

Investigation of the Strategy of Automatic Step Size Control for the Method Dormand-Prince of Solving Ordinary Differential Equations

Ivan Khvyshchun

Ivan Franko National University of Lviv,
Faculty of Electronics and Computer Technologies,
Lviv, Ukraine, Tarnavskoho str 107,
e-mail: xiocompan@gmail.com

Bogdan Kutnyk

Ivan Franko National University of Lviv,
Faculty of Electronics and Computer Technologies,
Lviv, Ukraine, Tarnavskoho str 107,
e-mail: bogdankutnyk@ukr.net

Abstract—Automatic design of technical devices, mathematical modelling of dynamical routines is important and complex operation. This task solves the problem of behavior of the device in time based on preset first state and internal structure defined by mathematical model.

Index Terms—mathematical modelling, Runge-Kutta methods, ordinary differential equations, step size control.

If the model of the device is the system of common differential equations in normal Coushi form, then implicit or explicit integration method is selected based on the properties of the model [1]. One of the powerful explicit integration method is one step multi stage method of Dormand-Prince. Automatic selection of integration step for mathematical model equations allows to minimize CPU consumption while preserving the precision of calculations.

In digital analysis Dormand-Prince method is explicit one-step multistage method for solving the systems of common differential equations. The method belongs to Runge-Kutta methods family. It consists of seven stages, but it uses only six estimates of right part of the system function on the each step of integration. It has the property "first like last": the last stage is evaluated the same time together with the first stage of the next step. Six evaluations of functions are used for calculation of current values of the solution of fourth and fifth grades. Difference between the evaluations is named the divergence of the solution (fourth grade).

The authors of the method had chosen their method coefficients for minimization of divergence of solution of fifth grade. This is the main difference between Dormand-Prince and Felberg method. Felberg method is designed to minimize divergence of fourth grade. This is why Dormand-Prince method is more suitable, when the solution of higher grade is used for continuation of integration – local extrapolation.

Dormand-Prince method is designed to have "the method of adaptive step" – procedure choosing optimal value of step. Each step of integration [3] provides two different values to be compared. If these two values are greater than defined precision, then the value of step is decreased. If the difference is less then defined precision, the value of step is increased, optimizing integration process.

On the k -th step of integration with Dormand-Prince method we have the following equations.

Given $y = f(t, y)$ [4, 5, 6]:

$$k_1 = h \cdot f(t_k, y_k);$$

$$k_2 = h \cdot f(t_k + h/5, y_k + k_1/5);$$

$$k_3 = h \cdot f(t_k + 3 \cdot h/10, y_k + 3 \cdot k_1/40 + 9 \cdot k_2/40);$$

$$k_4 = h \cdot f(t_k + 4 \cdot h/5, y_k + 44 \cdot k_1/45 - 56 \cdot k_2/15 + 32 \cdot k_3/9);$$

$$k_5 = h \cdot f(t_k + 8 \cdot h/9, y_k + 19372 \cdot k_1/6561 - 25360 \cdot k_2/2187 + 64448 \cdot k_3/6561 - 212 \cdot k_4/729);$$

$$k_6 = h \cdot f(t_k + h, y_k + 9017 \cdot k_1/3168 - 355 \cdot k_2/33 - 46732 \cdot k_3/5247 + 49 \cdot k_4/176 - 5103 \cdot k_5/18656);$$

$$k_7 = h \cdot f(t_k + h, y_k + 35 \cdot k_1/384 + 500 \cdot k_3/1113 + 125 \cdot k_4/192 + 2187 \cdot k_5/6784 - 11 \cdot k_6/84);$$

h – value of integration step, t_k – argument of function y_k on k -th integration step, $k_1 \dots k_7$ – coefficients, used for calculation of the next integration point. Next we calculate the point on $(k+1)$ -th step using Runge-Kutta method of 4-th degree of precision:

$$y_{k+1} = y_k + 35/384 \cdot k_1 + 500/1113 \cdot k_3 + 125/192 \cdot k_4 - 2187/6784 \cdot k_5 + 11/84 \cdot k_6$$

Next is the point on $(k+1)$ -th step using Runge Kutta method of 5-th degree:

$$z_{k+1} = y_k + 5179/57600 \cdot k_1 + 7571/16695 \cdot k_3 + 393/640 \cdot k_4 - 92067/339200 \cdot k_5 + 22/525 \cdot k_6 + 1/40 \cdot k_7,$$

y_{k+1} – value calculated with Runge-Kutta method of 4-th degree on $(k+1)$ -th step, z_{k+1} – value calculated with Runge-Kutta method of 5-th degree on $(k+1)$ -th integration step [5].

Divergence on $(k+1)$ -th integration step is calculated as absolute value of the difference:

$$E_{k+1} = |z_{k+1} - y_{k+1}|.$$

Following we compare this value with defined precision (by default relative error is $\varepsilon_{rel} = 10^{-3}$, absolute error $\varepsilon_{tol} = 10^{-6}$ correction of integration step h is done using the condition:

$$\varepsilon_{n+1} \leq \max(\varepsilon_{rel} \cdot y_k, \varepsilon_{tol}).$$

Correction on each step is done using the figure [2]:

$$h_{opt} = ((\varepsilon \cdot h) / (2 \cdot |z_{k+1} - y_{k+1}|))^{1/5} \cdot h.$$

ε – pre defined integration precision, h – current integration step, h_{opt} – optimal step for finding the next point.

Increasing the precision of absolute or relative error leads to significant increase of CPU time needed for problem solving. That is why in order to increase the precision of function calculation, and to find the optimal quantity of integration steps we propose to use correction method on the step. We have to calculate the value of the function on the current integration step and using the integration step divided by two.

Local error on k -th step is evaluated as [7]:

$$E_{Ti} = 16 (y_{i(h/2)} - y_{i(h)}) / 15,$$

$y_{i(h)}$, $y_{i(h/2)}$ – calculated values of the function using current and divided by two value of integration step.

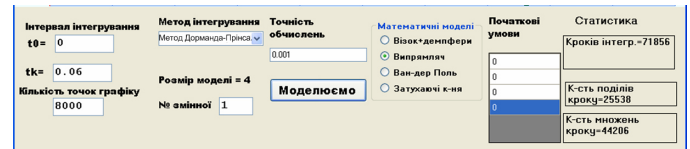
When the error is greater then defined value, integration step is reduced and calculation is repeated. Using this method allows to increase precision of function calculation and to find the proper quantity of integration steps, but the amount of needed calculations is increased.

In general, bigger values of integration step increase the speed of calculation of the model, but it decrease the precision, smaller values – vice versa. Choosing very small values of step h leads to increase of influence of errors caused by rounding. In method with integration step correction, the step is changed in order to get the needed value of precision.

Dormand-Price method was developed in programmatic complex, created in Microsoft Visual Studio 2017 using language C#. Programmatic complex has the open architecture, allowing to add new integration methods and new mathematical models in normal form of Coushi.

Results of modelling can be visualized using charts of every variable of equations of mathematical model. Statistical data about automatic choosing of integration step value is also outputted to the screen.

The following is the user interface of this programmatic complex:



In order to test proposed algorithms we have chosen the systems, mathematical models of which have different dimensions and properties – autonomous, not autonomous, stiff or nonstiff systems. Mathematical models where presented in coordinate basis of state variables. Those models correspond to normal form Coushi, that can be integrated using every known numerical methods [1].

REFERENCES

- [1] I. O. Khvyshchun, Programming and Mathematical Modeling: Handbook – K.: Publishing House "InJure", 2007. (in Ukrainian)
- [2] On Dormand-Prince Method: http://depa.fquim.unam.mx/amyd/archivero/DormandPrince_19856.pdf
- [3] J. R. Dormand, P. J. Prince, "A family of embedded Runge-Kutta formulae", J. Comput. Appl. Math. 1980 vol. 6, pp. 19-26.
- [4] I. V. Olemskoj, Numerical method for integrating systems of ordinary differential equations, "Mathematical Methods of Modeling and Analysis of Controlled Processes", 1986. pp. 157-160. (in Russian)
- [5] I. V. Olemskoj, Embedded five-stage fifth-order method of the Dorman-Prince type. "Journal of Computational Mathematics and Mathematical Physics" 2005. vol. 45. № 7. pp. 81-1191. (in Russian)
- [6] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations. Nonstiff problems. M.: Mir, 1990. (in Russian)
- [7] J. R. Dormand, P. J. Prince, New Runge-Kutta algorithms for numerical simulation in dynamical astronomy "Celestial Mech", 1978, pp. 223-232.