of DQM results and the stability of the issue plays a very important role, So the choice of grid points depends on the subject [31]. The DQM has been applied to various equation problems by researchers. In [32], using DQ method, the authors have been investigated for deviation, buckling and analysis of free vibration of beams and plates with different boundary conditions and to obtain the numerical solution of the nonlinear Schrödinger equation using the DQ method based on the modified cube B-spline for the ordinary differential equation system [33]. Based on modified cube B-spline for solving linear and nonlinear PDEs in higher dimensions has also been studied in [34]. In [35], the RBF-DQ method is used on a bunch of Lane-Emden type equations and for solving linear and nonlinear systems, the second order value problem is considered in [36] by LRBF-DQ method. Characteristics of time domain DQ method have been studied in [37]. For the two-dimensional and three-dimensional equation system, B-spline functions have been used to develop the new DQ method [38] and the chaotic Lorenz, Chen, Genesio and Rössler systems have been studied by the DQ method in [39].

In Section 2, we present the DQ method for the first and upper order derivatives with respect to x and y respectively to obtain the weight coefficients. Therefore, to solve the system (3), PDQ and FDQ methods and the use of Lagrange interpolation have been expressed as test functions to obtain weighting coefficients.

Also, in this section, we propose three types of grid points that are with uniform and non- uniform for convergence and accuracy of numerical results. We will also show that the choice of grid points depends on the type of issue.

In Section 3. On the other hand, we seek to solve (3) by using hybrid of sub-gradient idea and the PDQ or FDQ methods. Therefore, we give two class of algorithms. So, we give some numerical examples of (3) in 2-D and 3-D problems. To solve the numerical solution of the system of equations, we use the weighted coefficient of the first order derivatives of the proposed methods.

In Section 5, To measure the accuracy of our errors, we consider the  $L_1$ ,  $L_\infty$  norm and the root mean square error for different problems using PDQ and FDQ methods. Each of these issues can be a model of wave propagation equations, which is an important physical phenomenon that has been presented by the true solution of the system of equations or as a field of velocity. In this section, we have specifically considered five examples for two-dimensional and three-dimensional systems, and for each of the problems we have presented errors and order of convergence in the tables.

Also, to compare the accuracy of the errors, we have shown the error plots of the PDQ and FDQ methods in the figures and in Section 6, the results are summarized.

### 2 An overview of DQM

DQM is a numerical method suitable for numerical solution of ordinary and PDE because this method has high accuracy and little computational cost. One of the special advantages of the DQ method is its ease of use, so it has been implemented extensively for a variety of topics. The main idea of this method started from ordinary integral quadrature [31, 40]. The method of DQ was introduced by Bellman in 1972, Quan and Chang in 1989, and then the general approach was presented by Shu in 1991 [31]. Now we consider the approximations of the derivatives of the first order for the 2D problem as follows:

$$u_{x}^{(1)}\left(x_{i}, y_{j}\right) = \sum_{k=1}^{N_{x}} Q_{ik}^{(1)} u\left(x_{k}, y_{j}\right), \quad i = 1, 2, ..., N_{x}, \quad (4)$$

$$u_{y}^{(1)}\left(x_{i}, y_{j}\right) = \sum_{k=1}^{N_{y}} Q_{jk}^{(1)} u\left(x_{i}, y_{k}\right), \quad j = 1, 2, ..., N_{y}.$$
 (5)

Where  $u_x^{(1)}$ ,  $u_y^{(1)}$  the first-order derivatives,  $Q_{ik}^{(1)}$  and  $Q_{jk}^{(1)}$  are the weighting coefficients of the derivatives of the u(x, y)function with respect to x and y respectively. Also, we consider that  $N_x$  and  $N_y$  are the number of grid points in the xand y directions at the domain  $a = x_1, x_2, ..., x_N = b$ . Similarly, for high-order derivatives, we can write as follows:

$$u_{x}^{(n)}\left(x_{i}, y_{j}\right) = \sum_{k=1}^{N_{x}} G_{ik}^{(n)} u\left(x_{k}, y_{j}\right),$$
(6)

$$u_{y}^{(m)}(x_{i}, y_{j}) = \sum_{k=1}^{N_{y}} G_{jk}^{(m)} u(x_{i}, y_{k}), \qquad (7)$$

For  $i = 1, 2, ..., N_x$ , n = 1, 2, ..., N - 1,  $j = 1, 2, ..., N_y$ , or m = 1, 2, ..., M - 1 [34, 35, 41].

## 2.1 Polynomial differential quadrature (PDQ) method

Now, to calculate the weighting coefficients, we use the Lagrange interpolated polynomials as basic functions [30, 40], so we can write,

$$Q_k(x) = \frac{Q(x)}{(x - x_k)Q^{(1)}(x_k)}, \quad k = 1, 2, ..., N$$
(8)

Where

$$Q(x) = (x - x_1)(x - x_2)...(x - x_N)$$
(9)

$$Q^{(1)}(x_i) = \prod_{\substack{k=1\\k\neq i}}^N (x_i - x_k).$$
 (10)

Using Equation (8), the weighting coefficients of the first order derivatives can be obtained as follows:

$$\begin{cases} Q_{ij}^{(1)} = \frac{Q^{(1)}(x_i)}{(x_i - x_j)Q^{(1)}(x_j)} & \text{for } i, j = 1, 2, ..., N, \quad j \neq i, \\ Q_{ii}^{(1)} = -\sum_{\substack{j=1\\i\neq j}}^{N} Q_{ij}^{(1)}, \quad i = 1, 2, ..., N, \end{cases}$$
(11)

And for the weighted coefficients of the second order derivatives, we can write as follows,

$$\begin{cases} Q_{ij}^{(2)} = 2Q_{ij}^{(1)} \left\lfloor Q_{ii}^{(1)} - \frac{1}{x_i - x_j} \right\rfloor, & \text{for } i, j = 1, 2, ..., N, \\ & j \neq i, \\ Q_{ii}^{(2)} = -\sum_{\substack{j=1\\i \neq j}}^{N} Q_{ij}^{(1)}, & i = 1, 2, ..., N. \end{cases}$$
(12)

Here  $x_i$  are the coordinates of the grid points and for i = 1, 2, ..., N. Similarly, for high-order derivatives, you can refer to [31] for more information.

#### 2.2 Superior selection from grid points

In the DQ method, grid points are a key element in the numerical solution of differential equations because grid points affect the convergence and accuracy of numerical results. The choice of grid points depends on the type of problem [31, 41]. It should be noted that the DQ method has been tested for various partial differential equation problems that the non-uniform grid points are better than the uniform grid points, and have a high accuracy answer. So, we focus on the following three types of grid points:

$$x_i = a + (b - a) \frac{(i - 1)}{N - 1}$$
  $i = 1, 2, ..., N,$  (13)

$$x_{i} = a + \frac{(b-a)}{2} \left( 1 - \cos\left(\frac{2i-1}{N-1}\pi\right) \right), \quad (14)$$
  
$$i = 1, 2, \dots, N,$$

or

$$\begin{aligned} x_{i} &= a + \frac{(b-a)}{2} \left( \frac{\cos[\frac{(i-1)\pi}{N-1}]}{\cos[\frac{\pi}{2N}]} \right. \end{aligned} \tag{15} \\ &+ 1 - \frac{\cos[\frac{(2i-3)\pi}{2(N-1)}]}{\cos[\frac{\pi}{2(N-1)}]} + 1 \right), \quad i = 1, 2, ..., N. \end{aligned}$$

- 439

Grid points (13) grid points are uniform (equally spaced points), grid points (14) are non-uniform grid points (unequally spaced points) that are called Chebyshev-Gauss-Lobatto points and are usually used in most articles [30, 38, 41, 43]. We are inspired by the difference between the two Chebyshev-Gauss-Lobatto in [42] and define (15). The grid points (15) are non-uniform. Here N indicates the number of grid points in the directions of x, y and z in the domain [a, b].

Therefore, by putting (11) and (12) in (3) with one class of grid points (13) or (14) or (15), we have a minimization problem. Our goal in this article is to solve the optimization problem by DQ. On the other hand, we consider the following the differential quadrature method for combining with (3).

## 2.3 Fourier differential quadrature (FDQ) method

In the FDQ method, we use the Fourier series expansion to obtain weighting coefficients, which means that the function is approximated by the Fourier series expansion [30, 41, 43]. We presented the Fourier series expansion as follows,

$$f(x) = b_0 + \sum_{k=1}^{N/2} \left( c_k \cos \varphi + d_k \sin \varphi \right).$$
(16)

In (16) we consider the period of  $\varphi = \frac{k\pi x}{L}$  and L represents the length of the interval (physical domain). Using Lagrange interpolated trigonometric polynomials, the weighting coefficients of the first and second-order derivatives can be obtained as follows,

$$\begin{cases} Q_{ij}^{(1)} = \frac{\pi_{2L}g(x_i)}{g(x_i)\sin\frac{(x_i-x_j)}{2L}\pi} & \text{for } j \neq i, \quad i, j = 1, 2, ..., N, \\ Q_{ii}^{(1)} = -\sum_{\substack{j=1\\i\neq j}}^{N} Q_{ij}^{(1)}. \end{cases}$$
(17)

$$\begin{cases} Q_{ij}^{(2)} = Q_{ij}^{(1)} \left[ 2Q_{ii}^{(1)} - \frac{\pi}{L} \cot \frac{(x_i - x_j)}{2L} \pi \right] & \text{for } j \neq i, \\ i, j = 1, 2, ..., N, \quad (18) \\ Q_{ii}^{(2)} = -\sum_{\substack{j=1\\i\neq j}}^{N} Q_{ij}^{(2)}, \end{cases}$$

where

$$g(x_i) = \prod_{\substack{j=0\\i\neq j}}^N \sin \frac{(x_i - x_j)}{2L} \pi.$$
 (19)

See [31] for more information. Therefore, by putting (17) and (18) in (3) with one class of grid points (13) or (14) or (15), we have a minimization problem with FDQ. Our goal in this article is to solve the optimization problem by FDQ.

**DE GRUYTER** 

## 3 Implementation of the methods based on sub-Gradients [5], [6]

In this section, we implement FDQ and PDQ methods on the system (3). Let  $A = |\nabla u|$ ,  $B = |\nabla v|$ , so,  $\alpha_1\beta_2 - \alpha_2\beta_1 \neq 0$ , We discrete the nonlinear system for  $(x, y) \in \Omega \subset R^2$ , where  $\alpha_1(x_i, y_j)$ ,  $\alpha_2(x_i, y_j)$ ,  $\beta_1(x_i, y_j)$ ,  $\beta_2(x_i, y_j)$  and the functions  $f_1(x_i, y_j)$ ,  $f_2(x_i, y_j)$  are determined from (1). it can be written as follows,

$$\begin{cases} A = \sqrt{\left(\sum_{k=1}^{N_x} Q_{i,k}^{(1)} u_{k,j}\right)^2 + \left(\sum_{k=1}^{N_y} \overline{Q}_{j,k}^{(1)} u_{i,k}\right)^2}, \\ B = \sqrt{\left(\sum_{k=1}^{N_x} Q_{i,k}^{(1)} v_{k,j}\right)^2 + \left(\sum_{k=1}^{N_y} \overline{Q}_{j,k}^{(1)} v_{i,k}\right)^2}. \end{cases}$$
(20)

Similarly, for  $(x, y, z) \in \Omega \subset R^3$ , where  $\alpha_1(x_i, y_j, z_k)$ ,  $\alpha_2(x_i, y_j, z_k)$ ,  $\beta_1(x_i, y_j, z_k)$ ,  $\beta_2(x_i, y_j, z_k)$  and the functions  $f_1(x_i, y_j, z_k)$ ,  $f_2(x_i, y_j, z_k)$ . We present the discrete form A and B as the following:

$$\begin{cases} A = \left( \left( \sum_{k=1}^{N_{x}} Q_{i,k}^{(1)} u_{k,j,\kappa} \right)^{2} + \left( \sum_{k=1}^{N_{y}} \overline{Q}_{j,k}^{(1)} u_{i,k,\kappa} \right)^{2} + \left( \sum_{k=1}^{N_{z}} \widetilde{Q}_{k,k}^{(1)} u_{i,j,k} \right)^{2} \right)^{1/2}, \\ B = \left( \left( \sum_{k=1}^{N_{x}} Q_{i,k}^{(1)} v_{k,j,\kappa} \right)^{2} + \left( \sum_{k=1}^{N_{y}} \overline{Q}_{j,k}^{(1)} v_{i,k,\kappa} \right)^{2} + \left( \sum_{k=1}^{N_{z}} \widetilde{Q}_{k,k}^{(1)} v_{i,j,k} \right)^{2} \right)^{1/2}. \end{cases}$$

$$(21)$$

Therefore, in system of (20) and (21),  $Q_{i,k}^{(1)}$ ,  $\overline{Q}_{j,k}^{(1)}$ ,  $\widetilde{Q}_{\kappa,k}^{(1)}$  are the weighting coefficients of the first order derivatives, with respect to x ,y and z respectively. The weighting coefficients in the PDQ method are calculated using (11) and in the FDQ method are obtained from (17) and (19). Where  $N_x$ ,  $N_y$  and  $N_z$  are the number of grid points in the x, y and z directions at the domain [a, b].

#### 3.1 Algorithms of sub-Gradient for PDQ or FDQ at grid points

The subgradient (related to subderivative and subdifferential) of a convex functional is a idea of generalizing or approximating the derivative of a convex functional at nondifferentiable points. The definition of a subgradient is as follows: g(u, v) is a subgradient of J(u, v) at (u,v) if, for all (U,V), the following is true:

$$J(U, V) \geq J(u, v) + g(U - u, V - v).$$

We know that for minimizing J the subgradient method uses the following iteration:

$$u^{(k+1)} = u^{(k)} - \alpha_k g_1^{(k)}$$

and

$$v^{(k+1)} = v^{(k)} - \alpha_k g_2^{(k)}$$

Where k is the number of iterations,  $u^{(k)}$  and  $u^{(k)}$  are the kth iterate,  $g^{(k)} = (g_1^{(k)}, g_2^{(k)})^T$  is any subgradient at  $(u^{(k)}, v^{(k)})$ and  $\alpha_k (> 0)$  is the kth step size. Thus, at each iteration of the subgradient method, we take a step in the direction of a negative subgradient. As explained above, when f is differentiable,  $g^{(k)}$  simply reduces to  $\nabla I(u^{(k)}, v^{(k)})$ . It is also important to note that the subgradient method is not a descent method in that the new iterate is not always the best iterate. Thus we need some way to keep track of the best solution found so far. i.e. the one with the smallest function value. We can do this by, after each step, setting

$$J_{\text{best}}^{(k)} = \min\{J_{\text{best}}^{(k-1)}, J(u^{(k)}, v^{(k)})\}$$

and setting  $i_{\text{best}}^{(k)} = k$  if  $x^{(k)}$  is the best (smallest) point found so far. Thus we have:

$$J_{\text{best}}^{(k)} = \min\{J(u^{(1)}, v^{(1)}), J(u^{(2)}, v^{(2)}), \dots, J(u^{(k)}, v^{(k)})\}$$

which gives the best objective value found in k iterations. Since this value is decreasing, it has a limit (which can be -∞).

Two class of algorithms are provided below for the sub-Gradient method:

#### Algorithm of PDQ by Sub-gradient

- Step 1 By putting (11) and (12) in (3) with one class of grid points (13) or (14) or (15), we define J, the step size of  $\alpha$ , sub-Gradient g and initial ( $u^{(0)}, v^{(0)}$ ).
- **Step 2** Iterate along  $u^{(k+1)} = u^{(k)} \alpha_k g_1^{(k)}$  and  $v^{(k+1)} =$ Step 2 – Relate using I –  $\kappa_1$   $v^{(k)} - \alpha_k g_2^{(k)}$ .. Step 3 – Set  $J_{\text{best}}^{(k)} = \min\{J_{\text{best}}^{(k-1)}, J(u^{(k)}, v^{(k)})\}.$ Step 4 – Check if  $\lim_{k\to\infty} ||J_{\text{best}}^{(k)} - J^{(k-1)}|| < \epsilon$  if not return to
- step 2.
- Step 5 If the condition of step 4 is satisfied optimal solution is found.

#### Algorithm of FDQ by Sub-gradient

- Step 1 By by putting (17) and (18) in (3) with one class of grid points (13) or (14) or (15), we define J, the step size of  $\alpha$ , sub-Gradient g and initial ( $u^{(0)}, v^{(0)}$ ).
- **Step 2** Iterate along  $u^{(k+1)} = u^{(k)} \alpha_k g_1^{(k)}$  and  $v^{(k+1)} = v^{(k)} \alpha_k g_2^{(k)}$ .

 $\begin{array}{l} \textbf{Step 3-} & \text{Set } J_{\text{best}}^{(k)} = \min\{J_{\text{best}}^{(k-1)}, J(u^{(k)}, v^{(k)})\}.\\ \textbf{Step 4-} & \text{Check if } \lim_{k \to \infty} \|J_{\text{best}}^{(k)} - J^{(k-1)}\| < \epsilon \text{ if not return to} \end{array}$ sten 2.

Step 5 – If the condition of step 4 is satisfied optimal solution is found.

### 4 Numerical results

In this study, we implement numerical algorithms of FDQ and PDQ by sub-gradient on the Eikonal equation system and consider five examples to develop the equation system. Our goal is to solve the system (1) and obtain its errors, Therefore, we use the following three criteria to measure accuracy and convergence:

$$L_{2} = \sqrt{\sum_{i=1}^{N} \left| u_{i}^{(exact)} - u_{i}^{(app)} \right|^{2}},$$
 (22)

$$L_{\infty} = \max_{i=1}^{N} \left| u_i^{(exact)} - u_i^{(app)} \right|, \tag{23}$$

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left| u_i^{(exact)} - u_i^{(app)} \right|^2}.$$
 (24)

Where *N* is the number of grid points and  $u_i^{(exact)}$  and  $u_i^{(app)}$ show the exact solution and approximate solution respectively. The order of convergence is calculated by the following formula,

$$Order = \frac{1}{\log 2} \log \frac{E(\frac{1}{2}N_2)}{E(N_1)}.$$
 (25)

Where *E* is  $L_2$ ,  $L_{\infty}$  and (RMS) errors obtained for  $N_1$  and  $N_2$  and  $N_1$ ,  $N_2$  indicate the number of grid points.

**Remark 1.** For investigating of stability of methods, we know that a small perturbation in initial data may generate a large amount of perturbations in the solution. Therefore, we apply noisy data to show the stability of the proposed methods:

$$\tilde{Q}_{i,j} = Q_{i,j}(1 + \varepsilon.rand(ij)), \quad i, j = 1, 2, ..., N$$

where  $Q_{i,j}$  is the exact data and (*ij*) rand is a random number with uniform distribution of the interval [-1, 1]. The magnitude  $\varepsilon$  displays the noise level of the measurement data.

**Example 1.** We consider the two-dimensional problem of the equation system (1) in the domain  $\Omega = [-1, 1] \times [-1, 1]$ . Where  $\alpha_1 = x^2 y$ ,  $\alpha_2 = xy^2$ ,  $\beta_1 = \sin(x, y)$ ,  $\beta_2 = \cos(x, y)$ ,  $\Gamma$ 

order are presented in Table 1. Error plots of FDQ and PDQ algorithms are shown in Figure 1.

			DQ	PDQ					
	Ν	$L_2$ – error	Order	$L_{\infty}$ – error	Order	$L_2$ – error	Order	$L_{\infty}$ – error	Order
u(x, y)	16	3.68 <i>E</i> – 4		5.63 <i>E</i> – 5		8.27 <i>E</i> – 8		1.30 <i>E</i> – 8	
	20	1.40E - 5	4.7	1.68E - 6	5.1	7.32E - 11	10.1	9.08 <i>E</i> – 12	10.5
	22	2.72E - 6	2.4	2.90 <i>E</i> – 7	2.5	1.85 <i>E</i> – 12	5.3	2 <b>.</b> 26 <i>E</i> – 13	5.3
	28	1.97 <i>E</i> – 8	7.1	1.56 <i>E</i> – 9	7.5	1.03 <i>E</i> – 13	4.2	1 <b>.</b> 95 <i>E</i> – 14	3.5
v(x, y)	16	5.51 <i>E</i> – 6		1.03E - 6		1.39 <i>E</i> – 8		1.91 <i>E</i> – 9	
	20	2.02E - 7	4.8	3.34 <i>E</i> – 8	4.9	1.21E - 11	10.2	1.32 <i>E</i> – 12	10.5
	22	3.86 <i>E</i> – 8	2.9	5.86 <i>E</i> – 9	2.5	3.22 <i>E</i> – 13	5.2	3.69 <i>E</i> – 14	5.2
	28	2.75E - 10	7.1	3.65E - 11	7.3	5.03E - 14	2.9	1.50E - 14	1.3

Table 1: Numerical results and Convergence Order For example 1.



**Figure 1:** Comparison of errors( $L_2$ ,  $L_\infty$ ) and convergence for example 1.

**Example 2.** Consider Eq. (1) with  $\Omega = [-1, 1] \times [-1, 1]$ where  $\alpha_1 = y \sin(x)$ ,  $\alpha_2 = y \cos(x)$ ,  $\beta_1 = x \sin(y)$ ,  $\beta_2 = x \cos(y)$  and the true solution is  $u(x, y) = e^{\sin(x) \sin(y)}$ ,  $v(x, y) = e^{\cos(x) \cos(y)}$ . In this example, we use the grid points (3.3) in algorithms of FDQ and PDQ. Also, errors  $L_2$ ,  $L_{\infty}$  and convergence of the methods are shown in Table 2, and for the convergence accuracy, we have also presented the error plots in Figure 2. In this example,  $\Gamma$  has three separate points,  $\Gamma$  is the points  $\Gamma = \left\{ \left(\frac{1}{4}, \frac{1}{4}\right), \left(\frac{1}{2}, \frac{1}{2}\right), \left(\frac{2}{5}, \frac{2}{5}\right) \right\}$  and  $g_1\left(\frac{1}{4}, \frac{1}{4}\right) = g_1\left(\frac{1}{2}, \frac{1}{2}\right) = g_1\left(\frac{2}{5}, \frac{2}{5}\right) = 1, g_2\left(\frac{1}{4}, \frac{1}{4}\right) = g_2\left(\frac{1}{2}, \frac{1}{2}\right) = g_2\left(\frac{2}{5}, \frac{2}{5}\right) = 2.718.$ 

			DQ						
	Ν	$L_2$ – error	Order	$L_{\infty}$ – error	Order	$L_2 - error$	Order	$L_{\infty}$ – error	Order
u(x, y)	6	6.01 <i>E</i> – 2		2.21 <i>E</i> – 2		4.42 <i>E</i> – 2		2.12 <i>E</i> – 2	
	8	7.08E - 8	19.7	2.03E - 8	20.1	4.65 <i>E</i> – 5	9.9	1.34 <i>E</i> – 5	10.6
	12	2.47E - 10	8.2	2.11E - 10	6.6	1.72E - 10	18.0	1.63 <i>E</i> – 13	16.3
	18	3.12 <i>E</i> – 15	16.3	3.11 <i>E</i> – 15	16.1	2.66 <i>E</i> – 15	16.0	2.66 <i>E</i> – 14	15.9
v(x, y)	6	1.38 <i>E</i> – 4		7.55 <i>E</i> – 5		1.51 <i>E</i> – 4		8.92 <i>E</i> – 5	
	8	2.02E - 8	12.7	1.07E - 8	12.9	1.36 <i>E</i> – 5	3.5	7.25E - 6	3.6
	12	2.15E - 10	6.6	1.74E - 10	5.9	1.41E - 10	16.6	1.38E - 10	15.7
	18	6.22 <i>E</i> – 15	15.1	6.22 <i>E</i> – 15	14.8	7.33 <i>E</i> – 15	14.2	7.33 <i>E</i> – 15	14.2

Table 2: Numerical results and Convergence Order For example 2.



**Figure 2:** Comparison of errors  $(L_2, L_\infty)$  and convergence for example 2.

racy, we use grid points (3.1). where the exact solution is  $u(x, y) = 0.4x \cos\left(\frac{y}{3}\right), v(x, y) = 0.9 \sin\left(\frac{x}{3}\right) \sin\left(\frac{y}{3}\right)$ . Error plots for *u*, *v* functions are shown in Figure 3. Errors ( $L_{\infty}$ , *RMS*) and convergent order are reported in Table 3.

			DQ	PDQ					
	Ν	RMS – error	Order	$L_{\infty}$ – error	Order	RMS – error	Order	$L_{\infty}$ – error	Order
u(x,y)	8	1.36 <i>E</i> – 2		9.22 <i>E</i> – 3		1.02E - 6		7.01 <i>E</i> – 7	
	10	8.98 <i>E</i> – 3	0.6	5.55E - 3	0.7	9.41 <i>E</i> – 8	3.4	4 <b>.</b> 53 <i>E</i> – 8	4.0
	12	5.46 <i>E</i> – 3	0.7	3.21 <i>E</i> – 3	0.8	1.04E - 11	13.1	5.80 <i>E</i> – 12	12.9
	14	3.51 <i>E</i> – 3	0.7	1.86 <i>E</i> – 3	0.8	7.67 <i>E</i> – 13	3.8	2.83 <i>E</i> – 13	4.4
v(x, y)	8	3.07 <i>E</i> – 3		2.09 <i>E</i> – 3		7.65 <i>E</i> – 9		5.21 <i>E</i> – 9	
	10	8.01E - 4	1.9	4 <b>.</b> 93 <i>E</i> – 4	2.1	2.24 <i>E</i> – 9	1.8	1.38 <i>E</i> – 9	1.9
	12	1.99 <i>E</i> – 4	2.0	1.12E - 4	2.1	1.34E - 12	10.7	7.76E - 13	10.8
	14	5.46 <i>E</i> – 5	1.8	2 <b>.</b> 97 <i>E</i> – 5	1.9	1.65E - 14	6.3	1.42E - 14	5.8

Table 3: Numerical results and Convergence Order For example 3.



**Figure 3:** Comparison of errors  $(L_{\infty}, RMS)$  and convergence for example 3.

**Example 4.** In this example, let  $\alpha_1 = \frac{1}{5}$ ,  $\alpha_2 = \frac{1}{4}$ ,  $\beta_1 = \frac{1}{3}$ ,  $\beta_2 = \frac{1}{7}$  be constant. The computational domain is  $\Omega = [-1, 1] \times [-1, 1]$  and the exact solution of the equation system (1) is

$$u(x, y) = \sqrt{(2 - x)^2 + (2 - y)^2} + \sin(x^2 + y^2),$$
  
$$v(x, y) = \sin(2\pi x)\sin(2\pi y) + \cos(x^2 + y^2).$$

Table 4: Numerical results and Convergence Order For example 4.

Where  $\Gamma$  is the point (0, 0). We implement FDQ and PDQ algorithms on the equation system. For high accuracy, we use the grid points (14). The convergence of the methods is compared in Figure 4. The numerical results of the errors and the convergence order are considered in Table 4. The two-dimensional and three-dimensional numerical solutions are shown in Figure 6.

		FDQ				PDQ			
	Ν	$L_2$ – error	Order	$L_{\infty}$ – error	Order	$L_2$ – error	Order	$L_{\infty}$ – error	Order
u(x, y)	14	1.11 <i>E</i> – 3		2.85 <i>E</i> – 4		1.48E - 6		4.34 <i>E</i> – 7	
	16	3.42 <i>E</i> – 4	1.7	1.11E - 4	1.4	5.69 <i>E</i> – 8	4.7	1.98E - 8	4.5
	22	8.42E - 6	5.3	1.65E - 6	6.1	1.11E - 12	15.6	1.91 <i>E</i> – 13	16.7
	30	3.08 <i>E</i> – 7	4.8	4.91 <i>E</i> – 8	5.1	1.84 <i>E</i> – 13	2.6	3.29 <i>E –</i> 14	2.5
v(x, y)	14	4 <b>.</b> 25 <i>E</i> – 3		1.48 <i>E</i> – 3		2.03 <i>E</i> – 2		6.40 <i>E</i> – 3	
	16	3 <b>.</b> 99 <i>E</i> – 4	3.4	1.23E - 4	3.6	1.79 <i>E</i> – 4	6.8	3.80 <i>E</i> – 5	7.4
	22	3.45 <i>E</i> – 4	0.2	9.11 <i>E</i> – 5	0.4	8.47 <i>E</i> – 9	14.4	2.52 <i>E</i> – 9	14.0
	30	3.60 <i>E</i> – 7	9.9	6.69 <i>E</i> – 8	10.4	4.84 <i>E</i> – 13	14.1	7.05E - 14	15.1



**Figure 4:** Comparison of errors  $(L_2, L_{\infty})$  and convergence for example 4.

**Example 5.** In this example, we consider the threedimensional problem of system (1). Where  $\alpha_1 = 6$ ,  $\alpha_2 = 2$ ,  $\beta_1 = 2$ ,  $\beta_2 = 4$ .  $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1]$ .  $\Gamma$  is the point (0, 0, 0) with the true solution  $u(x, y, z) = zy \cos(\frac{x}{3})$ ,  $v(x, y, z) = zx \sin\left(\frac{y}{3}\right)$ . We use both methods in the system of equations and using grid points (14). Errors accuracy (*RMS*,  $L_{\infty}$ ) and convergence order are given in Table 5. Error plots are shown in Figure 5.

		FDQ				PDQ				
	Ν	RMS – error	Order	$L_{\infty}$ – error	Order	RMS – error	Order	$L_{\infty}$ – error	Order	
u(x, y)	4	1.42 <i>E</i> – 1		5.53 <i>E</i> – 2		9.51 <i>E</i> – 5		4.43 <i>E</i> – 5		
	6	5.56E - 2	1.3	1.71E - 2	1.7	1.67E - 7	9.2	8.26E - 8	9.1	
	8	8.56 <i>E</i> – 3	2.7	2.03 <i>E</i> – 3	3.1	7.45E - 10	7.8	1.66E - 10	9.0	
	10	1.83 <i>E</i> – 3	2.3	3.52 <i>E</i> – 4	2.5	6.43 <i>E</i> – 15	16.8	2.61 <i>E</i> – 15	16.0	
v(x, y)	4	6.95 <i>E</i> – 2		3 <b>.</b> 15 <i>E</i> – 2		1.78 <i>E</i> – 5		9 <b>.</b> 18 <i>E</i> – 6		
	6	1.37E - 2	2.3	4.68 <i>E</i> – 3	2.7	1.33E - 8	10.4	6.99 <i>E</i> – 9	10.4	
	8	2 <b>.</b> 72 <i>E</i> – 3	2.3	7.27E - 4	2.7	3.24 <i>E</i> – 11	8.7	1.30E - 11	9.1	
	10	5.35 <i>E</i> – 4	2.3	1.22 <i>E</i> – 4	2.6	1.59 <i>E</i> – 15	14.3	5.76 <i>E</i> – 16	14.5	

Table 5: Numerical results and Convergence Order For example 5.



**Figure 5:** Comparison of errors (*RMS*,  $L_{\infty}$ ) and convergence for example 5.



Figure 6: 2-D and 3-D contour plot numerical solution for example 4.

## **5** Conclusion

In this paper, we propose the hybrid of sub-gradient ideas with PDQ and FDQ methods for solving the system (1) and developing the Eikonal equation system. We have used three grid points (uniform and non-uniform grid) on different issues for the accuracy and convergence of the equation system (1). In addition, we have shown that the choice of grid points to reduce errors, accuracy and convergence depends on the type of problem. As you can see, we implemented the new grid points in the second example. We have shown that the scheme of new grid points is also effective and useful for solving the system of nonlinear equations. In this study, we found that to solve the equation system (1), the FDQ method requires less computational time than the PDQ method, and the PDQ method has a high accuracy compared to the FDQ method and reduces errors sufficiently. As you can see, we have shown the norm of errors, the root mean square error, order of convergence, and the accuracy of the methods presented in the tables, and we have also obtained error plots. As a result, we have achieved the convergence of problems and acceptable numerical results. The items to be researched in this article are as follows:

1. One of the biggest challenges in sub-gradient optimization is stage size selection. In this article, I state that our numerical results are a trial and error and finding the best step size. Because we need to study convergence theory to describe it accurately, which will be explored in future articles.

- 2. This method requires a modified pattern for nonconvex problems. Which we will discuss in future work.
- 3. Error analysis of this method and its limits should be proved using functional analysis.
- 4. Applied benchmarks in vehicle design and solid mechanics can create a new perspective on the usefulness of this method.
- 5. We observe that the results of this method for stiff systems are not suitable and we need to modified it.

**Acknowledgement:** The authors would like to thank the referees for some comments which have led to an improvement of the paper.

**Funding information:** The authors state no funding involved.

**Author contributions:** All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

**Conflict of interest:** The authors state no conflict of interest.

### References

- Li W, Qian J. Newton-type Gauss-Seidel Lax-Friedrichs highorder fast sweeping methods for solving generalized eikonal equations at large-scale discretization. Comput Math with Appl. 2019;79(4):1122–1239.
- [2] Cruz-Sampedro J, Skibsted E. Global solutions to the eikonal equation. J Differ Equ. 2013;255(12):4337–77.
- [3] Huang G, Hu Q, Luo S, Li H, Zhang H, Nobes DC. 2-D fast sweeping method for the factored Eikonal equation and its improvement on inversion accuracy. J Appl Geophys. 2019;166:68–76.
- [4] Dehghan M, Salehi R. A boundary-only meshless method for numerical solution of the Eikonal equation. Comput Mech. 2011;47(3):283–94.
- [5] Boyd S, Xiao L, Mutapcic A. Subgradient Methods, lecture notes of EE3920 Stanford University. 2003.
- [6] Nedic A, Bertsekas DP. Incremental subgradient methods for nondifferentiable optimization. SIAM J Optim. 2001;12(1):109– 38.
- [7] Bagirov AM, Jin L, Karmitsa N, Al Nuaimat N, Sultanova N. Subgradient method for nonconvex nonsmooth optimization. J Optim Theory Appl. 2012;157(2):416–35.
- [8] Gustavsson E, Patriksson M, Strömberg AB. Primal convergence from dual subgradient methods for convex optimization. Math Program. 2015;150(2):365–90.
- [9] Bagirov A, Ganjehlou AN. An approximate subgradient algorithm for unconstrained nonsmooth, nonconvex optimization. Math Methods Oper Res. 2008;67(2):187–206.
- [10] Niutta CB, Wehrle EJ, Duddeck F, Belingardi G. Surrogate modeling in design optimization of structures with discontinuous responses. Struct Multidiscipl Optim. 2018;57(5):1857–69.
- [11] Nimana N. A Fixed-Point Subgradient Splitting Method for Solving Constrained Convex Optimization Problems. Symmetry (Basel). 2020;377(12):2–16.
- [12] Li X, Chen S, Deng Z, Qu Q, Zhu Z, Man-Cho So A. Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. SIAM J Optim. 2021;31(3):1605–34.
- [13] Strömberg AB, Larsson T, Patriksson M. Mixed-Integer Linear Optimization: Primal-Dual Relations and Dual Sub-gradient and Cutting-Plane Methods. In: Bagirov A, Gaudioso M, Karmitsa N, Mäkelä M, Taheri S, editors. Numerical Non-Smooth Optimization. Cham: Springer; 2020 p. 499–547.
- [14] Santos P, Scheimberg S. An inexact subgradient algorithm for equilibrium problems. Comput Appl Math. 2011;30:91–107.
- [15] Nesterov Y, Shikhman V. Dual subgradient method with averaging for optimal source allocation. Eur J Oper Res. 2018;270(3):907– 16.
- [16] Nesterov Y, Shikhman V. Quasi-monotone subgradient methods for nonsmooth convex minimization. J Optim Theory Appl. 2015;165(3):917–40.
- [17] Thuy LQ, Hai TN. A projected subgradient algorithm for bilevel equilibrium problems and applications, J. Optim. Theory Appl. 175 (2017) 411–431.

- [18] P. N. Anh, L.D. Muu, A hybrid subgradient algorithm for nonexpansive mapping and equilibrium problems. Optim Lett. 2014;8:727– 38.
- [19] Gebrie AG, Wangkeeree R. Hybrid projected subgradientproximal algorithms for solving split equilibrium problems and split common fixed-point problems of nonexpansive mappings in Hilbert spaces. J Fixed Point Theory Appl. 2018;5:2–28.
- [20] Yen LH, Huyen NT, Muu LD. A subgradient algorithmfor a class of nonlinear split feasibility problems: application to jointly constrained Nash equilibrium models. J Glob Optim. 2019;73(4):849– 58.
- [21] Liu C, Li H, Shi Y. A unitary distributed subgradient method for multi-agent optimization with different coupling sources. Automatica. 2020;114(1):108834.
- [22] Mateos-Nunez D, Cortes J. Distributed saddle-point subgradient algorithm with Laplacian averaging. IEEE Trans Automat Contr. 2017;62(6):2720–35.
- Yen LH, Muu LD. A subgradient method for equilibrium problems involving quasiconvex bifunctions. Oper Res Lett. 2020;48(20):1– 6.
- [24] Ghali AE, Moudden ME. Reduced subgradient bundle method for linearly constrained non-smooth non-convex problems. Optimization. 2020;70(3).
- [25] Dacorogna B, Glowinski R, Pan TW, Numerical methods for the solution of a system of Eikonal equations with Dirichlet boundary conditions. C R Acad Sci Paris. 2003;I(336):511–518.
- [26] Caboussat A, Glowinski R, Pan TW. On the Numerical Solution of Some Eikonal Equations: An Elliptic Solver Approach. Chin Ann Math Ser. B. 2015;36:689–702.
- [27] Camilli F, Loreti P. Comparison results for a class of weakly coupled systems of eikonal equations. Hokkaido Math J. 2008;37(2):349-62.
- [28] Jin S, Liu H, Osher S, Tsai R. computing multi-valued physical observables for the high frequency limit of symmetric hyperbolic systems. J Comput Phys. 2005;210(2):497–518.
- [29] Pinchover Y, Rubinstein AJ. An Introduction Partial Differential Equation. 1st ed. New York: Cambridge University Press; 2005. p. 15–57.
- [30] Shu C. Approximation of differential quadrature method to simulate natural convection in a concentric annulus. Int J Numer Methods Fluids. 1999;30(8):977–93.
- [31] Shu C. Differential quadrature and its application in engineering. Springer-Verlag London Ltd., Great Britain, 2000. p. 29–128. https://doi.org/10.1007/978-1-4471-0407-0.
- [32] Wang X, Bert CW, Striz AG. Differential quadrature analysis of deflection, buckling, and free vibration of beams and rectangular plates. Comput Struc. 1993;48(3):473–9.
- [33] Bashan A, Yagmurlu NM, Ucar Y, Esen A. An effective approach to numerical soliton solutions for the Schrödinger equation via modified cubic B-spline differential quadrature method. Chaos Solit Fractals. 2017;100:45–56.
- [34] Mittal RC, Dahiya S. Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method. Appl Math Comput. 2017;313:442–52.
- [35] Parand K, Hashemi S. RBF-DQ method for solving non-linear differential equations of Lane-Emden type. Ain Shams Eng J. 2016;9(4):615–629.
- [36] Dehghan M, Nikpour A. Numerical solution of the system of second-order boundary value problems using the local radial basis functions based differential quadrature collocation method.

Appl Math Model. 2013;37(18-19):8578-99.

- [37] Fangzong W, Xiaobing L, Xiong X. Characteristics of the differential quadrature method and its improvement. Math Probl Eng, 2015;6:1–9.
- [38] Ghasemi M. High order approximations using spline-based differential quadrature method: implementation to the multidimensional PDEs. Appl Math Model. 2017;46:63–80.
- [39] Eftekhari SA, Jafari AA. Numerical simulation of chaotic dynamical systems by the method of differential quadrature. Sci Iran. 2012;19(5):1299–315.
- [40] Shu C, Chew YT, Richards BE. Generalized differential and integral quadrature and their application to solve boundary layer equations. Int J Numer Methods Fluids. 1995;21(9):723–33.
- [41] Civalek O. Application of differential quadrature (DQ) and harmonic differential quadrature (HDQ) for buckling analysis of thin isotropic plates and elastic columns. Eng Struct. 2004;26(2):171– 86.
- [42] Li H, Shantanu S. Mulay, Meshless Methods and their Numerical Properties. Boca Raton London New York: CRC PressTaylor and Francis Group; 2013 p 132-134, 189–192.
- [43] Yücel U, Boubaker K. U. Yu čel, K. Boubaker, Differential quadrature method (DQM) and Boubaker Polynomials Expansion Scheme (BPES) for efficient computation of the eigenvalues of fourth-order Sturm-Liouville problems. Appl Math Model. 2012;36(1):158–67.