{MERAA} Mathematics in Education, Research and Applications

Math Educ Res Appl., 2016(2), 1

Received: 2016-04-20
Accepted: 2016-06-06
Online published: 2016-10-30
DOI: http://dx.doi.org/10.15414/meraa.2016.02.01.08-14

Original paper

Numerical differentiation of stochastic function

Jozef Rédl*, Dušan Páleš

Slovak University of Agriculture, Faculty of Engineering, Department of Machine Design
Nitra, Slovak Republic

ABSTRACT

In this paper we are presenting the derivation of the finite difference approximation of the first derivative of the discrete stochastic function. The set of data was represented with time depending stochastic function of center of gravity dislocation and velocity with constant step size. The stochastic function was obtained from real experimental measurement and was processed with *Dynstab (Dynamic Stability)* software. Applied forward, backward and central differences algorithms were written in Microsoft Visual C#® 2010 programming language as a part of software named *Dynstat (Dynamic Statistics)*. These algorithms were used to calculate the first derivative of the function and error of differentiation. We compared the results of known precise solutions with calculated results. We also confront the results calculated of known generated trigonometric function. With this manner we are evaluated the inaccuracy of the stochastic function numerical differentiation.

KEYWORDS: numerical differentiation, algorithm, programming, error evaluating

JEL CLASSIFICATION: N40

INTRODUCTION

Solutions of many technical problems are based on obtaining the differentials of technical functions. The searched differentials of function are often evaluated algebraically where the accuracy of results is guaranteed. If the values technical functions are measured continuously its differentials must be evaluated numerically. In this case the accuracy is not guaranteed. The high instability of differentiating process is the reason for the algebraic evaluation if it's possible. In the specific case the algebraic evaluation is not possible and we have to design an algorithm of numerical solution. Methods of numerical differentiation algorithms are known and published by many authors.(e.g. [1], [2], [4]). In this contribution we deal with the derivation and the application of the finite difference method. This method has three variants

* Corresponding author: Jozer Rédl, Slovak University of Agriculture, Faculty of Engineering, Department of Machine Design, Tr. A. Hlinku 2, 949 76 Nitra, Slovak Republic. E-mail: jozef.redl@uniag.sk

namely forward, backward and central differences and they has defined in [1], [12]. Practical applications were published with [1], [3], [11]. Mathematical solution accuracy depends first of all on the chosen step of differentiation and from the order of approximation polynomial. The next factors are the applied method as well as the chosen numerical data type. When the exact solution is available and is obtained from numerical integration how is defined in [8], feasible solution of the inaccuracy we can get from results comparison.

MATERIAL AND METHODS

Finite difference method

The basic method of numerical differentiation (forward, backward, central) are defined in [4]. Its derivation we can realize from the initial value problem of the ordinary differentials equations in accordance with [2]. This method is the start point for derivation of numerical integration how is described in [6]. Let us assume the differential equation in the next form:

$$\dot{y} = f\left(x, y(x)\right). \tag{1}$$

The solution of this equation is the function y, and its satisfied at certain point $x_0 \in \langle a, b \rangle$, values η, hence $y(x_0) = \eta$. This initial value problem we call as a *Cauchy initial value problem*. To guarantee the existence of solution is necessary for the *Cauchy uniqueness* of solution in the area of function continuous substitute with *Lipschitz continuous condition*. The equation (1) rewriting to the form:

$$\dot{y} = f(x, y), \tag{2}$$

where $f(x, y)$ is the real vector function and is defined and continuous on the set $S = \langle a, b \rangle \times \mathcal{R}^m$. If the *Lipschitz condition* must be satisfied, there have to exist a constant L, which is independent on x, y such that: $\| f(x, y) - f(x, z) \| \leq L \| y - z \|$ for all $x_0 \in \langle a, b \rangle$ and all vectors y, z. Foremost we have to define the origin of the solving of initial value problem with numerical integration. The goal of the solution is the points calculation, $x_0 = a, x_1, x_2, \dots, x_n$, that are the approximations of exact solution points $(x_0), y(x_1), y(x_2) \dots y(x_n)$, in points $x_0, x_1, x_2, \dots x_n$, where step of integration method is $h_n = x_{n+1} - x_n$ or $h_{n-1} = x_n - x_{n-1}$ and must have to be satisfied that $h_n > 0, \forall n$. If $h_n = h$ we talking about method with constant step. The numerical integration method is based on the application of the *Taylor's series*, where the exact solution is approximated with the *Taylor's polynomial* of the highest order in the form:

$$\begin{aligned} y(x_{n+1}) &= y(x_n + h_n) = \\ &= y(x_n) + h_n \dot{y}(x_n) + \frac{h_n^2}{2!}\ddot{y}(x_n) + \ldots + \frac{h_n^p}{p!} y(x_n) + \frac{h_n^{p+1}}{(p+1)!} y^{p+1}(\xi_n), \end{aligned} \tag{3}$$

as are published in [6], [8], [9]. Modifying equation (3) and neglecting the highest degree members we get:

$$y(x_{n+1}) = y(x_n + h_n) = y(x_n) + h_n \dot{y}(x_n) + \frac{h_n^{p+1}}{(p+1)!} y^{p+1}(\xi_n). \tag{4}$$

The application of the basic equation for differentiating we get:

$$\dot{y} = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} - \frac{1}{2!}h\ddot{f}(\xi) \quad (5)$$

where $E(f) = -\frac{1}{2!}h\ddot{f}(\xi)$ is the error of numerical differentiation. Substituting in the equation (5) with $x = x_i, h = x_{i+1} - x_i$, we get forward-difference method in the form:

$$f(\dot{x}_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{1}{2!}h\ddot{f}(\xi). \quad (6)$$

Changing indexing like that $x = x_i, h = x_i - x_{i-1}$, we get backward-difference method:

$$f(\dot{x}_{i-1}) = \frac{f(x_i) - f(x_{i-1})}{h} - \frac{1}{2!}h\ddot{f}(\xi_{i-1}). \quad (7)$$

If we change $x = \frac{1}{2}(x_{i-1} - x_i)$, where x_{i-1} and x_i are symmetrical about x, we get $x_{i-1} = x - h,\ h = \frac{1}{2}(x_i - x_{i-1})$.

Inserting to (5) we got:

$$f(\dot{x}_{i-1}) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{h^2}{6}f(\bar{\bar{\xi}}_{i-1}), \quad (8)$$

what is the central-difference formula. If we apply the same manner as defined above and at the deriving process we change the values of $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$ for solution as defined in [6] and combining the four equations of *Taylor series*, we get the formulations of highest degrees. There is the fourth degree form:

$$f(\dot{x}_{i-2}) = \frac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12h}. \quad (9)$$

Stochastic data set

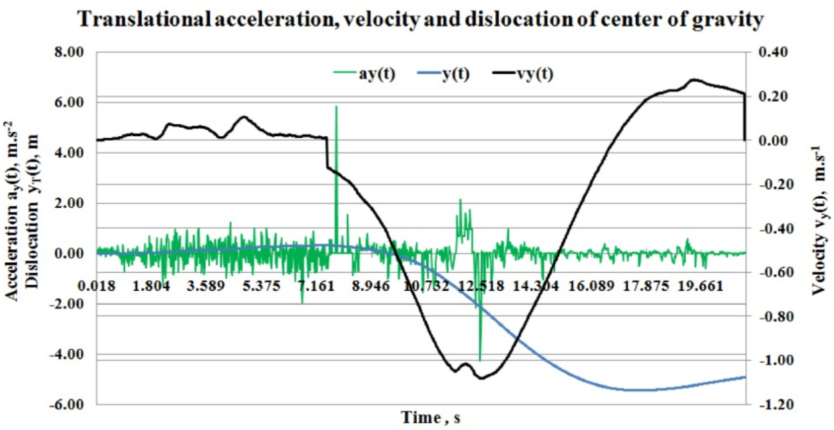

Fig. 1 Used technical function

We obtained the technical functions from the real experimental measurement of which we chose translational acceleration, translational velocity and dislocations of center of gravity moving with respect to the *y* axis. The methodology of discrete data processing in *Dynstab* software is described in [8]. Statistical processing of the stochastic technical functions is defined in [9]. The applied random data set is stationary and ergodic with zero mean value defined in the next form: $X = \{X(t); t \in (-\infty, +\infty)\}$. The graphs of stochastic functions $a_y(t), v_y(t), y(t)$ are depicted on the figure1.

RESULTS AND DISCUSSION

Visual C# algorithm

The relevant parts of algorithm were written in Microsoft Visual C# 2010 and we presented them in the abbreviated form. Non-mathematical parts of the algorithms (controls, forms, etc.) are not included in the list. Declarations of applied variables in the source code are:

```
private int dim; // dimension for arrays
private double[] h,fnc,dfnc; //Arrays of step, function, diff. function
```

Solving algorithms:

```
double stp; // step of differentiation
switch (diffmethComboBox.SelectedIndex)// select chosen method
      { case 0: //forward diff
        for (int i = 0; i < dim - 1; i++)
        { stp = h[0]; dfnc[i] = (1 / stp) * (fnc[i + 1] - fnc[i]);} break;
        case 1: //backward
        for (int i = 1; i < dim; i++)
        { stp = h[0]; dfnc[i - 1] = (1 / stp) * (fnc[i] - fnc[i - 1]);} break;
        case 2://central
        for (int i = 1; i < dim - 1; i++)
        { stp = 2 * h[0]; dfnc[i - 1] = (1 / stp) * (fnc[i + 1] –
                                          -fnc[i - 1]);} break;
        case 3://central-4th. order
        for (int i = 2; i < dim - 2; i++)
        { stp = 12 * h[0]; dfnc[i - 2] = (1 / stp) *(fnc[i - 2] -
                             8 * fnc[i - 1] + 8 * fnc[i+1] - fnc[i+2]);}}
```

To define the numeric variables we chose the type *Double* as defined in [5].

Evaluated results

Results of the numerical differentiation are depicted in figure 2. There is shown the exact solution as well as solution realized with variants of finite-difference method. We generated a sample function $y = \sin(x)$ and its exact solution $\dot{y} = \cos(x)$ in interval $x \in \langle 0,2\pi \rangle$ for comparison. The step of numerical differentiating of goniometric function was adapted to stochastic data count $n = 1178$. Solved step was $h = 0{,}00533377360541561$.

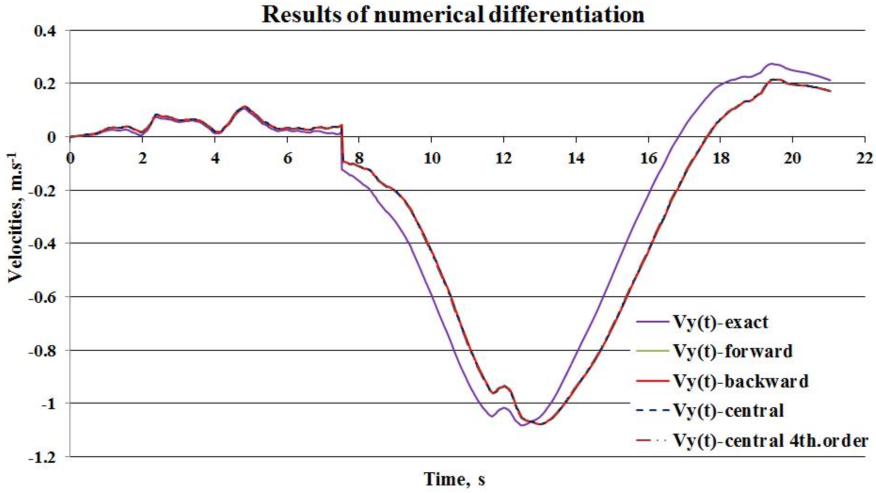

Fig. 2 Results of differentiating

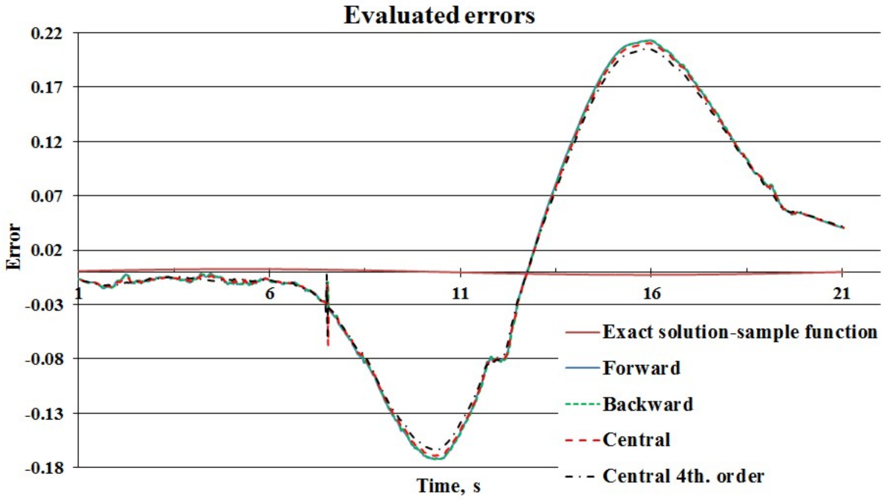

Fig. 3 Evaluated errors

The error of numerical differentiating for individual variants we evaluated with the next formula:

$$E(f)_{(i)} = {}_{exact}f'(x)_{(i)} - {}_{num}f'(x)_{(i)} \,. \tag{10}$$

Results are depicted in figure 3. The percentage expression of the average error is evaluated by:

$$E(f)_{(i)av\%} = \left[\frac{1}{n}\sum_{i=0}^{n} E(f)_{(i)}\right].100 \,. \tag{11}$$

CONCLUSIONS

In this paper we are dealing with the algorithm and the application of numerical differentiation. Applied methods were forward, backward a central a central 4th degree. Algorithm was written in the Microsoft Visual C# 2010 Professional programing language. The input data set was obtained from experimental measurement and processed with the *Dynstab* simulation software. Technical function had the stochastic character and was processed with the *Dynstab* software. For numerical differentiating purpose, we chose the functions of trajectory of center of gravity moving with respect to the y axis. The exact solution of differentiation we obtained from simulation program. According to the realized analysis we got the conclusion that the numerical differentiating of stochastic function is very unstable process. For the individual methods we evaluated the percentage average errors as follows. For forward we evaluated the error 19.19 %, for backward we evaluated the error 19.15 %, for central we evaluated the error 19.0 % and for central 4th degree we evaluated the error 18.7 %. Listed errors include also the error of the method as well as rounding errors. For comparison with generated sample periodical function $\sin(\boldsymbol{x})$ and its numerical differentiation, where we know an exact solution, is the average error $0.675 \cdot 10^{-6}$ %, what is classified as very high accuracy. In accordance with terms mentioned above, we can conclude that the stochastic characters of technical function have high degree of the participation on the instability and inaccuracy of the numerical differentiation.

Acknowledgement

The research was performed at the Department of Machine Design of the Faculty of Engineering of the Slovak University of Agriculture in Nitra and was supported by the Slovak Grant Agency for Science under grant VEGA No. 1/0786/14 titled “Effect of the environmental aspects of machinery interaction to eliminate the degradation processes in agro-technologies of field crops”.

REFERENCES

[1] Carnahan, B., Luther, H. A. & Wilkes, J. O. (1990). *Applied Numerical Methods.* Malabar: Krieger Publishing Company Orig. Ed. 1969, Reprint Ed.

[2] Collins, G. W. (2003). *Fundamental Numerical Methods and Data Analysis*. NASA ADS 2003. p. 254. Retrieved 2016-01-27 from http://ads.harvard.edu/books/1990fnmd.book/

[3] Dos Passos,W. (2010). *Numerical Methods, Algorithms and Tools in C# 2010*. New York: CRC Press, Taylor & Francis Group.

[4] Griewank, A. & Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. 2nd ed. SIAM.

[5] *ISO/IEC 23270:2006(E).* Information technology - Programming languages - C#, Second edition 2006-09-01. International Organization for Standardization, Switzerland. Retrieved 2016-01-27 from http:// www.iso.org

[6] Mathews, H. J. & Fink. K. K. (2004). *Numerical Methods Using Matlab*. 4th ed. Pearson.

[7] Rédl, J. (2010). *Development of technical applications for science and research*. Nitra: VES SPU Nitra (in Slovak).

[8] Rédl, J. (2012). Software For Determination Of Side Slope Stability Of Agricultural Vehicles. 2012. In: *MendelTech 2012: proceedings of scientific papers, March 29th-April 04th, 2012, Brno*. Brno: Mendel University in Brno.

[9] Rédl, J. (2012). Stochastic processes in vehicle dynamics. In: *Technika v technológiách agrosektora 2012*. Nitra: Slovenská poľnohospodárska univerzita v Nitre, pp.182-191 (in Slovak).

[10] Rédl, J., Váliková, V., Páleš, D. & Antl, J. (2015). Application of numerical integration in technical practice. *Mathematics in Education, Research and Applications*, 1(1), 12-17. doi:http://dx.doi.org/10.15414/meraa.2015.01.01.12-17

[11] Xu, J. (2008). *Practical Numerical Methods with C#*. UniCAD, Inc.

[12] Zarowski, J. Ch. (2004). *An Introduction To Numerical Analysis For Electrical And Computer Engineers*. New York: John Wiley & Sons, Inc.