

Comparison of Mesh-Based and Meshless Methods for Solving the Mathematical Model Arising of Launching Devices During the Firing

S. Sarabadan*, M. Kafili

Departeman of Mathematics, Imam Hossein University, Tehran, Iran

Abstract

In this paper, we apply different mesh-based and meshless methods for solving matrix system carried out from modelling the rockets stability during the firing. Sloped rocket launch and its stability during the firing is one of the most important kinds of defense instruments. The rockets stability during the firing path especially when they are unguided is very important for firing precision. Two mesh-based schemes as finite difference and B-spline methods and two meshless schemes as radial basis function (RBF) and radial basis functions based on finite difference (RBF-FD) are employed for solving underlying system. Numerical results are presented as tabular forms. They show that computational errors and CPU time for RBF-FD as a meshless method are better than other methods.

Keywords

Mesh-Based Method, Meshless Method, Sloped Rocket Launch

Received: July 5, 2015 / Accepted: July 22, 2015 / Published online: August 8, 2015

© 2015 The Authors. Published by American Institute of Science. This Open Access article is under the CC BY-NC license.

<http://creativecommons.org/licenses/by-nc/4.0/>

1. Introduction

Numerical methods can be divided into two major categories: Mesh-based and Meshless methods. The traditional mesh-based methods such as finite difference method (FDM) and B-spline method are widely used in many fields of science and engineering and some powerful packages have been developed for them. Some limitations in mesh generation, remeshing and constructing the approximation scheme tend the public interests to apply the meshless methods that remove the limitations of classical mesh-based methods. Meshless methods have been proved to treat scientific and engineering problems efficiently. They apply only a cloud of points without any information about nodal connections. One of the most popular meshless methods is constructed by radial kernels as basis called radial basis function (RBF) method. It is (conditionally) positive definite, rotationally and transnationally invariant. In other hand, system matrices with high condition numbers often result in this method is

one of the most defense of it. These properties make its application straightforward specially for approximation problems with high dimensions. RBFs include two useful characteristics: a set of scattered centers with possibility of selecting their locations and existence of a free positive parameter known as the shape parameter. "Shape parameter" is a customary name for RBF free parameter in the literature, but it is also called scale parameter, width, or (reciprocal of the) standard deviation. A progress version of RBF method is radial basis function based on finite difference that is the local version of RBF method. This method is very useful for solving systems with initial perturbed or random conditions [1, 2].

Here, we intend to compare both mesh-based and meshless methods for solving a system of first order differential equations arising of launching devices oscillations during the firing. Due to, the finite difference and B-spline methods as mesh-based schemes and RBF and RBF-FD methods as meshless schemes are compared in accuracy and CPU time.

* Corresponding author

E-mail address: s.sarabadan@yahoo.com (S. Saraabadan), kafilbaba@yahoo.com (M. Kafili)

The rest of the paper is organized as follows: The system of first order differential equations arising of launching devices oscillations during the firings explained in Section 2. In the Section 3, a brief review on the four mentioned methods is presented, then applied for underlying system. The numerical results are shown in the Section 4 and methods are compared and interpreted in tabular form.

2. The Mathematical Model

The study of launching device oscillations during the firing is necessary for the design of precise and efficient rocket-launching device systems, especially in the case of unguided rockets. We suppose that the launching device and the moving rocket form a complex oscillating system that join together a sum of rigid bodies bound by elastic elements (the vehicle chassis, the tilting platform and the rockets in the containers) [3]. Some authors have been considered all forces and moments to a real analysis of problem [3, 4]. It results in a matrix form of the second order differential equations system that describes the matrix form of dynamic equations of the rocket-launching device system motion.

Suppose the independent unknown dynamic variables of the rocket-launching device system motion can be presented in the form of the following column vector [4]:

$$X_{6 \times 1} = [s \ \varphi_y \ \varphi_z \ z_s \ \gamma_x \ \gamma_y]^T, \quad (2.1)$$

where the vehicle chassis translation z_s , the vehicle chassis rotation γ_y (the chassis pitch movement), the vehicle chassis rotation γ_x (the chassis rolling movement), the tilting platform rotation φ_z (the gyration movement around the vertical axes), the tilting platform rotation φ_y (the pitch movement) and the rocket translation s , are components of X . So, one can obtain the matrix form of the second order differential equations system that describes the rocket-launching system components motion:

$$\ddot{X}_{6 \times 1} = B_{6 \times 6} \cdot \dot{X}_{6 \times 1} + C_{6 \times 6} \cdot X_{6 \times 1} + N_{6 \times 15} \cdot \xi_{15 \times 1} + F_{6 \times 3} \cdot \phi_{3 \times 1} \quad (2.2)$$

Where $B_{6 \times 6} = (b_{i,j})_{i,j=1,\dots,6}$ is the matrix of the velocities coefficients, $\dot{X}_{6 \times 1}$; $C_{6 \times 6} = (c_{i,j})_{i,j=1,\dots,6}$ is the matrix of the unknown variables coefficients X . $N_{6 \times 15} = (n_{i,j})_{i=1,6 \atop j=1,15}$ is the matrix of the coefficients for the nonlinear combinations of the unknown variables:

$$\xi_{15 \times 1} = \begin{bmatrix} \dot{\gamma}_x^2 \dot{\gamma}_y^2 \dot{\gamma}_x \dot{\gamma}_y \dot{\varphi}_y^2 \dot{\varphi}_z^2 \dots \\ \dots \dot{\varphi}_y \dot{\varphi}_z \dot{\gamma}_x \dot{\varphi}_y \dot{\gamma}_x \dot{\varphi}_z \dot{\gamma}_y \dot{\varphi}_y \dots \\ \dots \dot{\gamma}_y \dot{\varphi}_z \dot{\gamma}_x \dot{s} \dot{\gamma}_y \dot{s} \dot{\varphi}_y \dots \\ \dots \dot{s} \dot{\varphi}_z \ \mu \sqrt{F_{Ry}^2 + F_{Rz}^2} \end{bmatrix}^T \quad (2.3)$$

For more information about the components of matrices (1.3), that can be specified randomly, one can see [3, 4]. And, the $F_{6 \times 3} = (f_{i,j})_{i=1,6 \atop j=1,3}$ is the matrix of the external forces that acts on the system:

$$\Phi_{3 \times 1} = [g \ T \ F_{jet}]^T. \quad (2.4)$$

The vector (2.4) is used to express the influence of the external forces on the motion system. In this vector, the first term corresponds to the weight force, the second term corresponds to the rocket thrust and the last term to the rocket jet force [4].

The mathematical model can be used to study any launching device like the underlying problem [3, 4]. For solving the underlying system, at the first, one can be reduce the system of second order differential equations (2.2) to a system of first order differential equations [4], we must introduce the following variables:

$$v_s = \dot{s} \quad (2.5)$$

$$v_{zs} = \dot{z}_s \quad (2.6)$$

$$\omega_{\varphi_y} = \dot{\varphi}_y \quad (2.7)$$

$$\omega_{\varphi_z} = \dot{\varphi}_z \quad (2.8)$$

$$\omega_{\gamma_x} = \dot{\gamma}_x \quad (2.9)$$

$$\omega_{\gamma_y} = \dot{\gamma}_y \quad (2.10)$$

Using those new variables (2.5)-(2.10), the unknown variables vector can be presented as following [4]:

$$X_{12 \times 1} = [v_s \ \omega_{\varphi_y} \ \omega_{\varphi_z} \ v_{zs} \ \omega_{\gamma_x} \ \omega_{\gamma_y} \ s \ \varphi_y \ \varphi_z \ z_s \ \gamma_x \ \gamma_y]^T \quad (2.11)$$

Using the notations (2.5)-(2.10) and the vector (2.11), as well as the equation (2.2), we obtain the new matrix form of the first order differential equations, which describes the motion of the rocket-launching device system:

$$\dot{X}_{12 \times 1} = P_{12 \times 12} \cdot X_{12 \times 1} + Q_{12 \times 15} \cdot \xi_{15 \times 1} + R_{12 \times 3} \cdot \phi_{3 \times 1} \quad (2.12)$$

where,

$$P_{12 \times 12} = \begin{pmatrix} B_{6 \times 6} & C_{6 \times 6} \\ I_{6 \times 6} & 0_{6 \times 6} \end{pmatrix} \quad (2.13)$$

$$Q_{12 \times 15} = \begin{pmatrix} N_{6 \times 15} \\ 0_{6 \times 15} \end{pmatrix}$$

$$R_{12 \times 3} = \begin{pmatrix} F_{6 \times 3} \\ 0_{6 \times 3} \end{pmatrix}$$

Which $0_{6 \times 6}$, $0_{6 \times 15}$ and $0_{6 \times 3}$ are zeros matrices and as mentioned before other blocks are the random matrices that their elements are random values that impose the launching

device during the firing. Here, we solve the matrix system (2.12) by applying radial basis functions. The 6 scalar equations are necessary to calculate the 6 unknown variables that describe the movement of the rocket-launching device system during firing $(s, \varphi_y, \varphi_z, z_s, \gamma_x, \gamma_y)$ while the other scalar equations allow to compute the evolutions of the differentials of 6 main unknown variables defined with (2.5)-(2.10) [4].

3. A Brief Review on Mesh-Based and Meshless Methods

In this section we proposed a summary of the applied mesh-based schemes: FDM and B-spline schemes, and meshless methods: RBF and RBF-FD schemes. So, these methods apply to solve the system matrix (2.12).

3.1. Mesh-Based Methods: FDM

Our goal is to approximate solutions to differential equations, i.e., to find a function (or some discrete approximation to this function) which satisfies a given relationship between various of its derivatives on some given region of space and/or time, long with some boundary conditions along the edges of this domain. In general this is a difficult problem and only rarely can an analytic formula be found for the solution. A finite difference method proceeds by replacing the derivatives in the differential equations by finite difference approximations. This gives a large algebraic system of equations to be solved in place of the differential equation, something that is easily solved on a computer[5].

Let $u(x)$ represent a function of one variable that, unless otherwise stated, will always be assumed to be smooth, meaning that we can differentiate the function several times and each derivative is a well-defined bounded function over an interval containing a particular point of interest \bar{x} .

Suppose we want to approximate $u'(\bar{x})$ by a finite difference approximation based only on values of u at a finite number of points near \bar{x} . One obvious choice would be to use

$$D_+u(\bar{x}) \equiv \frac{u(\bar{x}+h)-u(\bar{x})}{h} \tag{3.1.1}$$

for some small value of h . This is motivated by the standard definition of the derivative as the limiting value of this

expression as $h \rightarrow 0$. Note that $D_+u(\bar{x})$ is the slope of the line interpolating u at the points \bar{x} and $\bar{x} + h$.

The expression (3.1.1) is a one-sided approximation to u' since u is evaluated only at values of $x \geq \bar{x}$. Another one-sided approximation would be

$$D_-u(\bar{x}) \equiv \frac{u(\bar{x})-u(\bar{x}-h)}{h} \tag{3.1.2}$$

Each of these formulas gives a first order accurate approximation to $u'(\bar{x})$, meaning that the size of the error is roughly proportional to h itself.

Another possibility is to use the centered approximation

$$D_0u(\bar{x}) \equiv \frac{u(\bar{x}+h)-u(\bar{x}-h)}{2h} = \frac{1}{2}(D_+u(\bar{x}) + D_-u(\bar{x})). \tag{3.1.3}$$

This is the slope of the line interpolating u at $\bar{x} - h$ and $\bar{x} + h$, and is simply the average of the two one-sided approximations defined above. It should be clear that we would expect $D_0u(\bar{x})$ to give a better approximation than either of the one-sided approximations. In fact this gives a second order accurate approximation, the error is proportional to h^2 and hence is much smaller than the error in a first order approximation when h is small. Simplicity of applying it is one of the important advantages and the mesh generation and remeshing spatially for complex domains is one of its important disadvantages.

For solving the matrix system (2.12) by the FDM, at the first, we discretize the interval $[a, b]$ for t with n nodes with the step size $h = (b - a)/(n - 1)$ so, by collocating (2.12) and approximating the components of $\dot{X}_{12 \times 1}$ with the (3.1.3), so that for the nodal points $m = 2, \dots, n - 1$:

$$\dot{X}_i(t_m) \cong \frac{X(t_{m+1})-X(t_{m-1})}{2h}, \quad i = 1, 2, \dots, 12. \tag{3.1.4}$$

Where at the first node t_1 and the end node t_n the values of X is determined randomly and zero respectively. This scheme leads to 12 multi-diagonal matrix system. By solving the resulting systems one can obtain the solution in nodal points.

3.2. Mesh-Based Methods: B-Spline Method

In this section we describe the B-spline collocation method to use the matrix system (2.12). Let $t\Delta = \{a = t_0 < t_1 < \dots < t_N = b\}$ be the partition in $[a, b]$. We define the cubic B-spline for $j = -1, 0, \dots, N + 1$ by the following relation,

$$B_{3,j} = \frac{1}{h^3} \begin{cases} (t - t_{j-2})^3, & t \in [t_{j-2}, t_{j-1}], \\ h^3 + 3h^2(t - t_{j-1}) + 3h(t - t_{j-1})^2 - 3(t - t_{j-1})^3, & t \in [t_{j-1}, t_j], \\ h^3 + 3h^2(t_{j+1} - t) + 3h(t_{j+1} - t)^2 - 3(t_{j+1} - t)^3, & t \in [t_j, t_{j+1}], \\ (t_{j+2} - t)^3, & t \in [t_{j+1}, t_{j+2}], \\ 0, & \text{otherwise.} \end{cases} \tag{3.2.1}$$

Our numerical treatment for solving (2.12) using the collocation method with cubic B-splines is to find an approximate solution $\hat{S}(x)$ to exact solution $u(x, t)$ in the form,

$$\hat{S}(t) = \sum_{j=-1}^{N+1} \hat{c}_j B_j(t) \tag{3.2.2}$$

Where \hat{c}_j are unknown time dependent parameters to be determined from the boundary conditions and collocation of the differential equation. The values of $B_j(t)$ and its two derivatives may be tabulated as in Table 1.

Table 1. Values of B_j and its first derivative at the nodal points.

t	t_{j-2}	t_{j-1}	t_j	t_{j+1}	t_{j+2}
B_j	0	1	4	1	0
B'_j	0	$3/h$	$-12/h^2$	$6/h^2$	0

Using approximate function (3.2.2) and cubic B-spline (3.2.1), the approximate values at the knots of $\hat{S}(t)$ and its derivatives are determined in terms of the time dependent parameters $\hat{c}_j(t)$ as,

$$\hat{S}(t) = \hat{c}_{j-1} + 4\hat{c}_j + \hat{c}_{j+1} \tag{3.2.3}$$

$$h\hat{S}'(t) = 3(\hat{c}_{j-1} - \hat{c}_{j+1}) \tag{3.2.4}$$

Now for solving the matrix system (2.12) suppose that for each components of $X_{12 \times 1}$ the $\hat{S}_i(t)$ satisfies the (2.12) plus the boundary conditions for $t = a$ (that its value determined randomly) and $t = b$ (that its value can specified zero because of it is the end of process). This scheme leads us to the 12 tri-diagonal matrix system which by solving them one can obtain the solution of (2.12).

3.3. Meshless Methods: RBF

One of the most popular meshless methods is constructed by radial kernels as basis called radial basis function (RBF) method. It is (conditionally) positive definite, rotationally and translationally invariant. These properties make its application straightforward specially for approximation problems with high dimensions. Some of the well-known RBFs are as follows,

Multiquadric (MQ) : $\sqrt{1 + \varepsilon^2 r^2}$

Inverse multiquadric (IMQ): $(\sqrt{1 + \varepsilon^2 r^2})^{-1}$

Gaussian (GA): $\exp(-\varepsilon^2 r^2)$

where r is the Euclidean distance between any two points $x, y \in \mathbb{R}^d$, i.e. $r = \|x - y\|_2$, [1, 2]. The RBFs include two useful characteristics: a set of scattered centers $X_C = \{x_1^c, \dots, x_N^c\} \subseteq \mathbb{R}^d$ with possibility of selecting their locations and existence of a free positive parameter, ε , known as the

shape parameter.

Assume the ε_j be the shape parameter corresponding to j^{th} center x_j^c , we use following notation for translation of RBFs at j^{th} center,

$$\phi_j(x, \varepsilon_j) = \phi(\|x - x_j^c\|_2, \varepsilon_j), \quad j = 1, \dots, N.$$

Let data values $f_j^c = f(x_j^c)$ are given, the function $f(x)$ will be approximated using a linear combination of translates of a single RBF so that,

$$f(x) \approx S(x) = \sum_{j=1}^N \alpha_j \phi_j(x, \varepsilon_j), \tag{3.3.1}$$

where the unknown coefficients $\{\alpha_j\}_{j=1}^N$ will be determined by collocating (3.3.1) at the same set of centers, X_C .

The shape parameter plays an important role in RBFs, the choice of it controls the shape of the basis functions and interchanges the error and stability of interpolation process. This behavior is manifested as a classical tradeoff between accuracy and stability or Uncertainty Principle [6] which refers to the fact that an RBF approximant can not be accurate and well-conditioned at the same time.

Two scenarios are available for choosing shape parameters: constant shape parameter (CSP) strategies that all of shape parameters take the same value and variable shape parameter (VSP) strategies that assign different values to shape parameters corresponding to each center. Many scientists and mathematicians use CSPs in RBF approximations [7, 8, 9] because of their simple analysis as well as solid theoretical background rather than VSPs, but there are numerous results from a large collection of applications [10, 11, 12, 13] indicating the advantages of using VSPs.

For solving the matrix system (2.12), we approximate the components of $X_{12 \times 1}$ with the RBF interpolant (3.3.1) so that:

$$X_i \approx \sum_{j=1}^N \alpha_{ij} \phi_j(t, \varepsilon_j), \quad i = 1, 2, \dots, 12 \tag{3.3.2}$$

Where $\varepsilon_j = 1, j = 1, \dots, N$ is constant. By differentiating from (3.10) the components of $\dot{X}_{12 \times 1}$ obtained as follows:

$$\dot{X}_i \approx \sum_{j=1}^N \alpha_{ij} \dot{\phi}_j(t, \varepsilon_j), \quad i = 1, 2, \dots, 12 \tag{3.3.3}$$

Substituting the equations (3.3.2), (3.3.3) in (2.12) and collocating it in same $n + 1$ centers, we obtain 12 system of equations with unknown parameters $\alpha_{ij}, i = 1, 2, \dots, 12, j = 1, \dots, N$. Therefore, one can be approximate the unknown variables $X_i, i = 1, 2, \dots, 12$ from equations (3.3.2).

3.4. Meshless Methods: RBF-FD

As mentioned before to calculate the derivatives of function u at node x_i , first we approximate the function u by linear

combination of RBFs as (3.3.1), then apply the derivative operator L on the above RBF interpolant and evaluate it at the desired node location:

$$Lu(x_i) = \sum_{j=1}^N \alpha_j L\phi_j(x_i, \varepsilon_j), \tag{3.4.1}$$

The above formula is a global derivation. That is, to calculate the discretized derivative operator at one node, all the nodes in the domain are used. In RBF-FD method for each stencil, only n nodes out of the N nodes in the domain are used and the differential operator L is approximated at the node x_i by a linear combination of the function values u at the neighboring $n - 1$ nodes locations:

$$\underbrace{\begin{bmatrix} \phi_1(x_1, \varepsilon_1) & \phi_2(x_1, \varepsilon_2) & \dots & \phi_n(x_1, \varepsilon_n) \\ \phi_1(x_2, \varepsilon_1) & \phi_2(x_2, \varepsilon_2) & \dots & \phi_n(x_2, \varepsilon_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n, \varepsilon_1) & \phi_2(x_n, \varepsilon_2) & \dots & \phi_n(x_n, \varepsilon_n) \end{bmatrix}}_A \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}_w = \underbrace{\begin{bmatrix} L\phi_1(x_i, \varepsilon_1) \\ L\phi_2(x_i, \varepsilon_1) \\ \vdots \\ L\phi_n(x_i, \varepsilon_1) \end{bmatrix}}_f \tag{3.4.3}$$

In our work, the operator L is the first order derivative operator and among radial basis functions, we use Multiquadric (MQ) function. It can be shown that the system matrix A in (3.4.3) for this function is invertible. Now, we use the Maple software and explicitly present the weight coefficients for approximating the first order derivative. In general, assume that set $\{x_1, \dots, x_N\}$ be a non-uniform partition of the interval $[a, b]$. To approximate first order derivative of function u at points of x_i we apply a three point central scheme: $n = 3$ and $[x_i - h_1, x_i + h_2]$. After solving linear system (3.4.3) and suppose that $h_1 = h_2 = h$ and $c \gg h$ the weight coefficients of RBF-FD method for approximating the first derivative of function u are given as:

$$w_1^{(1)} = -w_3^{(1)} = -\frac{1}{2h} \left(1 + \frac{h^2}{2c^2} \right), \quad w_2^{(1)} = 0, \tag{3.4.4}$$

Which coincides with the standard central difference approximation to the first derivative with a correction term of order $\frac{h}{c^2}$ [14].

Now, we present a scheme for solving the matrix system (2.12). At the first, we approximate the values of components of $X_{12 \times 1}$ in node t_j by RBF-FD using $n = 3$ and $[t_j - h, t_j + h]$, as well as the components of $X_{12 \times 1}$ approximate so that:

$$X_i(t_j) = \frac{X_i(t_j-h) + X_i(t_j+h)}{2}$$

After substituting the weights (3.4.4), one can obtain 12 tri-diagonal matrix systems results in the solutions in nodal points t_j .

4. Numerical Results

In this Section, we compare both mesh-based and meshless

$$Lu(x_i) = \sum_{k=1}^n w_k u(x_k), \tag{3.4.2}$$

Unlike the conventional FD method, which uses the polynomials of degree $n - 1$ or less for computing the weights w_k in above linear combination, RBF-FD method calculates these weights by imposing the requirement that the above linear combination must be exact for RBFs, $\{\phi_k(x, \varepsilon_k)\}_{k=1}^n$, centered at each of the node locations $\{x_k\}_{k=1}^n$. Combining these constraints, the solution for the RBF-FD weights can be determined from the following linear system:

methods based on four mentioned schemes. Our numerical results let us carry out some results about advantages and disadvantages of different methods and finally about mesh-based and meshless methods. All numerical results carried out using Maple software. In all schemes, the matrices $P_{12 \times 12}, Q_{12 \times 15}, P_{12 \times 3}$ are produced with Maple's Random function that randomly assigns elements in interval $[0, 0.5]$ for the underlying matrices. The MQ function with the constant shape parameter $\varepsilon_j = 1$ for $j = 1, \dots, N$ is applied for basis function in both RBF and RBF-FD schemes. Support interval is specified as $[0, 5]$, also the number of nodal points (in meshless schemes: number of centers) are selected so that $n = 6$. Based on theoretical and numerical experiments [8, 9, 10, 11] it is clear that increasing n results in more accurate results. The approximation of the components of $X_{12 \times 1}$ are shown in tabular forms in tables 2-4 for some t , then some interpretations and experiments of applying different approaches are presented. Notice that by passing time (increasing t), the values of oscillations parameters tend to zero. It is clear that output results be different compared with those are presented for other random matrices (2.13), but according to our observations, they are similar to presented results for any random matrices.

Table 2. Comparison of different schemes for $t = 0.5$.

X_i	FDM	B-spline method	RBF	RBF-FD
X_1	-2.0001	-1.6544	-0.9563	-0.7650
X_2	0.3999	0.2225	0.1567	0.1198
X_3	-0.0190	-0.0122	-0.0065	-0.0054
X_4	-0.1666	-0.1589	-0.1011	-0.0776
X_5	0.0795	0.0650	0.0655	0.0400
X_6	-0.0561	-0.0444	-0.0340	-0.0201
X_7	-0.0231	-0.0226	-0.0121	-0.0098
X_8	-0.0990	-0.0760	-0.0602	-0.0341

X_i	FDM	B-spline method	RBF	RBF-FD
X_9	0.0808	0.0685	0.0700	0.0614
X_{10}	0.0861	0.0790	0.0502	0.0440
X_{11}	0.0902	0.0599	0.0407	0.0384
X_{12}	0.0357	0.0309	0.0255	0.0189
CPU time(second)	0.37	0.21	0.45	0.09

Table 3. Comparison of different schemes for $t = 2.5$.

X_i	FDM	B-spline method	RBF	RBF-FD
X_1	-1.0191	-0.8789	-0.5950	-0.4711
X_2	0.1391	0.1045	0.0865	0.0705
X_3	0.0142	0.0103	-0.0094	0.0074
X_4	-0.3093	-0.2999	-0.2374	-0.2090
X_5	0.1577	0.1209	0.0997	0.0413
X_6	-0.0342	-0.0268	-0.0104	-0.0099
X_7	0.0091	0.0079	0.0043	0.0038
X_8	-0.0788	-0.0595	-0.0375	-0.0214
X_9	0.1672	0.1544	0.1286	0.1174
X_{10}	0.0502	0.0491	0.0205	0.0188
X_{11}	0.1399	0.1390	0.1208	0.1186
X_{12}	0.1184	0.1096	0.0954	0.0609
CPU time(second)	0.37	0.21	0.45	0.09

Table 4. Comparison of different schemes for $t = 4.5$.

X_i	FDM	B-spline method	RBF	RBF-FD
X_1	-0.0177	-0.0185	-0.0054	-0.0001
X_2	0.0391	0.0287	0.0098	0.0019
X_3	0.0051	0.0048	0.0031	0.0008
X_4	-0.1291	-0.1019	-0.0848	-0.0666
X_5	0.0886	0.0696	0.0475	0.0109
X_6	-0.0099	-0.0087	-0.0054	-0.0039
X_7	-0.0009	-0.0006	-0.0005	-0.0001
X_8	-0.0502	-0.0299	-0.0154	-0.0076
X_9	0.0444	0.0390	0.0401	0.0389
X_{10}	0.1021	0.0991	0.0868	0.0753
X_{11}	0.0608	0.0582	0.0397	0.0115
X_{12}	0.0299	0.0289	0.0221	0.0190
CPU time(second)	0.37	0.21	0.45	0.09

As it is shown in numerical results, the RBF-FD scheme as a meshless method results in less values as solutions because of less computational errors rather than other results. As well as the CPU time in this case is less than others. The RBF scheme as another meshless method has run 0.45 second that is more than CPU time for other methods that its reason is the system matrices with high condition numbers in this scheme. It results in that the mesh-based methods are better than the RBF method as meshless method but the RBF-FD as a meshless method that is produced by combining of FDM and RBF methods has better performance rather than other approaches.

Notations and Symbols

X : independent unknown dynamic variables of the rocket-

- launching device system motion
- z_s : vehicle chassis translation
- γ_y : chassis pitch movement
- γ_x : chassis rolling movement
- φ_z : gyration movement around the vertical axes
- φ_y : pitch movement
- s : rocket translation
- ξ : matrix of the coefficients for the nonlinear combinations of the unknown variables
- Φ : external forces that acts on the system
- ε : shape parameter
- r : Euclidean distance
- X_C : set of centers
- v_s : derivative of s
- v_{z_s} : derivative of z_s
- ω_{φ_y} : derivative of φ_y
- ω_{γ_x} : derivative of γ_x
- ω_{γ_y} : derivative of γ_y

References

- [1] H. Wendland, Scattered Data Approximation, Cambridge University Press, (2005).
- [2] M. D. Buhmann, Radial Basis Functions: Theory and Implementation, University of Gissen, Cambridge University Press, (2004).
- [3] P. Somoia, C. Moldoveanu, Numerical research on the stability of launching devices during firing, Defence Technology 9, (2013), 242-248.
- [4] P. Somoia, F. Moraru, D. Safta, C. Moldoveanu, A mathematical model for the motion of a rocket-launching device system on a heavy vehicle, Military technical academy, Romania, (2012).
- [5] R. J. Leveque, Finite difference methods for differential equations, University of Washington, (2005).
- [6] R. Schaback, Improved error bounds for scattered data interpolation by radial basis functions, Math Comput, 68, (1999), 201-206.
- [7] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, J Geophys Res, 76, (1971), 1905-1915.
- [8] R. Franke, Scattered data interpolation: tests of some methods, Math Comput, 38, (1982), 181-200.
- [9] C. S. Huang, C. F. Lee, A. D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method, Eng Anal Bound Elem, 31, (2007), 614-623.

- [10] R. E. Carlson, T. A. Foley, The parameter \mathbb{R}^2 in multiquadric interpolation, *Comput Math Appl*, 21, (1991), 29-42.
- [11] E. J. Kansa, R. E. Carlson, Improved accuracy of multiquadric interpolation using variable shape parameters, *Comput Math Appl*, 24, (1992), 99-120.
- [12] E. J. Kansa, R. C. Aldredge, L. Ling. Numerical simulation of two-dimensional combustion using mesh-free methods, *Eng Anal Bound Elem*, 33, (2009), 940-950.
- [13] S. A. Sarra, D. Sturgill, A random variable shape parameter strategy for radial basis function approximation methods, *Eng Anal Bound Elem*, 33, (2009), 1239-1245.
- [14] V. Bayona, M. Moscoso, M. Carretero, M. Kindelan, RBF-FD formulas and convergence properties, *Journal of computational physics*, 229, (2010), 8281-8295.