Results in Applied Mathematics 6 (2020) 100104

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

Interval versions for special kinds of explicit linear multistep methods

Andrzej Marciniak [a,b,*], Malgorzata A. Jankowska [c]

[a] Poznan University of Technology, Institute of Computing Science, Piotrowo 2, 60-965 Poznan, Poland
[b] State University of Applied Sciences in Kalisz, Department of Computer Science, Poznanska 201-205, 62-800 Kalisz, Poland
[c] Poznan University of Technology, Institute of Applied Mechanics, Jana Pawła II 24, 60-965 Poznan, Poland

ARTICLE INFO

Article history:
Received 27 December 2019
Received in revised form 19 February 2020
Accepted 4 April 2020
Available online 20 April 2020

Keywords:
Initial value problem
Multistep methods
Explicit interval multistep methods
Floating-point interval arithmetic

ABSTRACT

In classical theory of explicit linear multistep methods there are known special kinds of methods which have less function evaluations (in comparison to other multistep methods) and, nevertheless, they give the same accuracy (order) of the approximations obtained. In this paper for such methods we propose their interval versions. It appears that enclosures to the exact solutions obtained by these methods are better in comparison to interval versions of other multistep methods with the same number of steps. The numerical examples presented show that sometimes these enclosures are even better than those obtained by interval methods based on high-order Taylor series.

1. Introduction

The majority of differential equations occurring in science and engineering, given in the form of initial value problem, cannot be integrated analytically. In these cases we apply approximate methods (see, e.g., [1–4]), which introduce the errors of methods. Realizations of such methods in floating-point arithmetic cause two other kinds of errors: representation errors and rounding errors.

In interval arithmetic, which bases are described, among others, in [5–9], the errors of methods are included in the interval solutions obtained by interval algorithms. Applying interval methods for solving the initial value problem in floating-point interval arithmetic (see, e.g., [10]) we can obtain enclosures to the solutions in the form of intervals which contain all possible numerical errors. These intervals can also include data uncertainties.

During the last decades a lot of interval methods for approximating the initial value problem have been developed. The first method was described by R. E. Moore in 1965 [11]. Other known methods are based on the high-order Taylor series (see, e.g., [12–19]), explicit Runge–Kutta methods [9,20–22], implicit ones [23–27], explicit and implicit multistep methods [9,21,27–34].

Explicit linear multistep methods are interesting due to their simplicity. To use such methods we should have approximations to the exact solution in several starting points (values), and then we can apply recursively multistep formulas based on the numerical approximations of a number of successive steps. There are several possibilities for obtaining the starting values. For example, we can use the Taylor series expansion to the exact solution or any one-step method, e.g., a Runge–Kutta method. It appears that there are some kinds of explicit linear multistep methods which reduce the number of function evaluations, without reducing the accuracies (the orders of methods). In this paper we present interval versions of such methods.

* Corresponding author at: Poznan University of Technology, Institute of Computing Science, Piotrowo 2, 60-965 Poznan, Poland.
E-mail addresses: andrzej.marciniak@put.poznan.pl (A. Marciniak), malgorzata.jankowska@put.poznan.pl (M.A. Jankowska).

https://doi.org/10.1016/j.rinam.2020.100104

In Section 2 we shortly remain conventional explicit multistep methods and a way to obtain them. Here, we give the well-known formulas of great importance for interval versions developed in Section 3. In Section 4 we present numerical examples which show that these interval versions for special kinds of explicit linear multistep methods give better enclosures to the exact solution than other interval multistep methods of the same order. Moreover, in some cases they give better enclosures even then methods based on high-order Taylor series. Some conclusions bring our paper to the end.

2. The conventional explicit multistep methods

For the initial value problem

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \tag{1}$$

where $t \in [0, a]$, $y \in \mathbb{R}$, and $f : [0, a] \times \mathbb{R} \to \mathbb{R}$, by integrating the differential equation between the limits t_{k-l} and t_k, we get

$$y(t_k) = y(t_{k-l}) + \int_{t_{k-l}}^{t_k} f(\tau, y(\tau))\, d\tau. \tag{2}$$

To carry out integration in (2), we approximate $f(\tau, y(\tau))$ by an adequate interpolation polynomial and then we integrate this polynomial.

The explicit multistep methods we can obtain if in (2) we use a polynomial of the degree $n-1$ (see, e.g., [3]). Let us denote such a polynomial by $P(\tau)$, and let

$$P\left(t_{k-j}\right) = f\left(t_{k-j}, y\left(t_{k-j}\right)\right), \quad j = 1, 2, \ldots, n.$$

If we exchange the variable τ for t in such a way that $\tau = t_{n-1} + th$, where $h = t_i - t_{i-1}$ for each $i = k-n+1$, $k-n+2, \ldots, n$, then we can write the polynomial $P(\tau)$ as follows:

$$\begin{aligned} P(t_{k-1} + th) = {} & f(t_{k-1}, y(t_{k-1})) + t\nabla f(t_{k-1}, y(t_{k-1})) \\ & + \cdots + \frac{t(t+1)\ldots(t+n-2)}{(n-1)!} \nabla^{n-1} f(t_{k-1}, y(t_{k-1})), \end{aligned}$$

where ∇ denotes the backward difference operator.[1]

We approximate the integrand in (2) by the polynomial $P(\tau)$, i.e., we substitute

$$f(\tau, y(\tau)) = P(\tau) + r(\tau),$$

where $r(\tau)$ denotes the interpolation error given by

$$r(t_{k-1} + th) = f^{(n)}(\xi(t))\, h^n \frac{t(t+1)\ldots(t+n-1)}{n!},$$

and where $\xi(t)$ is an intermediate point in $[t_{k-n}, t_{k-1}]$. After integration we get

$$y(t_k) = y(t_{k-l}) + h \sum_{j=0}^{n-1} \gamma_j \nabla^j f(t_{k-1}, y(t_{k-1})) + h^{n+1}\varphi_n, \tag{3}$$

where

$$\nabla^j f(t_{k-1}, y(t_{k-1})) = \sum_{m=0}^{j} (-1)^m \binom{j}{m} f(t_{k-1-m}, y(t_{k-1-m})), \tag{4}$$

$$\gamma_0 = l, \quad \gamma_j = \frac{1}{j!} \int_{-l+1}^{1} s(s+1)\ldots(s+j-1)\, ds, \quad j = 1, 2, \ldots, n-1,$$

and where $h^{n+1}\varphi_n$ is an error term. Using (4) we can write (3) in the form

$$y(t_k) = y(t_{k-l}) + h \sum_{j=1}^{n} \beta_{nj} f\left(t_{k-j}, y\left(t_{k-j}\right)\right) + h^{n+1}\varphi_n, \tag{5}$$

where

$$\beta_{nj} = (-1)^{j-1} \sum_{m=j-1}^{n-1} \binom{m}{j-1} \gamma_m, \quad j = 1, 2, \ldots, n.$$

Note that the coefficients γ_j (and hence also β_{nj}) are different for different l.

[1] Given the sequence $\{p_n\}$, $n = 0, 1, \ldots$, the backward difference ∇p_n is defined by $\nabla p_n = p_n - p_{n-1}$ for $n \geq 1$. Higher powers are defined recursively by $\nabla^k p_n = \nabla^{k-1}(\nabla p_n)$.

In conventional explicit multistep methods we omit the error term, and after replacing the unknown values $y(t_{k-n})$, $y(t_{k-n+1}), \ldots, y(t_{k-1})$ with approximations $y_{k-n}, y_{k-n+1}, \ldots, y_{k-1}$ we are given the formula for finding y_k. For $l = 1$ we obtain the n-step Adams–Bashforth methods. Taking $l = 2$ we have the n-step methods of Nyström, and for $l = 4$ – the n-step Milne's methods (see, e.g., [1–3]). The formulas we get with l evens and with l differences in (3) are of particular interest since in these cases it can be seen that coefficient of the lth difference is equal to zero what reduces the number of calculations, and the use of $l-1$ or l differences gives the same accuracy. Such methods for $l = 2, 4$ and 6 are as follows:

$$y_k = y_{k-2} + 2hf(t_{k-1}, y_{k-1}), \tag{6}$$

$$y_k = y_{k-4} + \frac{4h}{3}\left[2f(t_{k-1}, y_{k-1}) - f(t_{k-2}, y_{k-2}) + 2f(t_{k-3}, y_{k-3})\right], \tag{7}$$

$$\begin{aligned} y_k = y_{k-6} + \frac{h}{10}&\left[33f(t_{k-1}, y_{k-1}) - 42f(t_{k-2}, y_{k-2}) + 78f(t_{k-3}, y_{k-3})\right. \\ &\left. -42f(t_{k-4}, y_{k-4}) + 33f(t_{k-5}, y_{k-5})\right]. \end{aligned} \tag{8}$$

These methods are of second, fourth and sixth order, respectively.

In an interval version of explicit multistep methods the error term is very important, since it contains the error of the method. In (3) and (5) we have

$$h^{n+1}\varphi_n = h^{n+1}\frac{1}{n!}\int_{-l+1}^{1} s(s+1)\ldots(s+n-1)f^{(n)}(\xi(t))\,ds, \tag{9}$$

what we write in the form

$$h^{n+1}\varphi_n = h^{n+1}\left[\gamma_n^{*}\psi\left(\eta^{*}, y\left(\eta^{*}\right)\right) + \gamma_n^{**}\psi\left(\eta^{**}, y\left(\eta^{**}\right)\right)\right], \tag{10}$$

where $\psi(\eta, y(\eta)) \equiv f^{(n)}(\eta, y(\eta)) \equiv y^{(n+1)}(\eta)$, η^* and η^{**} are intermediate points (not necessary equal) in the interval considered, and

$$\gamma_n^{*} = \frac{1}{n!}\int_{-l+1}^{0} s(s+1)\ldots(s+n-1)\,ds, \quad \gamma_n^{**} = \frac{1}{n!}\int_{0}^{1} s(s+1)\ldots(s+n-1)\,ds.$$

Let us note that in general we cannot write the error term (10) in the form

$$h^{n+1}\varphi_n = h^{n+1}\gamma_n\psi(\eta, y(\eta)),$$

since the function $s(s+1)\ldots(s+n-1)$ changes sign over the interval $[-l+1, 1]$. Only when $l = 1$, i.e. for the Adams–Bashforth methods, we have $\gamma_n^{*} = 0$.

The title of this paper reflects the fact that the explicit methods obtained from (3) or (5) are the special cases of the linear multistep methods of the form

$$\begin{aligned} y_k &= a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_k y_{k-n} \\ &\quad + h\left(b_0 f(t_k, y_k) + b_1 f(t_{k-1}, y_{k-1}) + \cdots + b_k f(t_{k-n}, y_{k-n})\right), \end{aligned}$$

where h is the constant step-size and $a_1, a_2, \ldots, a_k, b_0, b_1, \ldots, b_k$ are given real constants. The general multistep or k-step method for the solution of (1) may be written as

$$\begin{aligned} y_k &= a_1 y_{k-1} + a_2 y_{k-2} + \cdots + a_k y_{k-n} \\ &\quad + h\Phi\left(t_k, t_{k-1}, \ldots, t_{k-n}, f(t_k, y_k), f(t_{k-1}, y_{k-1}), \ldots, f(t_{k-n}, y_{k-n})\right), \end{aligned}$$

where Φ denotes a function of $n+1$ independent variables t_i and $n+1$ dependent variables $f_i = f(t_i, y_i)$.

3. Explicit interval multistep methods

Let us denote:

- Δ_t and Δ_y — bounded sets in which the function $f(t, y)$, occurring in (1), is defined, i.e.,

$$\Delta_t = \{t \in \mathbb{R} : 0 \le t \le a\}, \quad \Delta_y = \left\{y \in \mathbb{R} : \underline{b} \le y \le \overline{b}\right\},$$

- $F(T, Y)$ — an interval extension of $f(t, y)$,[2]
- $\Psi_n(T, Y)$ — an interval extension of $\psi(t, y(t)) \equiv f^{(n)}(t, y(t)) \equiv y^{(n+1)}(t)$,

[2] An interval extension of the function $f : \mathbb{R} \times \mathbb{R} \supset \Delta_t \times \Delta_y \to \mathbb{R}$ we call a function $F : \mathbb{IR} \times \mathbb{IR} \supset \mathbb{I}\Delta_t \times \mathbb{I}\Delta_y \to \mathbb{IR}$ such that $(t, y) \in (T, Y) \Rightarrow f(t, y) \in F(T, Y)$, where $\mathbb{IR}$ denotes the space of real intervals.

and let us assume:

- the function $F(T,Y)$ is defined and continuous for all $T \subset \Delta_t$ and $Y \subset \Delta_y$,[3]
- the function $F(T,Y)$ is monotonic with respect to inclusion, i.e.,

$$T_1 \subset T_2 \wedge Y_1 \subset Y_2 \Rightarrow F(T_1, Y_1) \subset F(T_2, Y_2),$$

- for each $T \subset \Delta_t$ and for each $Y \subset \Delta_y$ there exists a constant $\Lambda > 0$ such that

$$w(F(T,Y)) \leq \Lambda (w(T) + w(Y)),$$

  where $w(A)$ denotes the width of the interval A,
- the function $\Psi_n(T,Y)$ is defined for all $T \subset \Delta_t$ and $Y \subset \Delta_y$,
- the function $\Psi_n(T,Y)$ is monotonic with respect to inclusion.

Assuming that $y(0) \in Y_0$, and the intervals Y_k such that $y(t_k) \in Y_k$ for $k = 1, 2, \ldots, n-1$ are known, for (5) we propose the following interval version

$$Y_k = Y_{k-l} + h \sum_{j=1}^{n} \beta_{nj} F_{k-j} + h^{n+1} \left(\gamma_n^* \Psi_n + \gamma_n^{**} \Psi_n \right), \tag{11}$$

where $k = n, n+1, \ldots, m$, $h = a/m$, and

$$F_{k-j} = F\left(T_{k-j}, Y_{k-j}\right),$$
$$\Psi_n = \Psi\left(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h] F\left(\Delta_t, \Delta_y\right)\right),$$

and where $\Psi(T,Y)$ denotes an interval extension of $f^{(n)}(t, y(t))$. The additional starting intervals Y_k $(k = 1, 2, \ldots, n-1)$ one can obtain by applying an interval one-step method, for example an interval method of Runge–Kutta type (see, e.g., [9,20,21,23–25]) or interval methods based on the Taylor series (see, e.g., [12–17]).

Let us note that in (11) we cannot write $\left(\gamma_n^* + \gamma_n^{**}\right) \Psi_n$ instead of $\gamma_n^* \Psi_n + \gamma_n^{**} \Psi_n$, because in general $\left|\gamma_n^* + \gamma_n^{**}\right|$ may be different from $\left|\gamma_n^*\right| + \left|\gamma_n^{**}\right|$.

Two important theorems can be proved for the method (11).

Theorem 1. *If $y(0) \in Y_0$ and $y(t_i) \in Y_i$ for $i = 1, 2, \ldots, n-1$, then for the exact solution $y(t)$ of the initial value problem (1) we have*

$$y(t_k) \in Y_k$$

for $k = n, n+1, \ldots, m$, where $Y_k = Y(t_k)$ are obtained from the method (11).

The proof of this theorem for $l = 1$ can be found in [21,29] and for $l = 2$ – in [21,31]. For other values of l the proof is similar.

Theorem 2. *If the intervals Y_k are known for $k = 0, 1, \ldots, n-1$, $t_i = ih \in T_i$ for $i = 0, 1, \ldots, m$, and Y_k for $k = n, n+1, \ldots, m$ are obtained from (11), then*

$$w(Y_k) \leq A \max_{q=0,1,\ldots,n-1} w\left(Y_q\right) + B \max_{j=1,2,\ldots,m-1} w\left(T_j\right) + Ch^n,$$

where $w(X)$ denotes the width of interval X and the constants A, B and C are independent of h.

The proof of Theorem 2 for $l = 1, 2$ one can find in [21,29,31], and for other values of l the proof is similar.

As we mentioned in Section 2, the formulas we get with l evens and with l differences are of particular interest. Interval versions of (6)–(8) are as follows:

$$Y_k = Y_{k-2} + 2hF_{k-1} - \frac{h^3}{12}\left(\Psi_2 - 5\Psi_2\right), \tag{12}$$

$$Y_k = Y_{k-4} + \frac{4h}{3}\left(2F_{k-1} - F_{k-2} + 2F_{k-3}\right) - \frac{h^5}{720}\left(27\Psi_4 - 251\Psi_4\right), \tag{13}$$

$$\begin{aligned} Y_k = Y_{k-6} &+ \frac{h}{10}\left(33F_{k-1} - 42F_{k-2} + 78F_{k-3} - 42F_{k-4} + 33F_{k-5}\right) \\ &- \frac{h^7}{60480}\left(1375\Psi_6 - 19087\Psi_6\right). \end{aligned} \tag{14}$$

[3] The function $F(T,Y)$ is continuous at (T_0, Y_0) if for every $\varepsilon > 0$ there is a positive number $\delta = \delta(\varepsilon)$ such that $d(F(T,Y), F(T_0, Y_0)) < \varepsilon$ whenever $d(T, T_0) < \delta$ and $d(Y, Y_0) < \delta$. Here, d denotes the interval metric defined by $d(X_1, X_2) = \max\{|\underline{X}_1 - \underline{X}_2|, |\overline{X}_1 - \overline{X}_2|\}$, where $X_1 = [\underline{X}_1, \overline{X}_1]$ and $X_2 = [\underline{X}_2, \overline{X}_2]$ are two intervals.

In Section 4 the above methods are compared with other explicit interval methods of the same order. The method (12) (of the second order) is compared with the method

- of Adams–Bashforth type

$$Y_k = Y_{k-1} + \frac{h}{2}\left(3F_{k-1} - F_{k-2}\right) + \frac{5h^3}{12}\Psi_2. \tag{15}$$

The method (13) (of the fourth order) is compared with the methods

- of Adams–Bashforth type

$$Y_k = Y_{k-1} + \frac{h}{24}\left(55F_{k-1} - 59F_{k-2} + 37F_{k-3} - 9F_{k-4}\right) + \frac{251h^5}{720}\Psi_4, \tag{16}$$

- of Nyström type

$$Y_k = Y_{k-2} + \frac{h}{3}\left(8F_{k-1} - 5F_{k-2} + 4F_{k-3} - F_{k-4}\right) - \frac{h^5}{720}\left(19\Psi_4 - 251\Psi_4\right), \tag{17}$$

and the method (14) (of the sixth order) we compare with the methods

- of Adams–Bashforth type

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{1440}&\left(4277F_{k-1} - 7923F_{k-2} + 9982F_{k-3} - 7298F_{k-4}\right. \\ &\left.+2877F_{k-5} - 475F_{k-6}\right) + \frac{19087h^7}{60480}\Psi_6, \end{aligned} \tag{18}$$

- of Nyström type

$$\begin{aligned} Y_k = Y_{k-2} + \frac{h}{90}&\left(297F_{k-1} - 406F_{k-2} + 574F_{k-3} - 426F_{k-4}\right. \\ &\left.+169F_{k-5} - 28F_{k-6}\right) - \frac{h^7}{60480}\left(863\Psi_6 - 19087\Psi_6\right), \end{aligned} \tag{19}$$

- of Milne type

$$\begin{aligned} Y_k = Y_{k-4} + \frac{h}{45}&\left(148F_{k-1} - 186F_{k-2} + 344F_{k-3} - 196F_{k-4}\right. \\ &\left.+84F_{k-5} - 14F_{k-6}\right) - \frac{h^7}{60480}\left(783\Psi_6 - 19087\Psi_6\right). \end{aligned} \tag{20}$$

Because in (11) the function $\Psi_n = \Psi_n(T, Y)$ is an interval extension of $f^{(n)}(t, y(t)) \equiv y^{(n+1)}(t)$, it is useful to have $f^{(n)} = f^{(n)}(t, y(t))$ expressed by its partial derivatives. For $n = 6$ we have

$$\begin{aligned} f^{(6)} = {} & f^{(6)}_{t^6} + 6f^{(6)}_{t^5y}f + 15f^{(6)}_{t^4y^2}f^2 + 20f^{(6)}_{t^3y^3}f^3 + 15f^{(6)}_{t^2y^4}f^4 + 6f^{(6)}_{ty^5}f^5 + f^{(6)}_{y^6}f^6 \\ & + 15\left(f^{(5)}_{t^4y}f + 4f^{(5)}_{t^3y^2}f + 6f^{(5)}_{t^2y^3}f^2 + 4f^{(5)}_{ty^4}f^3 + f^{(5)}_{y^5}f^4\right)f' \\ & + 20\left(f^{(4)}_{t^3y} + 3f^{(4)}_{t^2y^2}f + 3f^{(4)}_{ty^3}f^2 + f^{(4)}_{y^4}f^3\right)f'' \\ & + 45\left(f^{(4)}_{t^2y^2} + 2f^{(4)}_{ty^3}f + f^{(4)}_{y^4}f^2\right)\left(f'\right)^2 + 15\left(f^{(3)}_{t^2y} + 2f^{(3)}_{ty^2}f + f^{(3)}_{y^3}f^2\right)f''' \\ & + 60\left(f^{(3)}_{ty^2} + f^{(3)}_{y^3}f\right)f'f'' + 15f^{(3)}_{y^3}\left(f'\right)^3 + 6\left(f^{(2)}_{ty} + f^{(2)}_{y^2}f\right)f^{(4)} \\ & + 5f^{(2)}_{y^2}\left(3f'f''' + 2\left(f''\right)^2\right) + f^{(1)}_{y}f^{(5)}, \end{aligned} \tag{21}$$

where

$$f' = f^{(1)}_t + f^{(1)}_y f,$$

$$f'' = f^{(2)}_{t^2} + 2f^{(2)}_{ty}f + 2f^{(2)}_{y^2}f^2 + f^{(1)}_y f',$$

$$f''' = f^{(3)}_{t^3} + 3f^{(3)}_{t^2y}f + 3f^{(3)}_{ty^2}f^2 + f^{(3)}_{y^3}f^3 + 3\left(f^{(2)}_{ty} + f^{(2)}_{y^2}f\right)f' + f^{(1)}_y f'',$$

$$f^{(4)} = f^{(4)}_{t^4} + 4f^{(4)}_{t^3y}f + 6f^{(4)}_{t^2y^2}f^2 + 4f^{(4)}_{ty^3}f^3 + f^{(4)}_{y^4}f^4$$

$$
\begin{aligned}
&+6\left(f_{t^2y}^{(3)}+2f_{ty^2}^{(3)}f+f_{y^3}^{(3)}f^2\right)f'+4\left(f_{ty}^{(2)}+f_{y^2}^{(2)}f\right)f'' \\
&+3f_{y^2}^{(2)}\left(f'\right)^2+f_y^{(1)}f''',
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
f^{(5)}=\;&f_{t^5}^{(5)}+5f_{t^4y}^{(5)}f+10f_{t^3y^2}^{(5)}f^2+10f_{t^2y^3}^{(5)}f^3+5f_{ty^4}^{(5)}f^4+f_{y^5}^{(5)}f^5 \\
&+10\left(f_{t^3y}^{(4)}+3f_{t^2y^2}^{(4)}f+3f_{ty^3}^{(4)}f^2+f_{y^4}^{(4)}f^3\right)f' \\
&+10\left(f_{t^2y}^{(3)}+2f_{ty^2}^{(3)}f+f_{y^3}^{(3)}f^2\right)f'' \\
&+15\left(f_{ty^2}^{(3)}+f_{y^3}^{(3)}f\right)\left(f'\right)^2+5\left(f_{ty}^{(2)}+f_{y^2}^{(2)}f\right)f''' \\
&+10f_{y^2}^{(2)}f'f''+f_y^{(1)}f^{(4)},
\end{aligned}
$$

and where we denote

$$
f=f\left(t,y\right),\quad f_{t^py^q}^{(l)}=\frac{\partial^l f}{\partial t^p\partial y^q},\quad l=p+q,
$$

to short the notations. Mathematical software (e.g., Derive, Matlab, Mathematica) can be very helpful to find analytical forms of the above derivatives, and then their interval extensions can be easy determined.

4. Numerical examples

In the examples presented below we have used our own implementation of floating-point interval arithmetic in Delphi Pascal. This implementation has been written in the form of a unit called *IntervalArithmetic32and64* (the current version of this unit is presented in [35]). This unit takes advantage of the Delphi Pascal floating-point *Extended* type.[4] All programs written in Delphi Pascal for the examples presented can be found in [36]. In [36] it is also included a Delphi Pascal program for solving any initial value problem by the interval methods considered in this paper. This program requires the user only to write a dynamic link library with definitions of interval functions $F(T, Y)$ and $\Psi_n(T, Y)$ $(n = 2, 4 \text{ or } 6)$ and to determine starting intervals.

The calculations for all examples presented (like for many others considered by us) have been carried out on Lenowo® Z51 computer with Intel® Core i7 2.4 GHz processor using Embarcadero® Delphi XE 10 package. From a lot of examples analyzed by us, ad hoc we choose three ones: a very simple with known exact solution, a more complicated, but also with known exact solution, and an example for which the exact solution is unknown, but the results obtained by us can be compared with ones obtained by the well-known VNODE-LP package [37].

Example 1. Firstly, let us consider the simple initial value problem (the commonly used test problem)

$$
y' = 0.5y,\quad y(0) = 1. \tag{23}
$$

This problem has the exact solution $y = \exp(0.5t)$ and with the precision of 16 digits after decimal point at $t = 1$ we have $y = 1.6487212707001282$. Since for the problem (23) the exact solution is known, the starting intervals for any interval method considered can be obtained directly from this solution (see the program *EIMM_Example_1* in [36] for details). Assuming

$$
\Delta_t = \{t\in\mathbb{R} : 0 \le t \le 1\},\quad \Delta_y = \left\{y\in\mathbb{R} : 1 \le y \le \overline{1.65}\right\},
$$

where $\bar{x}$ denotes the smallest machine number greater or equal to x (similarly, by $\underline{x}$ we will denote the largest machine number less or equal to x), we have tested the methods (12)–(20) for different step sizes. The widths of intervals obtained are presented in Figs. 1–4. We see that the methods (12)–(14) give better results than other interval multistep methods of the same order. Moreover, for each method of particular order there is a step size h which gives the best results (the intervals with the smallest widths). For the second order methods (12) and (15) we find $h = 0.00002$, for the fourth order methods (13), (16) and (17) we have $h = 0.001$, while for the sixth order methods we find $h = 0.01$. For smaller values of these step sizes the numbers of calculations grow what causes a growth of rounding errors, and for larger values of step sizes the errors of methods are increased. In both cases the intervals obtained have larger widths.

It is also interesting that a higher order method with an appropriate step size not necessary gives better results than a method of smaller order (compare the results presented in Tables 1 and 2). We can generalize this note: the order of the method (the number of steps) and the step size should be suitably selected for each problem considered.

[4] In all our papers concerning interval calculations (see, e.g., [21,23,24,26–31]) we use Delphi Pascal floating-point *Extended* type instead of IEEE-754 *Double* type. The *Extended* type has both greater range and greater precision (it uses 10 bytes to store a sign bit, a 15-bit exponent, and a 64-bit mantissa) in comparison to the *Double* type, which uses 8 bytes (a sign bit, a 11-bit exponent, and a 52-bit mantissa).

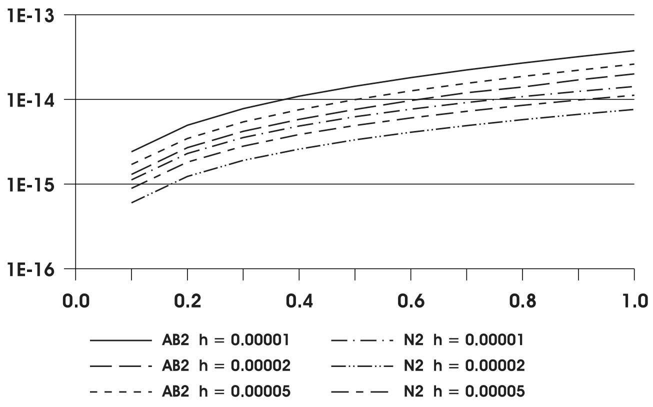

Fig. 1. The widths of interval solutions of (23) obtained by the two-step method (12) (N2) and (15) (AB2).

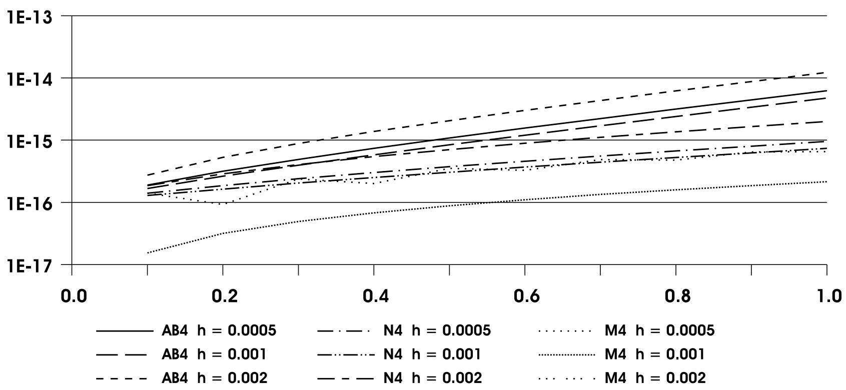

Fig. 2. The widths of interval solutions of (23) for the four-step method (13) (M4), (16) (AB4) and (17) (N4).

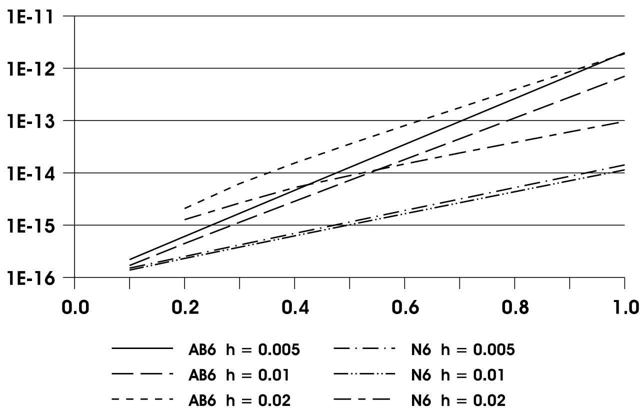

Fig. 3. The widths of interval solutions of (23) for the six-step method (18) (AB6) and (19) (N6).

Table 1
The intervals obtained for the problem (23) at $t = 1$ by the four-step method (13) (M4), (16) (AB4) and (17) (N4) with $h = 0.001$.

Method	Y	Width
AB4	[1.6487212707001258E+0000, 1.6487212707001306E+0000]	$\approx 4.74 \cdot 10^{-15}$
N4	[1.6487212707001278E+0000, 1.6487212707001286E+0000]	$\approx 7.40 \cdot 10^{-16}$
M4	[1.6487212707001280E+0000, 1.6487212707001283E+0000]	$\approx 2.15 \cdot 10^{-16}$

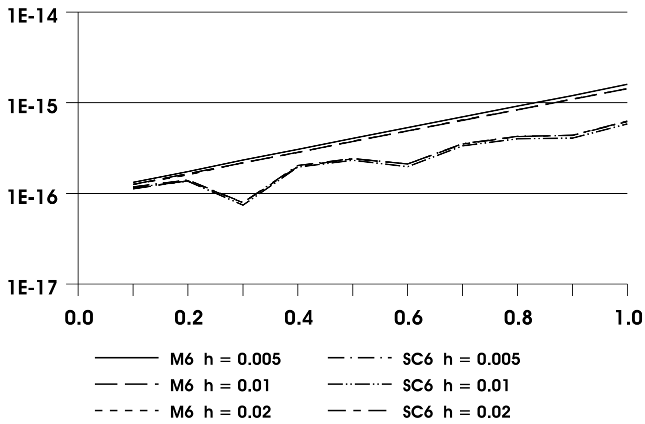

Fig. 4. The widths of interval solutions of (23) for the six-step method (14) (SC6) and (20) (M6).

Table 2
The intervals obtained for the problem (23) at $t = 1$ by the six-step method (14) (SC6), (18) (AB6), (19) (N6) and (20) (M6) with $h = 0.01$.

Method	Y	Width
AB6	[1.6487212706997775E+0000, 1.6487212707004788E+0000]	$\approx 7.01 \cdot 10^{-13}$
N6	[1.6487212707001224E+0000, 1.6487212707001339E+0000]	$\approx 1.14 \cdot 10^{-14}$
M6	[1.6487212707001274E+0000, 1.6487212707001289E+0000]	$\approx 1.43 \cdot 10^{-15}$
SC6	[1.6487212707001278E+0000, 1.6487212707001285E+0000]	$\approx 6.18 \cdot 10^{-16}$

Table 3
The approximate (with 16 digits after decimal point) exact solution to (24).

t	$y(t)$
0.5	1.7425955377077778
1.0	1.7081615480566544
1.5	1.0969902685624423
2.0	0.5409760832487151

Example 2. In the problem considered in Example 1 all partial derivatives occurring in (21)–(22), except of $\partial f/\partial y$ are equal to zero. Thus, let us take into account a more complicated initial value problem

$$y' = \frac{1}{\exp(t/4)}\left(2\cos 2t - \frac{\sin^2 2t}{4y\exp(t/4)} - \frac{\sin 2t}{4y}\right), \quad y(0) = 1, \tag{24}$$

with the exact solution

$$y = 1 + \frac{\sin 2t}{\exp(t/4)}. \tag{25}$$

For $t = 0.5, 1.0, 1.5, 2.0$ the numerical values of the solution are given in Table 3.

Because the methods (13)–(14) give smaller intervals than other multistep methods of the same order, let us consider only them, and let us assume that

$$\Delta_t = \{t \in \mathbb{R} : 0 \le t \le 2\}, \quad \Delta_y = \left\{y \in \mathbb{R} : \underline{0.4} \le y \le \overline{1.9}\right\}.$$

Starting intervals can be easy obtained from (25) (see the program *EIMM_ Example_2* in [36] for details). Analyzing different step sizes we have found that the best step sizes are: $h = 0.0002$ for the method (13), and $h = 0.001$ for the method (14). The intervals obtained are presented in Tables 4 and 5. Of course, in each case the exact solution is within these intervals. The method (14) gives not only smaller intervals, but is also a little bit faster (taking into account the time of computations to achieve $t = 2.0$ — see Fig. 5).

Example 3. Finally, let us consider a problem, for which the exact solution is unknown. Let us take the initial value problem (the problem A5 in [38, p. 23])

$$y' = \frac{y - t}{y + t}, \quad y(0) = 4. \tag{26}$$

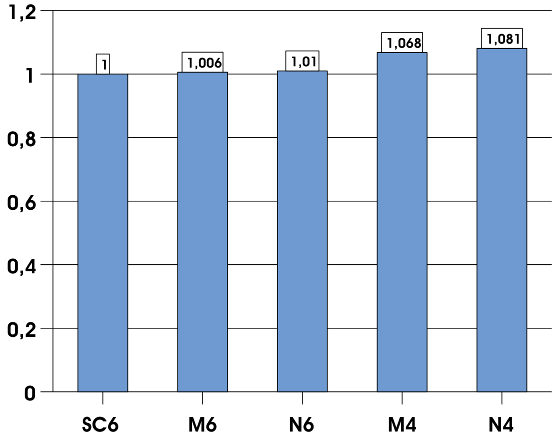

Fig. 5. A comparison of computational times for the methods (13) (M4), (14) (SC6), (17) (N4), (19) (N6) and (20) (M6), and the problem (24) (it is assumed that $t = 1$ unit $\approx$ 2 min 40 sec for the method (14)).

Table 4
The interval solution of (24) obtained by the method (13) with $h = 0.0002$.

$t = kh \in T_k$	Y_k	Width
0.5	[1.7425955377077775E+0000, 1.7425955377077780E+0000]	$\approx 4.00 \cdot 10^{-16}$
1.0	[1.7081615480566534E+0000, 1.7081615480566552E+0000]	$\approx 1.70 \cdot 10^{-15}$
1.5	[1.0969902685624375E+0000, 1.0969902685624471E+0000]	$\approx 9.45 \cdot 10^{-15}$
2.0	[5.4097608324868416E−0001, 5.4097608324874315E−0001]	$\approx 5.90 \cdot 10^{-14}$

Table 5
The interval solution of (24) obtained by the method (14) with $h = 0.001$.

$t = kh \in T_k$	Y_k	Width
0.5	[1.7425955377077776E+0000, 1.7425955377077780E+0000]	$\approx 2.55 \cdot 10^{-16}$
1.0	[1.7081615480566539E+0000, 1.7081615480566548E+0000]	$\approx 8.07 \cdot 10^{-16}$
1.5	[1.0969902685624403E+0000, 1.0969902685624444E+0000]	$\approx 3.99 \cdot 10^{-15}$
2.0	[5.4097608324870344E−0001, 5.4097608324872700E−0001]	$\approx 2.36 \cdot 10^{-14}$

Table 6
Starting intervals for the problem (26) and the methods (13), (14), (16)–(20).

Method	t_k	Y_k
(13), (16), (17)	$t_1 = 0.0005$	$Y_1 = [\,\underline{4.0004999375104142}, \overline{4.0004999375104143}]$
	$t_2 = 0.0010$	$Y_2 = [\,\underline{4.0009997500832942}, \overline{4.0009997500832943}]$
	$t_3 = 0.0015$	$Y_3 = [\,\underline{4.0014994377810524}, \overline{4.0014994377810525}]$
(14), (18)–(20)	$t_1 = 0.002$	$Y_1 = [\,\underline{4.0019990006660423}, \overline{4.0019990006660424}]$
	$t_2 = 0.004$	$Y_2 = [\,\underline{4.0039960053233546}, \overline{4.0039960053233547}]$
	$t_3 = 0.006$	$Y_3 = [\,\underline{4.0059910179495364}, \overline{4.0059910179495365}]$
	$t_4 = 0.008$	$Y_4 = [\,\underline{4.0079840425073461}, \overline{4.0079840425073462}]$
	$t_5 = 0.010$	$Y_5 = [\,\underline{4.0099750829447797}, \overline{4.0099750829447798}]$

Additional starting intervals are presented in Table 6 (these intervals have been obtained by an interval version of conventional fourth order method of Runge–Kutta type [9,21,24] for step sizes guaranteeing the tiniest intervals). For

$$\Delta_t = \{t \in \mathbb{R} : 0 \le t \le 1\}, \quad \Delta_y = \left\{y \in \mathbb{R} : 4 \le y \le \overline{4.9}\right\},$$

the results obtained by the methods (13), (14), (16)–(20) at $t = 1$ are shown in Table 7. These results can be compared with one obtained by the VNODE-LP package with an interval method based on high-order Taylor series [37]. This package produces the output

4.8075923778847[027, 117],

and we see that our low-order multistep methods, except of (16) and (18), give comparable better results.

Table 7
The interval solutions obtained by the methods (13), (14), (16)–(20) at $t = 1.0$.

Method	h	Y	Width
(16)	0.0005	[4.8075923778847016E+0000, 4.8075923778847110E+0000]	$\approx 9.28 \cdot 10^{-15}$
(17)	0.0005	[4.8075923778847052E+0000, 4.8075923778847073E+0000]	$\approx 2.05 \cdot 10^{-15}$
(13)	0.0005	[4.8075923778847059E+0000, 4.8075923778847067E+0000]	$\approx 7.53 \cdot 10^{-16}$
(18)	0.002	[4.8075923778844714E+0000, 4.8075923778849412E+0000]	$\approx 4.70 \cdot 10^{-13}$
(19)	0.002	[4.8075923778847026E+0000, 4.8075923778847099E+0000]	$\approx 7.13 \cdot 10^{-15}$
(20)	0.002	[4.8075923778847056E+0000, 4.8075923778847068E+0000]	$\approx 1.09 \cdot 10^{-15}$
(14)	0.002	[4.8075923778847060E+0000, 4.8075923778847066E+0000]	$\approx 5.26 \cdot 10^{-16}$

5. Conclusions

Interval methods executed in floating-point interval arithmetic yield approximations of solutions in the form of intervals, which contain all possible numerical errors. Explicit interval multistep methods are simple, what causes that their execution time is short. Among them are methods of special kinds which reduce the number of function evaluations. Numerical examples show that such methods give better enclosures to the exact solutions in comparison to other multistep methods with the same number of steps. Such methods are also competitive to interval methods based on high-order Taylor series.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Andrzej Marciniak: Conceptualization, Methodology, Software, Writing - original draft. **Malgorzata A. Jankowska:** Software, Validation, Writing - review & editing.

Acknowledgments

A. Marciniak was supported by the Poznan University of Technology (Poland) through the Grant No. 09/91/DSPB/0600. M.A. Jankowska was supported by the Poznan University of Technology (Poland) through the Grant No. 02/21/DSPB/3477.

References

[1] Butcher JC. The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods. Chichester: John Wiley & Sons; 1987.
[2] Hairer E, Norsett SP, Wanner G. Solving ordinary differential equations I – Nonstiff problems. Berlin: Springer–Verlag; 1987.
[3] Jain MK. Numerical solution of differential equations. New York: John Wiley & Sons; 1979.
[4] Milne WE. Numerical solution of differential equations. New York: Johm Wiley & Sons; 1953.
[5] Alefeld G, Herzberger J. Introduction to interval computations. New York: Academic Press; 1983.
[6] Hansen ER. Topics in interval analysis. London: Oxford University Press; 1969.
[7] Moore RE. Interval analysis. Englewood Cliffs: Prentice-Hall; 1966.
[8] Moore RE. Methods and applications of interval analysis. Philadelphia: SIAM Society for Industrial & Applied Mathematics; 1979.
[9] Shokin YI. Interval analysis. Novosibirsk: Nauka; 1981.
[10] Hammer R, Hocks M, Kulisch U, Ratz D. Numerical toolbox for verified computing I. Basic numerical problems, theory, algorithms, and pascal-XSC programs. Berlin: Springer-Verlag; 1993.
[11] Moore RE. The automatic analysis and control of error in digital computation based on the use of interval numbers. In: Rall LB, editor. Error in digital computation, Vol. 1. New York: John Wiley & Sons; 1965, p. 61–130.
[12] Berz M, Hoffstätter G. Computation and application of Taylor polynomials with interval remainder bounds. Reliab Comput 1998;4(1):83–97.
[13] Berz M, Makino K. Performance of Taylor model methods for validated integration of ODEs. In: Dongarra J, Madsen K, Wasniewski J, editors. Applied parallel computing. State of the art in scientific computing. Lecture notes in computer science, vol. 3732, 2005, p. 65–73.
[14] Corliss GF, Rihm R. Validating an a priori enclosure using high-order Taylor series. In: Scientific computing, computer arithmetic, and validated numerics. Akademie Verlag; 1996, p. 228–38.
[15] Jackson KR, Nedialkov NS. Some recent advances in validated methods for IVPs for ODEs. Appl Numer Math 2002;42(1–3):269–84.
[16] Lohner RJ. Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In: Cash JR, Gladwell I, editors. Computational ordinary differential equations. Oxford: Clarendon Press; 1992, p. 425–35.
[17] Nedialkov NS, Jackson KR, Corliss GF. Validated solutions of initial value problems for ordinary differential equations. Appl Math Comput 1999;105(1):21–68.
[18] Nedialkov NS. Interval tools for ODEs and DAEs. Tech. Rep. CAS 06-09-NN, Department of Computing and Software, McMaster University, Hamilton; 2006.
[19] Nickel K. Using interval methods for the numerical solution of ODE's. ZAMM-J Appl Math Mech/Z Angew Math Mech 1986;66(11):513–23.
[20] Kalmykov SA, Shokin JI, Juldashev EC. Solving ordinary differential equations by interval methods. In: Doklady an SSSR, Vol. 230. 1976.

[21] Marciniak A. Selected Interval Methods for Solving the Initial Value Problem. Poznan: Publishing House of Poznan University of Technology; 2009, http://www.cs.put.poznan.pl/amarciniak/IMforIVP-book/IMforIVP.pdf.
[22] Marciniak A, Szyszka B. Interval Runge-Kutta methods with variable step sizes. Comput Methods Sci Technol 2019;25(1):33–46.
[23] Gajda K, Marciniak A, Szyszka B. Three-and four-stage implicit interval methods of Runge-Kutta type. Comput Methods Sci Technol 2000;6(1):41–59.
[24] Gajda K, Jankowska M, Marciniak A, Szyszka B. A survey of interval Runge-Kutta and multistep methods for solving the initial value problem. In: Wyrzykowski R, Dongarra J, Karczewski K, Wasniewski J, editors. Parallel processing and applied mathematics. Lecture notes in computer science, vol. 4967, Berlin: Springer-Verlag; 2008, p. 1361–71.
[25] Marciniak A, Szyszka B. One-and two-stage implicit interval methods of Runge-Kutta type. Comput Methods Sci Technol 1999;5(1):53–65.
[26] Marciniak A. Implicit interval methods for solving the initial value problem. Numer Algorithms 2004;37(1–4):241–51.
[27] Marciniak A. On multistep interval methods for solving the initial value problem. J Comput Appl Math 2007;199(2):229–37.
[28] Jankowska M, Marciniak A. Implicit interval methods for solving the initial value problem. Comput Methods Sci Technol 2002;8(1):17–30.
[29] Jankowska M, Marciniak A. On explicit interval methods of adams-bashforth type. Comput Methods Sci Technol 2002;8(2):46–57.
[30] Jankowska M, Marciniak A. On two families of implicit interval methods of Adams-Moulton type. Comput Methods Sci Technol 2006;12(2):109–13.
[31] Marciniak A. Multistep interval methods of Nyström and Milne-Simpson types. Comput Methods Sci Technol 2007;13(1):23–40.
[32] Marciniak A, Jankowska MA, Hoffmann T. On interval predictor-corrector methods. Numer Algorithms 2017;75(3):777–808.
[33] Marciniak A, Jankowska MA. Interval methods of Adams-Bashforth type with variable step sizes. Numer Algorithms 2019. http://dx.doi.org/10.1007/s11075-019-00774-y.
[34] Marciniak A, Jankowska MA. Interval versions of Milne's multistep methods. Numer Algorithms 2018;79(1):87–105.
[35] Marciniak A. Interval arithmetic unit. 2016, http://www.cs.put.poznan.pl/amarciniak/IAUnits/IntervalArithmetic32and64.pas.
[36] Marciniak A. Delphi pascal programs for special kinds of explicit linear multistep methods. 2018, http://www.cs.put.poznan.pl/amarciniak/EIMM-Examples.
[37] Nedialkov NS. VNODE-LP – A validated solver for initial value problems in ordinary differential equations. Tech. Rep. CAS 06-06-NN, Department of Computing and Software, McMaster University, Hamilton; 2006.
[38] Enright WH, Pryce JD. Two fortran packages for assessing initial value methods. ACM Trans Math Software 1987;13(1):1–27.