{MERAA} Mathematics in Education, Research and Applications
ISSN 2453-6881
Math Educ Res Appl, 2015(1), 1

Received: 2014-11-12
Accepted: 2015-02-06
Online published: 2015-05-25
DOI:10.15414/meraa.2015.01.01.12-17

Original paper

Application of numerical integration in technical practice

Jozef Rédl*, Veronika Váliková, Dušan Páleš, Ján Antl

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

ABSTRACT
In this contribution we present the methodology of the solution of ordinary differential equation by the Runge – Kutta numerical method of fourth order. Analysed function is continuous and it has derivatives at every point. Records of the centre of the gravity velocities of agricultural technological vehicle are function values of the derivatives. Algorithm of numerical integration was implemented by the help of programming language C# in MS Visual Studio Pro 2010 developing environment. Trajectory of the centre of the gravity movement in three-dimensional space is the result of the listed algorithm.

KEYWORDS : numerical integration, algorithms, programming, trajectory

JEL CLASSIFICATION: N40

INTRODUCTION

When realising a measurement in agricultural conditions, technical exciting functions are usually measured continually with the constant increase of time. In this case, the use of Runge – Kutta method of fourth order, implemented into the algorithm in suitable programming language, is the most advantageous for numerical integration. Utilization of this method is extensive in science as well as in research.

Solution of the system of algebraic differential equations in form:

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

is listed in [1]. In [5] is introduced numerical integration of differential algebraic systems. The use of Runge – Kutta method to solve non-linear system of differential equations is presented in [3]. When time increase (or other parameter used as a step) is variable, it is appropriate to use modified Runge – Kutta – Nyström method, which is modified for variable step of numerical integration. This issue is closely addressed in [2]. Algorithmisation and implementation of numerical methods in programming language C++ was published in [6]. In [4] the issues of stability of numerical integration of linear differential equations by use of

* 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

Runge–Kutta method is described. Methods of numerical integration Runge–Kutta of higher order and solution of differential equations by these methods were published in [8]. Methods were implemented into programming language C#.

MATERIAL AND METHODS

Algorithm of derivation of Runge – Kutta method is published by several authors, for example in [1]. Implementation into C# language was presented in [7]. The base of Runge – Kutta method is the formulation of difference between *y* values in points x_{n+1} a x_n in form:

$$y_{n+1} - y_n = \sum_{i=1}^{m} w_i k_i, \qquad (2)$$

where w_i are constants and $k_i = h_n f(x_n + \alpha_i h_n, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j)$, where $h_n = x_{n+1} - x_n$ and $\alpha_1 = 0$.

By writing of both sides of equation (2) into Taylor series for i=1..4 members of the series, we get the system of equations for k_i and h_n^i. By comparing we obtain system of parameters:

$$\begin{gathered}
w_1 + w_2 + w_3 + w_4 = 1,\ w_2\alpha_2 + w_3\alpha_3 + w_4\alpha_4 = \frac{1}{2}, \\
w_2\alpha_2^2 + w_3\alpha_3^2 + w_4\alpha_4^2 = \frac{1}{3}, \\
w_3\alpha_2\beta_{32} + w_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) = \frac{1}{6},\ w_2\alpha_2^3 + w_3\alpha_3^3 + w_4\alpha_4^3 = \frac{1}{4}, \\
w_3\alpha_2^2\beta_{32} + w_4(\alpha_2^2\beta_{42} + \alpha_3^2\beta_{43}) = \frac{1}{12}, \\
w_3\alpha_2\alpha_3\beta_{32} + w_4(\alpha_2\beta_{42} + \alpha_3\beta_{43})\alpha_4 = \frac{1}{8}, \\
w_4\alpha_2\beta_{32}\beta_{43} = \frac{1}{24}.
\end{gathered} \qquad (3)$$

If we solve this system with two degrees of freedom we get:

$$\begin{gathered}
w_1 = \frac{1}{2} + \frac{1 - 2(\alpha_2 + \alpha_3)}{12\alpha_2\alpha_3},\ w_2 = \frac{2\alpha_3 - 1}{12\alpha_2(\alpha_3 - \alpha_2)(1 - \alpha_2)}, \\
w_3 = \frac{1 - 2\alpha_2}{12\alpha_2(\alpha_3 - \alpha_2)(1 - \alpha_2)}, \\
w_4 = \frac{1}{2} + \frac{2(\alpha_2 + \alpha_3) - 3}{12(1 - \alpha_2)(1 - \alpha_3)},\ \beta_{32} = \frac{\alpha_3(\alpha_3 - \alpha_2)}{2\alpha_2(1 - 2\alpha_2)},\ \alpha_4 = 1 \\
\beta_{42} = \frac{(1 - \alpha_2)\left[\alpha_2 + \alpha_3 - 1 - (2\alpha_3 - 1)^2\right]}{2\alpha_2(\alpha_3 - \alpha_2)\left[6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3\right]}, \\
\beta_{43} = \frac{(1 - 2\alpha_2)(1 - \alpha_2)(1 - \alpha_3)}{\alpha_3(\alpha_3 - \alpha_2)\left[6\alpha_2\alpha_3 - 4(\alpha_2 + \alpha_3) + 3\right]}.
\end{gathered} \qquad (4)$$

If we substitute $\alpha_2 = \alpha_3 = \frac{1}{2}$ into equations (4), we obtain general form of Runge – Kutta method of fourth order as follows:

$$Y_j = Y_{j-1} + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \tag{5}$$

where:

$$\begin{aligned}
k_1 &= f\left(x_{j-1}, Y_{j-1}, \ldots, {}^{n}Y_k\right),\\
k_2 &= f\left(x_{j-1} + \frac{\Delta t}{2}, Y_{j-1} + \frac{\Delta t}{2}.k_1, \ldots, Y_k + \frac{\Delta t}{2}.k_1\right),\\
k_3 &= f\left(x_{j-1} + \frac{\Delta t}{2}, Y_{j-1} + \frac{\Delta t}{2}.k_2, \ldots, Y_k + \frac{\Delta t}{2}.k_2\right),\\
k_4 &= f\left(x_{j-1} + \Delta t, Y_{j-1} + \Delta t.k_3, \ldots, Y_k + \Delta t.k_3\right).
\end{aligned} \tag{6}$$

It is also possible to realize the solution when we use this relation:

$$y_{i+1} = y_i + \left(a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4\right)h \tag{7}$$

The value $y = y_i$ for x_i, it holds that $y = y_{i+1}$ for x_{i+1}, ad $h = x_{i+1} - x_i$.
The equation (7) can be written through the Taylor series into form:

$$\begin{aligned}
y_{i+1} = y_i &+ \frac{dy}{dx}\Big|_{x_i, y_i}\left(x_{i+1} - x_i\right) + \frac{1}{2!}\frac{d^2 y}{dx^2}\Big|_{x_i, y_i}\left(x_{i+1} - x_i\right)^2\\
&+ \frac{1}{3!}\frac{d^3 y}{dx^3}\Big|_{x_i, y_i}\left(x_{i+1} - x_i\right)^3 + \frac{1}{4!}\frac{d^4 y}{dx^4}\Big|_{x_i, y_i}\left(x_{i+1} - x_i\right)^4
\end{aligned} \tag{8}$$

When we use known relation $\frac{dy}{dx} = f(x, y)$, $x_{i+1} - x_i = h$ and substitute it into the equation (8) we get:

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{1}{2!}f^{'}(x_i, y_i)h^2 + \frac{1}{3!}f^{''}(x_i, y_i)h^3 + \frac{1}{4!}f^{'''}(x_i, y_i)h^4 \tag{9}$$

By modification of equation (9) we obtain final form of the method, which is identical with the relation (5, 6).

RESULTS AND DISCUSSION

Technical functions

Technical functions were obtained from experimental manoeuvres of universal tool carrier MT8-222 Synona. Drive manoeuvre was the ride on the slope along the down-grade slope with 45 degree yaw angle with turning to the down-grade slope. The record of centre of gravity velocities of the vehicle is shown in Fig. 1

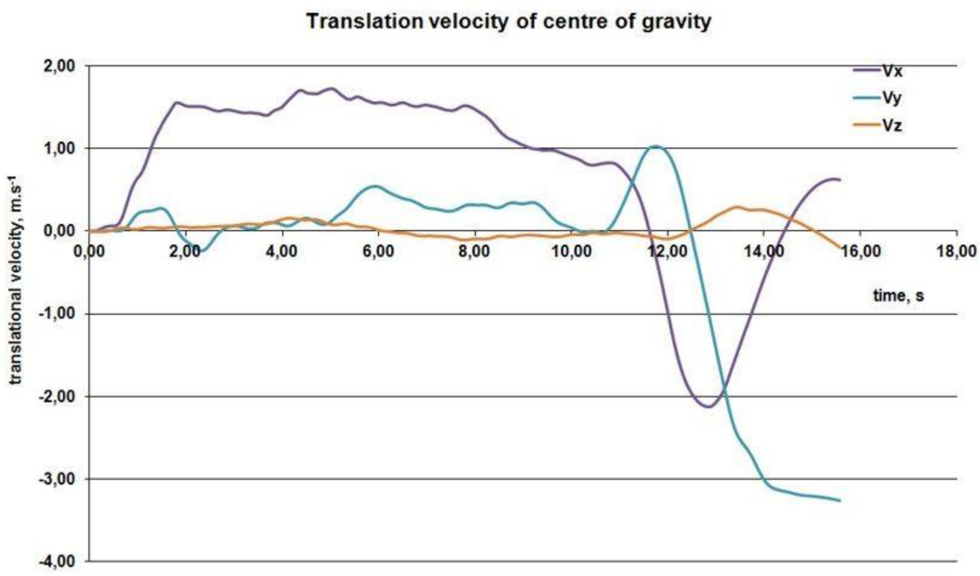

Fig. 1 Record of velocities. Source: Authors

Implementation of Runge-Kutta 4-th order with C#

Algorithm of numerical integration was implemented into computer language C#. We used standard models of object programming. Solitary function values with the needed parameters for evaluation are contained in the structure RK_DATA_STRUCT.

```
public struct RK_DATA_STRUCT //Definition of the structure
    {
      public DataTable RK_Table; //object Table
      public double RK_Step; //step of numerical integration
      public RK_Order_Enum RK_Order; //order of the method
    }

public enum RK_Order_Enum //Definition of enumerator
     {
       /// Classic, 4-th degree
       RK_4_1,
        .

    .
    }
```

Solitary evaluation of numerical integration is included in function Runge_Kutta, where input and output parameter of the function is defined by structure RK_DATA_STRUCT.

```
public RK_DATA_STRUCT Runge_Kutta(RK_DATA_STRUCT dti)
{

for(int i=1;i<=icount-1;i++)
    {
     rk_arr_out[i,j]= rk_arr_out[i-1,j]+dti.RK_Step;
     k1=rk_arr[i-1,j];
     k2=rk_arr[i-1,j]+(rk_arr[i,j]-rk_arr[i-1,j])/2d;
     k3=k2;
     k4=rk_arr[i,j];
```

```
      rk_arr_out[i,j]=rk_arr_out[i-1,j]+((dti.RK_Step/6d)
                         *(k1+(2d*k2)+(2d*k3)+k4));
   }
return rk_arr_out;
}
```

There was chosen following step of numerical integration: $\Delta t = \frac{1}{56}$.

Dislocations of centre of gravity towards inertial coordinate system are the result of numerical integration of technical exciting function of translational velocity of centre of gravity. The resulting functions are depicted in Fig. 2.

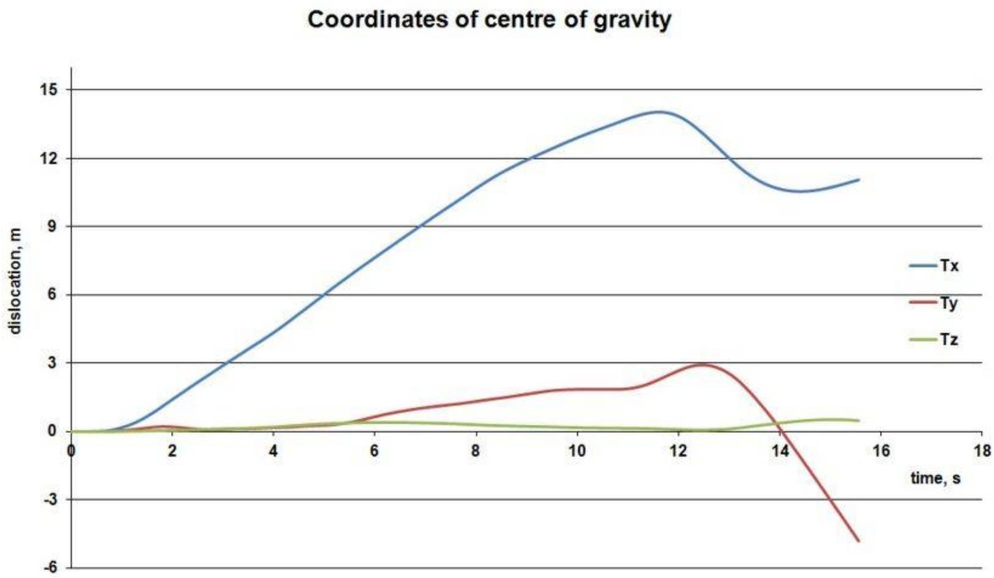

Fig. 2 Dislocation of centre of gravity. Source: Authors

CONCLUSIONS

Processing and experimental data holding in the conditions of science and research are essential factors, which in considerable rate influence the successfulness of scientific – research purpose. In the contribution we present analysis of the Runge Kutta method of fourth order for numerical integration of continuous technical functions with the constant step of numerical integration such as numerical solution of the system of simultaneous differential equations. We introduce algorithm of numerical integration implemented by programming language Microsoft Visual C#. As the demonstration of data processing we present the processing of real values of technical exciting functions of the centre of gravity velocities of off-road technologic vehicle and their visualisation.

REFERENCES

[1] Carnahan, B., Luther, H. A., Wilkes, O.J. (1990). *Applied Numerical Methods.* Washington: Krieger Pub.

[2] Cash, J.R., Girdlestone, S. (2006). *Variable step Runge-Kutta-Nyström methods for the numerical solution of reversible systems*. London: Imperial College London.

[3] Hoang, N.S., Sidje, R. B., Cong, N. H. (2006). On functionally – fitted Runge-Kutta methods. In: *BIT Numerical Mathematics*, vol. 46, pp. 861–874.

[4] Liu, M.Z., Liang, H., Yang, Z.W. (2007). Stability of Runge–Kutta methods in the numerical solution of linear impulsive differential equations. In: *Applied Mathematics and Computation,* vol. 192, pp. 346–357.

[5] Nipp, K. (2002). Numerical integration of differential algebraic systems and invariant manifolds. In: *BIT Numerical Mathematics,* vol. 42(2), pp. 408–439.

[6] Press, W. H. et al. (1997). *Numerical Recipes in C The Art of Scientific Computing*. Cambridge: University Press.

[7] Rédl, J. (2008). Využitie numerickej integrácie Runge-Kutta na spracovanie experimentálnych údajov. In *Technika v technológiách agrosektora 2008*, pp. 79-86.

[8] Rédl, J. (2010). Vývoj technických aplikácií pre vedu a výskum. Nitra: SPU. 146 p.

Reviewed by

1. doc. Ing. Ján Jobbágy, PhD, Slovak University of Agriculture, Faculty of Engineering, Department of Machinery and Production Systems, Tr. A. Hlinku 2, 949 76 Nitra, Slovak Republic
2. Ing. Rastislav Mikuš, PhD. Slovak University of Agriculture, Faculty of Engineering, Department of Quality and Engineering, Tr. A. Hlinku 2, 949 76 Nitra, Slovak Republic