A novel algorithm for solving the ordinary differential equations

Radosław GRZYMKOWSKI, Edyta HETMANIOK
and Mariusz PLESZCZYŃSKI

Abstract. Development of the advanced computational software, including packages for symbolic computations, created new possibilities for solving the initial value problems for ordinary differential equations. The specific methods, that could be considered only theoretically without the possibility of symbolic computations, became particularly important. One of such methods is the method based on one of the fundamental theorems in mathematical analysis, known in the professional literature as *the Taylor formula* [**?**]grzy. Idea of this approach is very simple, but its application requires the calculation of the higher order derivatives of the discussed function which could cause the problems in times when the technique of symbolic computations with the use of digital machines was not yet available. These problems however disappear when we have such computer techniques at our disposal. Obviously, there exist a number of methods suitable for solving problems of the considered kind, like, for example, the whole group of Runge-Kutty methods [6, 7, 8, 9], or the, less known, Adomian decomposition method [1, 4]. In this paper we compare the effectiveness of the investigated method with the effectiveness of the sixth order Runge-Kutty method and the numerical method implemented in *Mathematica* software [2].

Keywords: Taylor transformation, Runge-Kutty method, ordinary differential equation, numerical methods.

2010 Mathematics Subject Classification: 41A58, 65-04, 65L05, 65L06.

1. Theoretical grounds

We consider the problem described by the differential equation

$$y'(x) = f(x, y(x)) \tag{1}$$

R. Grzymkowski, E. Hetmaniok, M. Pleszczyński
Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland,
e-mail: {radoslaw.grzymkowski, edyta.hetmaniok, mariusz.pleszczynski}@polsl.pl

R. Wituła, B. Bajorska-Harapińska, E. Hetmaniok, D. Słota, T. Trawiński (eds.), *Selected Problems on Experimental Mathematics*. Wydawnictwo Politechniki Śląskiej, Gliwice 2017, pp. 103–112.

and initial condition

$$y(x_0) = y_0, \tag{2}$$

where the variable x belongs to interval $\langle x_0, x_0 + \delta\rangle$ and the functions $y(x)$ and $f(x, y(x))$ are of class $C_1\langle x_0, x_0+\delta\rangle$ and $C_n\langle x_0, x_0+\delta\rangle\times\langle y_0-\alpha, y_0+\alpha\rangle$, respectively, for x_0, y_0, δ and $\alpha \in \mathbb{R}$ (C_1 denotes the class of continuous functions with the continuous first derivatives and C_n is the class of continuous functions with all the $n-$order partial derivatives continuous).

In view of the taken assumptions, from the Taylor formula we have

$$\begin{aligned} f(x, y(x)) = {} & \sum_{k=0}^{n-1} \frac{F_k(x_0, y(x_0))}{k!}(x - x_0)^k + \\ & + \frac{F_n(x_0 + \theta(x - x_0), y(x_0 + \theta(x - x_0)))}{n!}(x - x_0)^n, \end{aligned} \tag{3}$$

where $\theta \in (0, 1)$ and the functions F_k, $k = 0, 1, \ldots, n$, are described by relations

$$\begin{aligned} F_0(x, y(x)) &= f(x, y(x)), \\ F_k(x, y(x)) &= \left[\frac{d}{dx} F_{k-1}(x, y(x))\right]_{y'=f(x,y(x))}, \quad k = 1, 2, 3, \ldots, n, \end{aligned} \tag{4}$$

so we have

$$\begin{aligned} F_1(x, y(x)) &= \left[\frac{d}{dx} F_0(x, y(x))\right]_{y'(x)=f(x,y(x))} = \\ &= \left[\frac{\partial F_0(x, y(x))}{\partial x} + \frac{\partial F_0(x, y(x))}{\partial y} y'(x)\right]_{y'(x)=f(x,y(x))} = \\ &= \frac{\partial f(x, y(x))}{\partial x} + \frac{\partial f(x, y(x))}{\partial y} f(x, y(x)), \end{aligned}$$

$$\begin{aligned} F_2(x, y(x)) &= \left[\frac{d}{dx} F_1(x, y(x))\right]_{y'(x)=f(x,y(x))} = \\ &= \left[\frac{d}{dx}\left(\frac{\partial f(x, y(x))}{\partial x} + \frac{\partial f(x, y(x))}{\partial y} f(x, y(x))\right)\right]_{y'(x)=f(x,y(x))} = \\ &= \frac{\partial^2 f(x, y(x))}{\partial x^2} + \frac{\partial^2 f(x, y(x))}{\partial x \partial y} f(x, y(x)) + \\ &+ \left(\frac{\partial^2 f(x, y(x))}{\partial y \partial x} + \frac{\partial^2 f(x, y(x))}{\partial y^2} f(x, y(x))\right) f(x, y(x)) + \\ &+ \frac{\partial f(x, y(x))}{\partial y}\left(\frac{\partial f(x, y(x))}{\partial x} + \frac{\partial f(x, y(x))}{\partial y} f(x, y(x))\right) \end{aligned}$$

and so on.

Although the presented calculations do not look inviting, especially for the large values of n, in real, when we may operate with a tool for symbolic computations, like for example the computational platform *Mathematica 8*, determination of the values of functions F_k, $k = 0, 1, \ldots, n$, is very simple. We may use for this purpose *Program 1*, the source code of which is as follows

```
(* Program 1 *)
Program1[f_,x0_,y0_,n_]:=Module[{ty,tpf},
    tpf=Table[f[x,y[x]],{n}];
    Do[tpf[[i]]=D[tpf[[i-1]],x];
    tpf[[i]]=tpf[[i]]/.{y'[x]->tpf[[1]]},{i,2,n}];
    tpf=tpf/.{y[x]->y0,x->x0};
    Print[Grid[Transpose[Join[{{"n",Subscript[F, "n"]}},
    Table[{i,tpf[[i]]},{i,n}]]],
    Dividers->{{False,True},{False,True,False}},
    Dividers->Center]]
  ];
```

The above program, for the given function $f(x, y)$ and condition $y(x_0) = y_0$, determines the values of functions F_k, $k = 0, 1, \ldots, n$ which is illustrated by Example 1.

Example 1.1. *For function $f(x, y) = x^y$, condition $y(1) = 1$ and $n = 12$, referring to Program 1 by using the instructions*

```
f[x_,y_]:=x^y;
Program1[f,1,1,12]
```

we obtain the following values of functions F_k, $k = 0, 1, \ldots, 12$:

k	1	2	3	4	5	6	7	8	9	10	11	12
F_k	1	1	2	6	22	110	596	4144	29932	265620	2405656	25932016

Continuing our considerations we observe that from (1) and (2) the following relation results

$$\begin{aligned} y'(x) = &\sum_{k=0}^{n-1} \frac{F_k(x_0, y_0)}{k!}(x - x_0)^k + \\ &+ \frac{F_n(x_0 + \theta(x - x_0), y(x_0 + \theta(x - x_0)))}{n!}(x - x_0)^n. \end{aligned} \tag{5}$$

Now, if we divide the interval $\langle x_0, x_0 + \delta\rangle$ into m equal parts, that is we execute its discretization according to formula

$$x_i = x_0 + hi, \quad i = 0, 1, , \ldots, m, \tag{6}$$

where $h = \frac{\delta}{m}$, then by integrating both sides of relation (5) in the subinterval $\langle x_0, x_1\rangle$ of interval $\langle x_0, x_0 + \delta\rangle$ we get

$$\begin{aligned} \int_{x_0}^{x_1} y'(t)dt = &\sum_{k=0}^{n-1} \frac{F_k(x_0, y_0)}{k!} \int_{x_0}^{x_1} (t - x_0)^k dt + \\ &+ \int_{x_0}^{x_1} \frac{F_n(x_0 + \theta(t - x_0), y(x_0 + \theta(t - x_0)))}{n!}(t - x_0)^n dt, \end{aligned} \tag{7}$$

and it means that ($y_1 = y(x_1)$):

$$y_1 = y_0 + \sum_{k=0}^{n-1} \frac{F_k(x_0, y_0)}{(k+1)!} h^{k+1} + \int_{x_0}^{x_1} \frac{F_n(x_0 + \theta(t - x_0), y(x_0 + \theta(t - x_0)))}{n!} (t - x_0)^n dt. \tag{8}$$

Since we know, from the assumption, that function f and its partial derivatives are the continuous functions in the closed region under consideration, thus the functions F_n are bounded, that is

$$\forall n \in \mathbb{N}\ \exists M_n \in \mathbb{R}^+\ \forall x \in \langle x_0, x_0 + \delta \rangle : |F_n| \leq M_n,$$

and it means that

$$\left| y_1 - y_0 - \sum_{k=0}^{n-1} \frac{F_k(x_0, y_0)}{(k+1)!} h^{k+1} \right| \leq \frac{M_n h^{n+1}}{(n+1)!}. \tag{9}$$

From relation (9) it results immediately that we can determine the approximate value y_1 of function $y(x)$ with precision $\Delta_1 = \frac{M_n h^{n+1}}{(n+1)!}$. We have

$$y_1 = y_0 + \sum_{k=0}^{n-1} \frac{F_k(x_0, y_0)}{(k+1)!} h^{k+1}. \tag{10}$$

Repeating the procedure leading to formula (10) for each of subintervals $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, \ldots, m-1$, of interval $\langle x_0, x_0 + \delta \rangle$, we receive

$$y_{i+1} = y_i + \sum_{k=0}^{n-1} \frac{F_k(x_i, y_i)}{(k+1)!} h^{k+1}, \quad i = 0, 1, \ldots, m-1, \tag{11}$$

where y_i, $i = 1, \ldots, m$, are the approximate values of function $y(x)$ at the points x_i, $i = 1, \ldots, m$, and these values are determined with the error not exceeding $\Delta_i = \frac{i M_n h^{n+1}}{(n+1)!}$, $i = 1, \ldots, m$. Number n, deciding on the exactness, can be treated as the order of presented method.

2. Examples

To demonstrate the usefulness of discussed method, we present now few examples in which we will solve the problems described by conditions (1) and (2) and for which we will know the analytical solutions (solutions possible to be found in *Mathematica* software as well as the solutions not possible to be found in that way). Solutions obtained with the aid of presented method will be compared with the exact solutions and with the approximate solutions determined by using the classical sixth order Runge-Kutty method or the method available in *Mathematica* software (executed by using the command `NDSolve`).

2.1. Example 1

Let us consider the equation $y' = (x-y)x - y$, for $x \in [0,2]$, with initial condition $y(0) = 1$. The analytical solution of such formulated problem is given by function $y = x - 1 + 2e^{-\frac{1}{2}x(2+x)}$. Dividing the interval $[0,2]$ into $m = 20$ equal parts and taking six terms ($n = 5$) in sum (11) we obtained the results presented in Figure 1. In the left figure there are displayed the solutions: the exact solution (solid line) and the approximate solutions obtained by using the sixth order Runge-Kutty method (squares), the numerical method from *Mathematica* software (diamonds) and the examined method (stars). Right figure presents the comparison of absolute errors Δ received by using all of these methods (with similar notation).

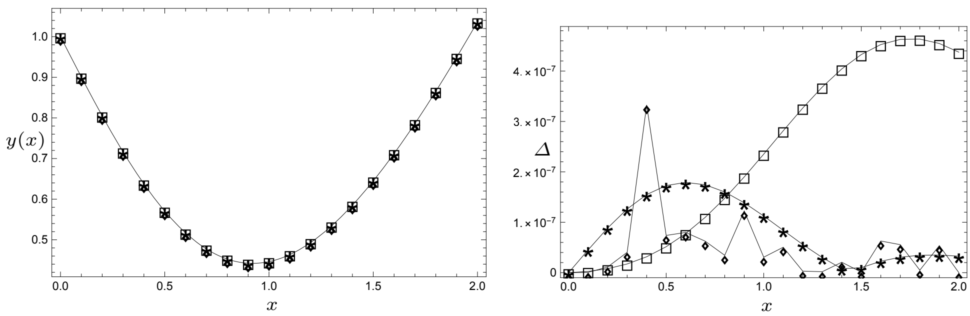

Fig. 1. Solutions: exact one (solid line) and approximate ones from the sixth order Runge-Kutty method ($\square$), *Mathematica* ($\Diamond$) and the examined method ($\bigstar$) (left figure) together with their absolute errors (right figure)

Obviously, by increasing the values of parameters m or n the method should give better results. We show these trends in Figure 2, where the left figure presents the absolute errors of results obtained for increased m and not changed n, in the right figure – inversely – we increased n not changing m.

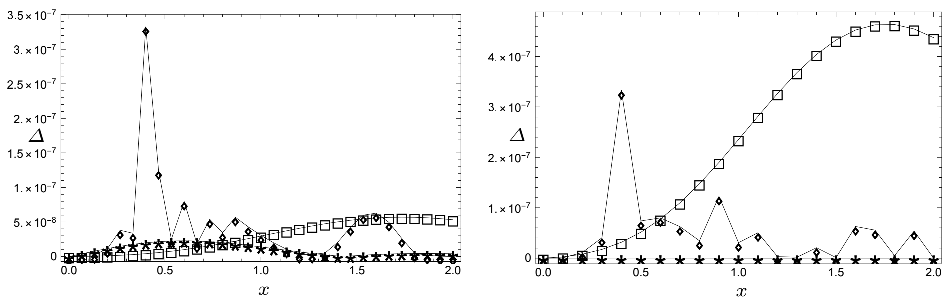

Fig. 2. Absolute errors of results for $m = 30$ and $n = 5$ (left figure) and for $m = 20$ and $n = 7$ (right figure)

As we can see, by increasing the dense of discretization as well as by increasing the order ofvexamined method we get very quickly much better reconstruction of exact solution than the reconstruction obtained by using the Runge-Kutty method or the method implemented in *Mathematica* software.

2.2. Example 2

We examine now the equation $y' = \cos 2x - 2y - 1$, for $x \in [0, 2\pi]$, with initial condition $y(0) = -1$. The discussed problem possesses the following analytical solution: $y = \frac{1}{4}\left(\cos 2x + \sin 2x - 2 - 3e^{-2x}\right)$. We divide the interval $[0, 2\pi]$ into $m = 50$ equal parts and we take $n = 6$ terms in sum (11). The obtained results are displayed in Figure 3, where the left figure presents all four solutions (the exact one and the approximate ones obtained from the Runge-Kutty method, the *Mathematica* numerical method and the investigated method) and the right figure shows the respective absolute errors Δ.

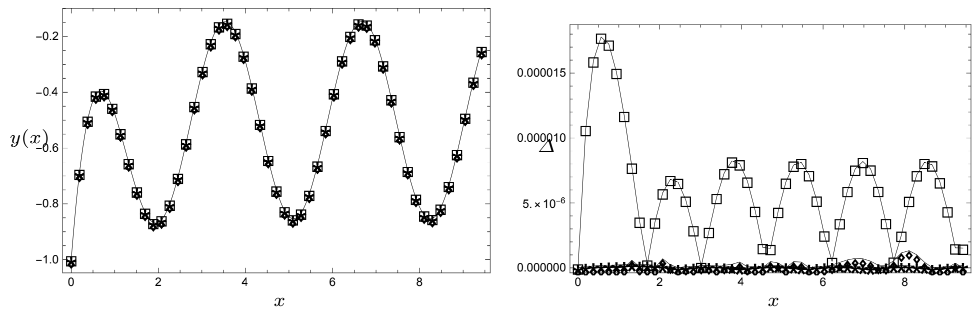

Fig. 3. Solutions: exact one (solid line) and approximate ones from the sixth order Runge-Kutty method (□), *Mathematica* (◊) and the examined method (★) (left figure) together with their absolute errors (right figure)

Since the errors received by applying the Runge-Kutty method are significantly bigger than the other errors, in the next figure we compare only the absolute errors obtained by solving the problem with the aid of *Mathematica* numerical method and the discussed method, first for the same values of n and m as previously (left figure) and next for the value of n increased by one (right figure).

Presented example indicates again the advantage of described method.

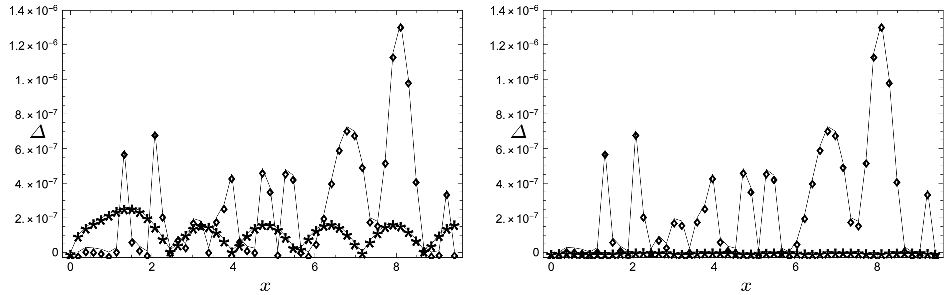

Fig. 4. Absolute errors of results obtained by using the *Mathematica* numerical method (◊) and the examined method (★) for $m = 50$ and $n = 6$ (left figure) and for $m = 50$ and $n = 7$ (right figure)

2.3. Example 3

Let us consider the equation $y' = \frac{1}{2}\sin 2x - \cos x\,(\ln(x+y) - y - x) - 1$, for $x \in [0,8]$, with initial condition $y(0) = 1$. The analytical solution of the problem is of the form $y = e^{\sin x} - x$ and it cannot be determined by *Mathematica* software. Dividing the interval $[0,8]$ into $m = 40$ equal parts and taking $n = 6$ terms in the proper sum we get the results presented in Figure 5. As previously, the left figure includes all the solutions and the right figure shows the comparison of the received absolute errors.

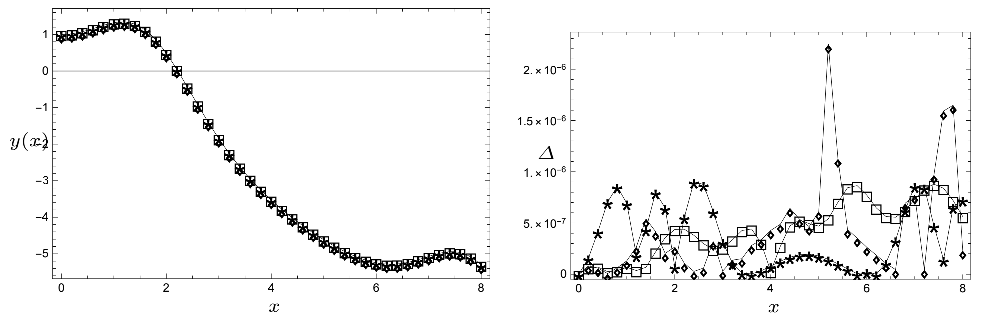

Fig. 5. Solutions: exact one (solid line) and approximate ones from the sixth order Runge-Kutty method (□), *Mathematica* (◇) and the examined method (★) (left figure) together with their absolute errors (right figure)

We can observe that for such selected parameters all the methods give comparable results. And, similarly as in the previous cases, we show now that the increase of the discretization dense or the increase of the method order improves the results. This time the errors obtained by applying the method build in *Mathematica* are clearly bigger that the other errors, therefore in the next figure we compare only the absolute errors received by using the Runge-Kutty method and the discussed method, first for the increased value of m (and not changed n – left figure) and next for the increased value of n (and not changed m – right figure).

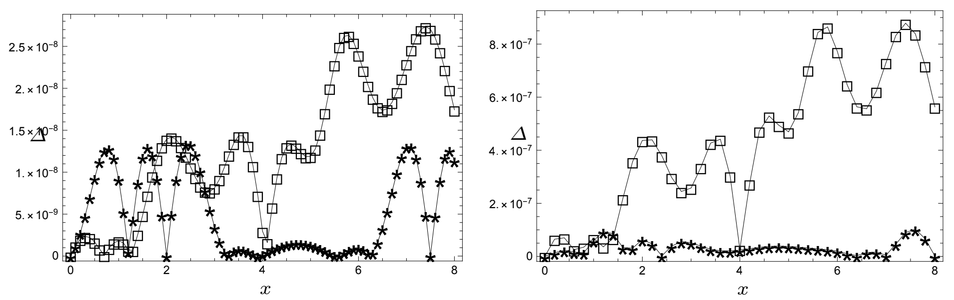

Fig. 6. Absolute errors of results obtained by using the Runge-Kutty method (□) and the examined method (★) for $m = 80$ and $n = 6$ (left figure) and for $m = 40$ and $n = 7$ (right figure)

3. Directions for future research

The works on generalizing this method for the systems of ordinary differential equations are already in progress and the first obtained results are very promising. As a confirmation of this statement let us present the example of applying the generalization of discussed method for the system of two ordinary differential equations (notations for parameters are analogical as in the previous sections). Theoretical introduction for this example can be found in [5], therefore we decided to omit it here.

We consider the system of equations

$$y' = y + z - \sin x + 2(1-x), \quad z' = 2 - (x-2)^2 - y,$$

for $x \in [0,3]$, with conditions $y(0) = 0$, $z(0) = 1$. The analytical solution of this problem is of the form

$$y = \sin x + x(2-x), \quad z = \cos x + x(x-2).$$

Dividing interval $[0,3]$ into $m = 10$ equal parts and including six terms ($n = 5$) in sums analogical to the one defined in (10), we obtained the results presented in Figures 7–9. The first figure shows the solutions: the exact solution (solid lien), the solution received with the aid of the sixth order Runge-Kutty method (squares), the solution obtained by using the numerical method from *Mathematica* software (diamonds) and finally the solution given by the examined method (stars) for function $y(x)$ – left figure and for function $z(x)$ – right figure. In Figure 8 there is displayedthe comparison of absolute errors Δ of the respective approximate solutions (notations and arrangement of the plots are analogical as in the previous sections).

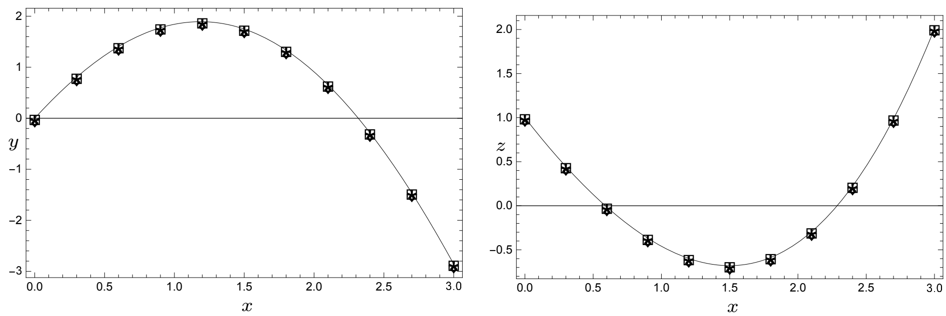

Fig. 7. Solutions: exact one (solid line) and approximate ones from the sixth order Runge-Kutty method (□), *Mathematica* (◊) and the examined method (★) for function $y(x)$ (left figure) and $z(x)$ (right figure)

Since the errors resulting from the application of Runge-Kutty method are significantly higher than the errors from the other methods, we present in Figure 9 the comparison of errors obtained only from these others methods (excluding the Runge-Kutty method).

Obviously, the examined method should give better results by increasing parameter m, as well as n, in formula (11). Let us show this in the next figures, the first one demonstrates the situation with bigger m and not changed n, the second one – with the bigger n and not changed m.

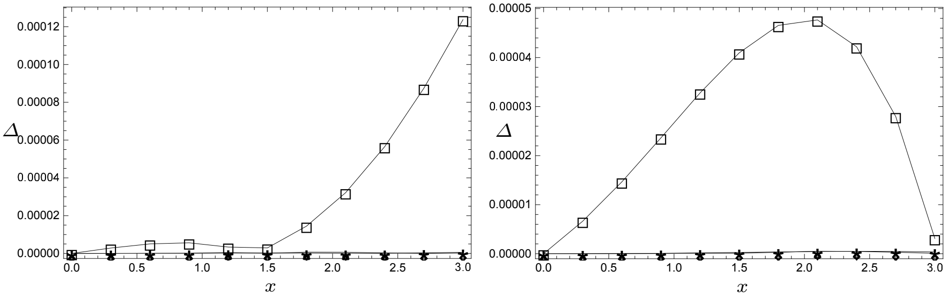

Fig. 8. Absolute errors of results for $m = 10$ and $n = 5$ for function $y(x)$ (left figure) and $z(x)$ (right figure)

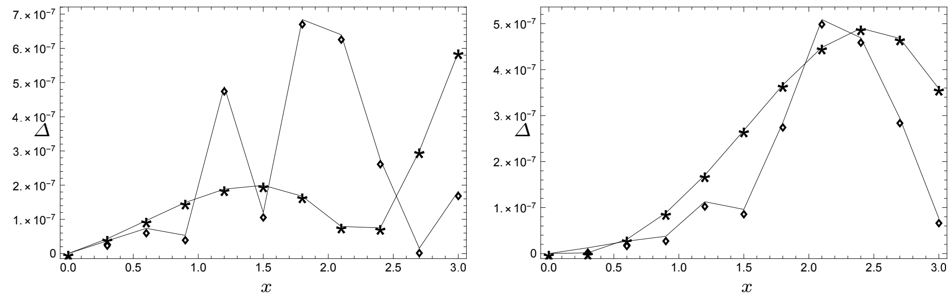

Fig. 9. Absolute errors of results obtained by using the *Mathematica* numerical method (◊) and the examined method (★) for $m = 10$ and $n = 5$ for function $y(x)$ (left figure) and $z(x)$ (right figure)

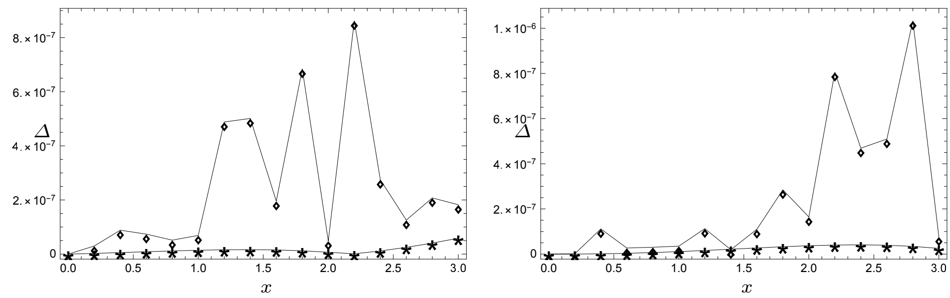

Fig. 10. Absolute errors of results obtained by using the *Mathematica* numerical method (◊) and the examined method (★) for $m = 15$ and $n = 5$ for function $y(x)$ (left figure) and $z(x)$ (right figure)

We observe that in both cases, that is by increasing the dense of discretization as well as by increasing the order of proposed method, we obtain very quickly the exact solution reconstruction of much better quality than the reconstruction obtained by using the Runge-Kutty method or the method implemented in *Mathematica* software.

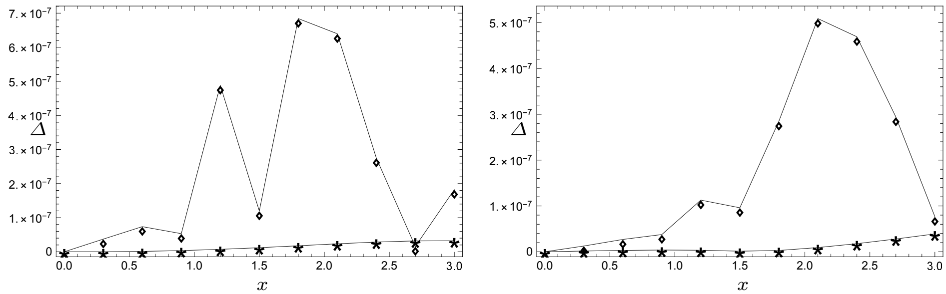

Fig. 11. Absolute errors of results obtained by using the *Mathematica* numerical method ($\Diamond$) and the examined method ($\bigstar$) for $m = 10$ and $n = 7$ for function $y(x)$ (left figure) and $z(x)$ (right figure)

4. Summary

Presented examples, as well as the other numerous experiments not included in this paper with regard to its limited length, indicate that the discussed method is a very good tool for solving the Cauchy problem described by means of conditions (1) and (2). Although the idea of this approach was known for some time, the method was not developed because of the lack of possibility to determine effectively the derivatives. Only the availability of the computer symbolic computations enabled to use efficiently the proposed procedure. Presented examples revealed not only the usefulness of the discussed method, but also its advantage in comparison with the classical sixth order Runge-Kutty method (and, in consequence, the most commonly used fourth order Runge-Kutty method) and the internal method `NDSolve` from the *Mathematica* software.

Bibliography

1. Adomian D.: *A review of the decomposition method in applied mathematics.* J. Math. Anal. Appl. **135** (1988), 501–544.
2. Gliński H., Grzymkowski R., Kapusta A., Słota D.: *Mathematica 8.* Jacek Skalmierski Computer Studio, Gliwice 2012 (in Polish).
3. Grzymkowski R.: *Taylor Transformation and its Applications.* Jacek Skalmierski Computer Studio, Gliwice 2015 (in Polish).
4. Grzymkowski R., Hetmaniok E., Słota D.: *Selected Computational Methods in the Variational Calculus and in the Differential and Integral Equations.* Jacek Skalmierski Computer Studio, Gliwice 2002 (in Polish).
5. Grzymkowski R., Pleszczyński M.: *A novel algorithm for solving the systems of ordinary differential equations* (in preparation).
6. Jastriebow A., Wcislik M.: *Introduction to the Numerical Methods.* Kielce University of Technology Press, Kielce 2000 (in Polish).
7. Kincaid D., Cheney W.: *Numerical Analysis.* WNT, Warsaw 2005 (in Polish).
8. Legras J.: *Practical Methods of Numerical Analysis.* WNT, Warsaw 1974 (in Polish).
9. Szopa R.: *Numerical Solving of the Ordinary Differential Equations.* Częstochowa University of Technology Press, Częstochowa 2005 (in Polish).