Andrzej Marciniak

Selected Interval Methods for Solving the Initial Value Problem

Poznań 2009

Publishing House of Poznań University of Technology

Presented by
TOMASZ SZULC

English proof-reading
MICHAŁ JANKOWSKI

Technical Editor
RENATA LUBAWY

Cover design
ANDRZEJ MARCINIAK

Typesetting with WordPerfect® Office X3 by
ANDRZEJ MARCINIAK

ISBN 978-83-7143-823-3

PUBLISHING HOUSE OF POZNAŃ UNIVERSITY OF TECHNOLOGY
pl. M. Skłodowskiej-Curie 2, 60-965 Poznań, Poland
tel. +48 (0) 61 665 3516; fax +48 (0) 61 665 3583
e-mail: office_ed@put.poznan.pl
www.ed.put.poznan.pl

Orders should be sent to:
Poznańska Księgarnia Akademicka
ul. Piotrowo 3, 61-138 Poznań, Poland
tel. +48 (0) 61 665 2324; fax +48 (0) 61 665 2326
e-mail: politechnik@politechnik.poznan.pl
www.politechnik.poznan.pl

Printed by:
Prymasowskie Wydawnictwo Gaudentinum – Drukarnia Cyfrowa
www.gaudentinum.pl

Contents

Introduction

Many scientific and engineering problems are described in the form of ordinary differential equations. If such equations cannot be solved analytically, we use computers and approximate methods (algorithms) to solve them, usually providing all calculations in floating-point arithmetic. This arithmetic is a fast way to perform calculations, but it is also error-prone and therefore particular attention must be paid to the reliability of the computed results.

There are two kinds of errors caused by floating-point arithmetic: representation errors and rounding errors. From the mathematical point of view, a real number is represented by an infinite decimal or a binary fraction. In a computer such a real number must be approximated by a finite fraction, which causes the representation error. If we have a lot of real numbers as our data, then already at the beginning of computations we have errors which will be propagated further. The second kind of errors, i.e. rounding errors, occur during each floating-point operation. Although each such operation on modern computers is of maximum accuracy, the result after only a small number of operations can be completely wrong. In classical numerical analysis the error of each individual floating-point operation is estimated, but it is impossible to do it if millions floating-point operations are performed in a computer. It is regrettable, the author believes, that many people do not allow for the tedious fact that the computed results could be inaccurate, not because of the method applied, but also because to the arithmetic used.

While solving ordinary differential equations on a computer (in the form of an initial value problem – see Chapter 2 for the exact definition), we apply approximate methods, which in turn introduce the third kind of errors – the errors of methods. In interval algorithms this kind of errors are included in the interval solutions obtained. Thus, applying interval methods for solving the initial value problem in floating-point interval arithmetic we can obtain solutions in the form of intervals which contain all possible numerical errors. But one should know that the floating--point interval arithmetic is not an antidote for all problems which accompany computations on computers. One inconvenience of this arithmetic is the wrapping effect (see Example 1.5 in Section 1.1) and further efforts should be done by scientists to eliminate it.

This monograph consists of five parts. In Chapter 1 some preliminaries of interval arithmetic are presented. Sections 1.1 and 1.2 deal with mathematical fundamentals of interval arithmetic. In Section 1.4 some hints for realizing floating-point interval arithmetic in any computer language are given and an implementation of it in the Delphi Pascal programming language is presented. In Chapter 2 a mathematical background of the initial value problem is described. This is the problem for which some interval methods presented in the next sections have been developed. It concerns one-step interval methods of Runge-Kutta type presented in Chapter 3 and multistep interval methods described in Chapter 4. For any interval algorithm presented in this monograph a proof of a relevant theorem on the inclusion of the exact solution in interval solutions is given and the widths of interval solutions are estimated. The computational complexities of interval methods are also studied (see Sections 3.6 and 4.6 for details). In Sections 3.7 and 4.7 numerical experiments with interval methods considered are presented (for selected problems) and some comparison between the interval methods are given. Other interval methods known are briefly discussed in Chapter 5.

Chapter 1

Preliminaries of Interval Arithmetic

1.1. Real and Complex Interval Arithmetic

Verified numerical computing requires a mathematical tool to describe operations performed on computers. Such a mathematical tool, called interval arithmetic, was developed by R. E. Moore [133, 134], C. Alefeld and J. Herzberger [5], and A. Neumaier [146]. In this chapter we present only the basic definitions of real and complex interval arithmetic and give some examples. The last section of this chapter deals with a practical implementation of floating-point interval arithmetic on computers.

A *real interval*, or simply an *interval*, is a closed and bounded subset of real numbers $\mathbf{R}$:

$$[x] = \left[\underline{x}, \overline{x}\right] = \left\{x \in \mathbf{R} : \underline{x} \le x \le \overline{x}\right\}, \tag{2.1}$$

where $\underline{x}$ and $\overline{x}$ denote the lower and upper bounds of the interval $[x]$, respectively. An interval is called a *point interval* if $\underline{x} = \overline{x}$. The set of real intervals will be denoted by $\mathbf{IR}$. Since intervals are sets, the well-known terms such as equality $(=)$, subset $(\subseteq)$, proper subset $(\subset)$, superset $(\supseteq)$, proper superset $(\supset)$, membership $(\in)$, union $(\cup)$, and intersection $(\cap)$ are defined in the usual sense of set theory.

For intervals we also define the *inner inclusion relation* $\left(\overset{\circ}{\subset}\right)$. We have

$$[x] \overset{\circ}{\subset} [y], \text{ if } \underline{y} < \underline{x} \text{ and } \overline{x} < \overline{y}. \tag{1.2}$$

About an interval $[x]$, which satisfies (1.2), we say that it is *contained in the interior* of $[y]$. Another relation called the *hull* of intervals is defined as follows:

$$[x] \underline{\cup} [y] = \left[\min\left\{\underline{x}, \underline{y}\right\}, \max\left\{\overline{x}, \overline{y}\right\}\right].$$

Let us note that this relation differs from the union of intervals defined in the common set theory sense.

Example 1.1

Let $[x] = [0, 3]$, and $[y] = [0, 5]$. We have $2 \in [x]$, $4 \notin [x]$, $[x] \subseteq [y]$, and $[y] \supset [x]$. Moreover, $[y] \underline{\cup} [6, 7] = [0, 7]$, but $[y] \cup [6, 7] = [0, 5] \cup [6, 7]$. We also have $[x] \cap [y] = [x]$, and $[x] \cup [y] = [y]$. Let us note that the interval $[x]$ is not contained in the interior of $[y]$, since $\underline{x} = \underline{y}$. ■

For an interval $[x]$ it is easy to define the terms *width* (sometimes called *diameter*), *radius* and *midpoint*:

$$w([x]) = \overline{x} - \underline{x},$$
$$r([x]) = \frac{\overline{x} - \underline{x}}{2},$$
$$m([x]) = \frac{\overline{x} + \underline{x}}{2}.$$

The *smallest* and the *greatest absolute value* of an interval $[x]$ are defined as follows:

$$\left\langle [x] \right\rangle = \min\left\{ \left| x \right| :\ x \in [x] \right\},$$
$$\left| [x] \right| = \max\left\{ \left| x \right| :\ x \in [x] \right\} = \max\left\{ \left| \underline{x} \right|, \left| \overline{x} \right| \right\}.$$

It is easy to check that if $0 \in [x]$, then $\left\langle [x] \right\rangle = 0$. Let us note that both the smallest and the greatest absolute value of an interval are real numbers, but the absolute value of an interval $[x]$, denoted by $\mathrm{abs}([x])$, is an interval and is one of elementary interval functions (see Section 1.2 for details). The width, radius, midpoint, the smallest and the greatest absolute value of an interval are sometimes called *attributes of an interval* (see Figure 1.1).

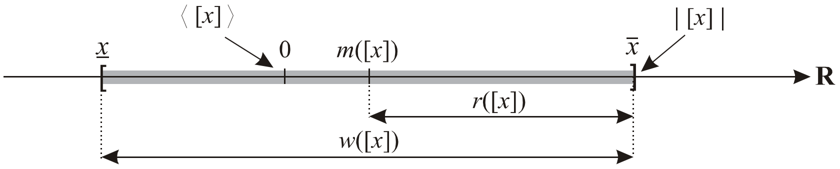


Figure 1.1. Attributes of an interval [x]

For intervals $[x]$ and $[y]$ we can introduce the *distance* $d([x],[y])$ in the following way:

$$d([x],[y]) = \max\left\{ \left| \underline{x} - \underline{y} \right|, \left| \overline{x} - \overline{y} \right| \right\}. \tag{1.3}$$

The distance fulfills all conditions for a metric (it is nonnegative, vanishes if and only if $[x] = [y]$, and holds the triangle inequality), which means that the set **IR** with the metric d is a *metric space*. It can be proved that this space is a complete metric space, i.e. each Cauchy sequence is convergent to an element of this space.

The elementary real operations (addition, subtraction, multiplication and division), i.e. any operation $\circ \in \{+, -, \cdot, /\}$ can be extended to interval arguments $[x]$, $[y]$ by defining the result of an elementary interval operation to be the set of real numbers which results from combining any two numbers included in intervals $[x]$ and $[y]$:

$$[x] \circ [y] = \left\{ x \circ y : \; x \in [x], y \in [y] \right\}. \tag{1.4}$$

From (1.4) it follows that

$$\begin{aligned}
[x] + [y] &= \left[\underline{x} + \underline{y}, \overline{x} + \overline{y} \right], \\
[x] - [y] &= \left[\underline{x} - \overline{y}, \overline{x} - \underline{y} \right], \\
[x] \cdot [y] &= \left[\min\left\{ \underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y} \right\}, \max\left\{ \underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y} \right\} \right], \\
[x] / [y] &= [x] \cdot \left[\frac{1}{\overline{y}}, \frac{1}{\underline{y}} \right], \quad 0 \notin [y].
\end{aligned} \tag{1.5}$$

Example 1.2

Using (1.5) we have

$$\begin{aligned}
&[-1, 0] + [0, \pi] = [-1, \pi], && -1 \cdot [2, 5] = [-5, 2], \\
&[1, 4] - [1, 4] = [-3, 3], && [-2, 3] \cdot [-2, 3] = [-6, 9], \\
&\left[\frac{1}{2}, 1 \right] - \left[0, \frac{1}{6} \right] = \left[\frac{1}{3}, 1 \right], && \left[1, \sqrt{2} \right] \cdot [-1, 1] = \left[-\sqrt{2}, \sqrt{2} \right], \\
&[2, 4] - 3 = [-1, 1], && [1, 2] / [-2, -1] = \left[-2, -\frac{1}{2} \right].
\end{aligned}$$

Let us note that an operation such as $[1, 2] / [-2, 1]$ is indeterminate, since the number 0 is included into the divisor. ■

All the elementary operations on intervals are *inclusion isotonic*, i.e.

$$[x] \subseteq [x'],\ [y] \subseteq [y'] \ \Rightarrow\ [x] \circ [y] \subseteq [x'] \circ [y'], \quad \circ \in \{+, -, \cdot, /\}.$$

Applying interval arithmetic we should know that the distribute law is not generally satisfied and in this arithmetic we have so-called the *subdistributive law*:

$$[x] \cdot ([x] + [z]) \subseteq [x] \cdot [y] + [x] \cdot [z].$$

Example 1.3

Let $[x] = [1, 2],\ [y] = [2, 3],\ [z] = [-4, -3].$ We have

$$[x] \cdot ([y] + [z]) = [-4, 0] \subset [-6, 3] = [x] \cdot [y] + [x] \cdot [z]. \qquad ■$$

The real interval arithmetic can be extended to complex intervals. If intervals $[x_{re}],\ [x_{im}] \in \mathbf{IR},$ then the set

$$[x] = [x_{re}] + i[x_{im}] = \left\{ x = x_{re} + ix_{im} \ :\ x_{re} \in [x_{re}], x_{im} \in [x_{im}] \right\},$$

where $i = \sqrt{-1},$ is called a *complex interval* and the set of complex intervals is denoted by **IC**. If $[x],\ [y] \in \mathbf{IC},$ then we get

$$\begin{aligned}
[x] + [y] &= [x_{re}] + [y_{re}] + i([x_{im}] + [y_{im}]),\\
[x] - [y] &= [x_{re}] - [y_{re}] + i([x_{im}] - [y_{im}]),\\
[x] \cdot [y] &= [x_{re}] \cdot [y_{re}] - [x_{im}] \cdot [y_{im}] + i([x_{re}] \cdot [y_{im}] - [x_{im}] \cdot [y_{rel}]),\\
[x] / [y] &= \frac{[x_{re}] \cdot [y_{re}] + [x_{im}] \cdot [y_{im}]}{[y_{re}]^2 + [y_{im}]^2} + i\frac{[x_{im}] \cdot [y_{re}] - [x_{ire}] \cdot [y_{im}]}{[y_{re}]^2 + [y_{im}]^2},\\
& \quad 0 \notin [y_{re}]^2 + [y_{im}]^2.
\end{aligned} \tag{1.6}$$

It should be pointed out that in the division $[x] / [y]$ the terms $[y_{re}]^2$ and $[y_{im}]^2$ are evaluated using the elementary interval square function (see Section 1.2) instead of evaluating them by self multiplications. This guarantees that for each $[y]$ with $0 \notin [y]$ we have $0 \notin [y_{re}]^2 + [y_{im}]^2.$

Example 1.4 [61, p. 40]

Let $[y] = [-2, 1] + i[1, 2].$ Then $0 = (0, 0) \notin [y],$ and thus

$$0 \notin [y_{re}]^2 + [y_{im}]^2 = [1, 8].$$

Using multiplication instead of the elementary square function (defined in Section 1.2) yields $0 \in [y_{re}] \cdot [y_{re}] + [y_{im}] \cdot [y_{im}] = [-1, 8].$ Thus, the division would fail. ■

There is an important difference between the definitions of elementary operations for real intervals and those for complex intervals. The continuous image of a complex interval is not necessarily another complex interval. Moreover, sometimes we can obtain the effect of overestimation, which is called the *wrapping effect* [35, 58, 112, 114, 115, 139, 148, 173].

Example 1.5 [61, p. 40]

Let $[x] \in \mathbf{IC}$ be a complex interval, and let the interval $[y] = \left[\underline{y}, \overline{y}\right] \in \mathbf{IC}$ be the point interval and such that $\underline{y} = \overline{y} = \cos\alpha + i\sin\alpha$. Multiplication of any $x \in [x]$ with $[y]$ results in a rotation of x by the angle α. Thus, unless α is a multiple of $\pi/2$, the set $\{x \cdot y :\ x \in [x]\}$ is a rectangle with sides not parallel to the coordinate axes. The complex interval multiplication $[x]\cdot[y]$ wraps the given set in a rectangle with sides parallel to the axes as shown in Figure 1.2. ■

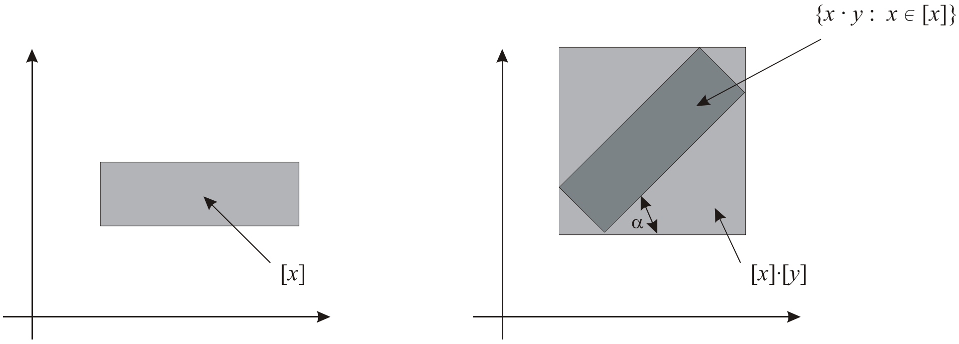


Figure 1.2. Wrapping effect caused by an multiplication of some intervals

Effects similar to the one presented in the above example we can get by applying interval real arithmetic. Understanding the wrapping effect, it will be not take into account in the next chapters.

In real interval arithmetic a division by an interval containing zero cannot be performed. This restriction may be removed in so-called *extended interval arithmetic*, which is defined in the set of extended real intervals

$$\mathbf{IR}^* = \mathbf{IR} \cup \{[-\infty, r]:\ r \in \mathbf{R}\} \cup \{[l, +\infty]:\ l \in \mathbf{R}\} \cup \{[-\infty, +\infty]\}.$$

The interval real arithmetic introduced so far has one inconvenience regarding the division of intervals. Sometimes the definition of the division of intervals is extended to the following:

$$[x]/[y]=\begin{cases}[-\infty,+\infty], & \text{if } \underline{x}<0<\overline{x} \text{ or } [x]=0 \text{ or } [y]=0,\\ \left[\overline{x}/\underline{y},+\infty\right], & \text{if } \overline{x}\geq 0 \text{ and } \underline{y}<0<\overline{y},\\ \left[-\infty,\overline{x}/\overline{y}\right]\cup\left[\overline{x}/\underline{y},+\infty\right], & \text{if } \overline{x}\leq 0 \text{ and } \underline{y}<0<\overline{y},\\ \left[-\infty,\overline{xy}\right], & \text{if } \overline{x}\leq 0 \text{ and } 0=\underline{y}<\overline{y},\\ \left[-\infty,\underline{x}/\underline{y}\right]\cup\left[\underline{x}/\overline{y},+\infty\right], & \text{if } 0\leq\underline{x} \text{ and } \underline{y}<0<\overline{y},\\ \left[\underline{x}/\overline{y},+\infty\right], & \text{if } 0\leq\underline{x} \text{ and } 0=\underline{y}<\overline{y}.\end{cases}$$

In addition to the operation of an extended division, an extended subtraction is often introduced. If $[x]\in\mathbf{IR}$ is a point interval and $[y]\in\mathbf{IR}^*$ has at least one infinite endpoint, we define

$$[x]-[y]=\begin{cases}\left[\overline{x}-\overline{y},+\infty\right], & \text{if } [y]=\left[-\infty,\overline{y}\right],\\ [-\infty,+\infty], & \text{if } [y]=[-\infty,+\infty],\\ \left[-\infty,\overline{x}-\underline{y}\right], & \text{if } [y]=\left[\underline{y},+\infty\right].\end{cases}$$

1.2. Interval Functions

Let $\varphi:\ D\subset\mathbf{R}\to\mathbf{R}$ denote a real-valued elementary function, for instance absolute value, square, square root, exponential function, power function, logarithm, sine, cosine, tangent, cotangent, arc sine, cosine, tangent or cotangent, hyperbolic sine, cosine, tangent or cotangent, inverse hyperbolic sine, cosine, tangent or cotangent, which is continuous on every closed interval contained in D. It is easy to extend φ to interval arguments $[x]\in D$ by the following definition:

$$\varphi([x])=\{\varphi([x]):\ x\in[x]\}=\bigcup_{x\in[x]}\varphi(x). \qquad (1.7)$$

Since we have assumed that φ is continuous, then $\varphi([x])$ is an interval. From (1.7) it follows that an elementary interval function is inclusion isotonic, i.e.

$$[x]\subseteq[y]\ \Rightarrow\ \varphi([x])\subseteq\varphi([y]).$$

If a real-valued elementary function φ is monotonic, then in the case in which $[x]$ is restricted to the domain D of φ, we can easy define an adequate interval elementary function. We have

$$\operatorname{abs}([x])=\left[\left\langle[x]\right\rangle,\left|[x]\right|\right], \qquad (1.8)$$

$$
\begin{aligned}
\varphi([x]) &= \left[\varphi(\underline{x}),\, \varphi(\overline{x})\right], \quad \varphi \in \{\text{arctan, arsinh, ln, sinh}\},\\
\varphi([x]) &= \left[\varphi(\overline{x}),\, \varphi(\underline{x})\right], \quad \varphi \in \{\text{arccot, arcoth}\},\\
[x]^2 &= \operatorname{sqr}([x]) = \left[\langle [x]\rangle^2,\, \left|[x]\right|^2\right],\\
\sqrt{[x]} &= \operatorname{sqrt}([x]) = \left[\sqrt{\underline{x}},\, \sqrt{\overline{x}}\right],\\
e^{[x]} &= \exp([x]) = \left[e^{\underline{x}},\, e^{\overline{x}}\right], \qquad\qquad \text{(1.8 cont.)}\\
[x]^n &= \begin{cases} \left[\underline{x}^n, \overline{x}^n\right], & \text{if } 0 < \underline{x} \text{ or } n \text{ odd},\\ \left[0, \left|[x]\right|^n\right], & \text{if } 0 \in [x] \text{ and } n \text{ even},\\ \left[\overline{x}^n, \underline{x}^n\right], & \text{if } \overline{x} < 0 \text{ and } n \text{ even}. \end{cases}
\end{aligned}
$$

Regarding to (1.5) and (1.8), it should be noted that in general we get only $[x]^2 \subset [x]\cdot[x]$ if $0 \in [x]$. The example with $[x] = [-1, 1]$, which yields

$$[-1, 1]^2 = [0, 1] \neq [-1, 1]\cdot[-1, 1] = [-1, 1],$$

confirms this fact.

For any real-valued function $f : D \subset \mathbf{R} \to \mathbf{R}$ we can extend it to interval arguments $[x] \in D$ in the same way as for a real-valued elementary function, i.e.

$$f([x]) = \bigcup_{x \in [x]} f(x). \tag{1.9}$$

An enclosure of $f([x])$ can be easly obtained if we substitute $[x]$ for x in the defining expression of f, and then evaluate f using interval arithmetic. This kind of evaluation is called an *interval extension* of f and is denoted by $f_{[\,]}([x])$. It should be noted that in general we have

$$f([x]) \subseteq f_{[\,]}([x]).$$

Moreover, one should bear in mind that a real-valued function may have several interval extensions, since it may be defined by several equivalent arithmetic expressions and such expressions do not necessarily yield equivalent interval extensions.

Example 1.6

Let us consider a real-valued function f defined as follows:

$$f(x) = \frac{2}{1-x^2} \equiv \frac{2}{1-x\cdot x} \equiv \frac{1}{1+x} + \frac{1}{1-x} \equiv \frac{2}{(1+x)\cdot(1-x)}, \quad |x| < 1.$$

All notations of f are mathematically equivalent, but their interval extensions are different. Let us specify:

$$f^{(1)}(x) = \frac{2}{1 - x^2}, \quad f^{(2)}(x) = \frac{2}{1 - x \cdot x},$$
$$f^{(3)}(x) = \frac{1}{1 + x} + \frac{1}{1 - x}, \quad f^{(4)}(x) = \frac{2}{(1 + x) \cdot (1 - x)}.$$

The function f is symmetric, decreasing for $x < 0$, increasing for $x > 0$, and for any interval $[x] \in (-1, 1)$ we have $f([x]) = \left[f\left(\left\langle [x] \right\rangle\right), f\left(\left|[x]\right|\right)\right]$, and hence if $[x] = \left[-\frac{1}{4}, \frac{1}{2}\right]$, we get $f([x]) = \left[2, \frac{8}{3}\right]$. But for the interval extensions we have (see Figure 1.3)

$$f_{[\,]}^{(1)}([x]) = \left[2, \tfrac{8}{3}\right] \subset f_{[\,]}^{(2)}([x]) = \left[\tfrac{16}{9}, \tfrac{8}{3}\right] \subset$$
$$\subset f_{[\,]}^{(3)}([x]) = \left[\tfrac{22}{15}, \tfrac{10}{3}\right] \subset f_{[\,]}^{(4)}([x]) = \left[\tfrac{16}{15}, \tfrac{16}{3}\right]. \quad \blacksquare$$

For computer implementations the definition (1.9) is not good, since in general it is impossible to find the union over all real numbers $x \in [x]$ (computers support only finite sets of numbers), and we usually use interval extensions to represent interval functions. In general, it is difficult to determine the best possible interval extension if we have a few mathematical equivalent notations of a real-valued function. However, it is an empirical fact that the fewer occurrences of $[x]$ within an interval extension, the better the result of the corresponding interval evaluation. This fact is confirmed in Example 1.6, where the interval $[x]$ occurs only once in $f_{[\,]}^{(1)}([x])$.

1.3. Floating-Point Interval Arithmetic

In computers real numbers are represented in a form of *floating-point numbers* (sometimes also called *machine numbers*). A floating-point number x is of the form

$$x = \pm m \cdot b^e = \pm 0.m_1 m_2 \ldots m_k \cdot b^e,$$

where m is a signed mantissa of fixed length k, b is the base, and e is the exponent. For the digits of the mantissa we have $1 \le m_1 \le b - 1$, and $0 \le m_i \le b - 1$ for $i = 2, 3, \ldots, k$. This yields $1/b \le m < 1$. The exponent is bounded by $e_{\min} \le e \le e_{\max}$. Floating-point numbers are usually represented with the base $b = 2$.

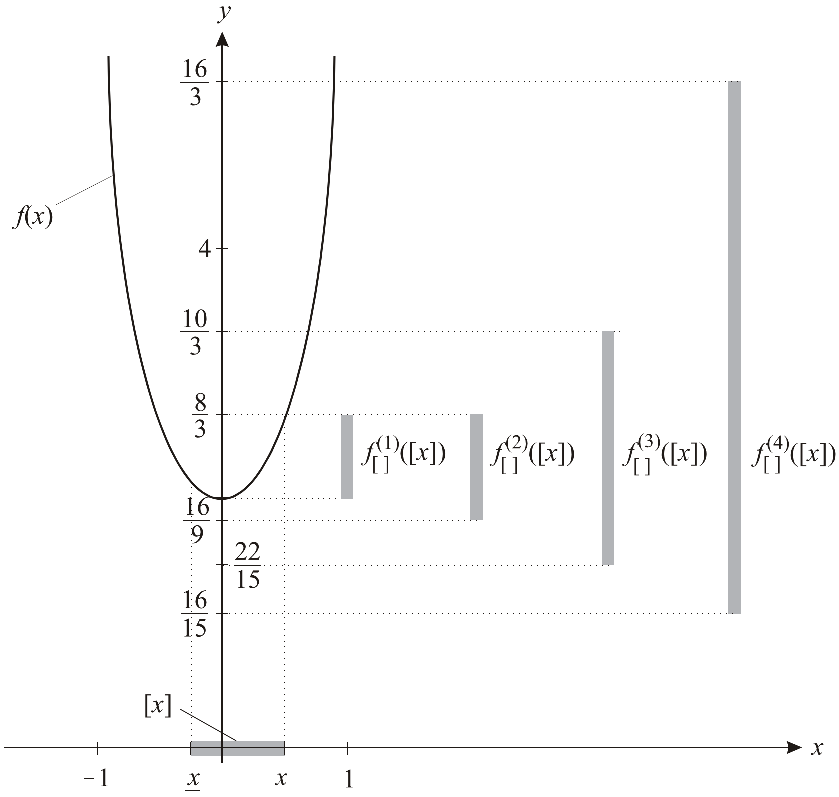


Figure 1.3. Different interval extensions as defined in Example 1.6

A floating-point system will be denoted by R. In order to represent real numbers by floating-point numbers we should map the set **R** into R. This mapping, denoted by $\lozenge$ and called *rounding*, is defined by the two following conditions:

$$\underset{x \in R}{\forall} \lozenge x = x \text{ and } \underset{x, y \in R}{\forall} x \le y \Rightarrow \lozenge x \le \lozenge y.$$

The first condition guarantees that the elements of the set R are not changed by rounding, and the second condition expresses the monotonicity of rounding.

Internal representations of floating-point (machine) real numbers of the ***Single***, the ***Double*** and the ***Extended*** types in the Delphi Pascal programming language are shown in Figure 1.4. The ***Single*** and the ***Double*** types answer a description of single- and double-precision floating-point numbers defined by the IEEE 754 standard [176]. But in Delphi Pascal we recommend using the ***Extended*** type (called *longdouble* in C++) for which the range and precision are the biggest.

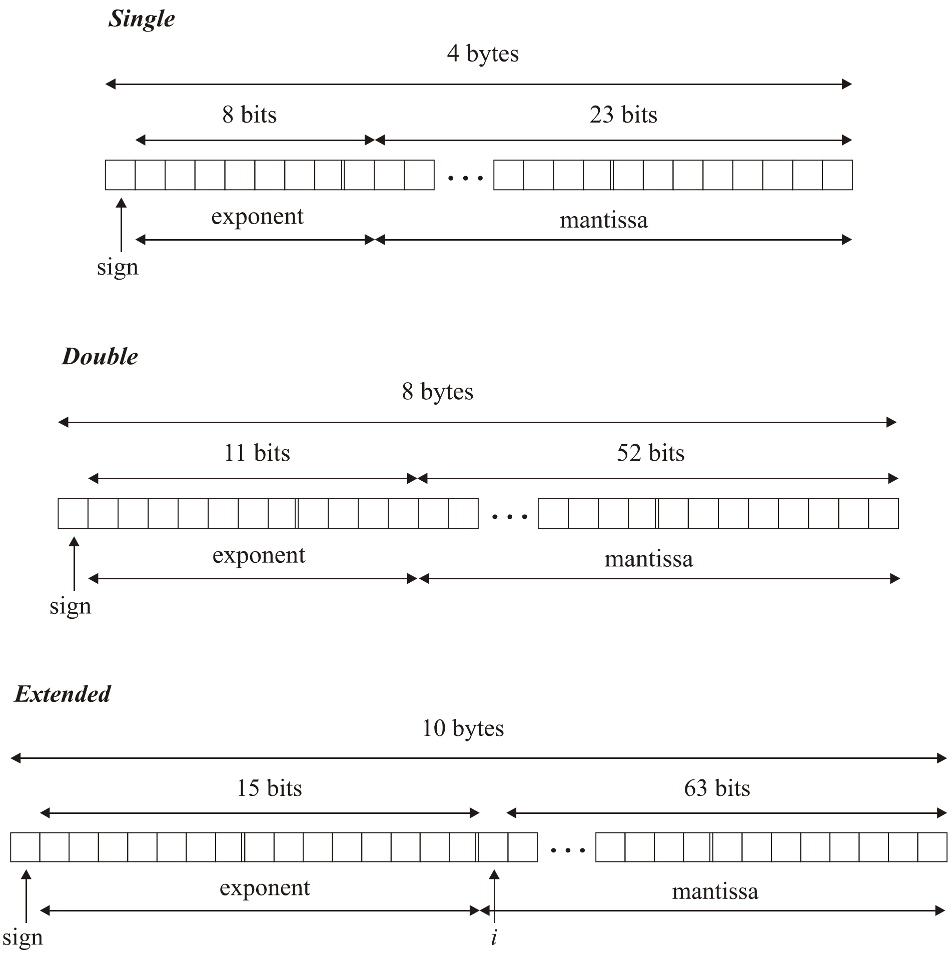


Figure 1.4. Internal representations of the *Single*, *Double* and *Extended* types

The value v of a number of the ***Single*** type is defined as follows:

$$v = \begin{cases} (-1)^s 2^{(e-127)}(1.m), & \text{if } \; 0 < e < 255, \\ (-1)^s 2^{-126}(0.m), & \text{if } \; e = 0 \; \text{ and } \; m \neq 0, \\ (-1)^s 0, & \text{if } \; e = 0 \; \text{ and } \; m = 0, \\ (-1)^s Inf, & \text{if } \; e = 255 \; \text{ and } \; m = 0, \\ NaN, & \text{if } \; e = 255 \; \text{ and } \; m \neq 0, \end{cases}$$

where s denotes the sign (0 or 1), *Inf* is a state of the computer floating-point unit (FPU) called *infinity*, and *NaN* denotes a state of FPU called *not-a-number*. The value v of a number of the ***Double*** type is given by

$$v = \begin{cases} (-1)^s 2^{(e-1023)}(1.m), & \text{if } \ 0 < e < 2047, \\ (-1)^s 2^{-1022}(0.m), & \text{if } \ e = 0 \ \text{ and } \ m \neq 0, \\ (-1)^s 0, \ \ \text{if } \ e = 0 \ \text{ and } \ m = 0, \\ (-1) Inf, \ \ \text{if } \ e = 2027 \ \text{ and } \ m = 0, \\ NaN, \ \ \text{if } \ e = 2047 \ \text{ and } \ m \neq 0. \end{cases}$$

For the value v of a number of the ***Extended*** type we have

$$v = \begin{cases} (-1)^s 2^{(e-16383)}(i.m), & \text{if } \ 0 \le e \le 32767, \\ (-1)^s Inf, \ \ \text{if } \ e = 32767 \ \text{ and } \ m = 0, \\ NaN, \ \ \text{if } \ e = 32767 \ \text{ and } \ m \neq 0, \end{cases} \tag{1.10}$$

where i is the bit before the mantissa point.

Even if one uses floating-point real numbers with the biggest range and precision (as the ***Extended*** type in Delphi Pascal or the *longdouble* type in C++), one should bear in mind that every floating-point number set is finite, not every real number can be represented exactly in this number set, and that standard input/output procedures of programming languages can produce additional rounding errors.

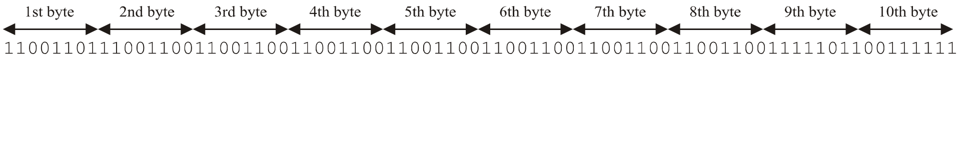
Example 1.7

The binary representation of the real number 0.1 is infinite. Thus, any representation of this number in a floating-point number set cannot be exact. Of course, the representation error depends on floating-point systems used.

Let us consider the ***Extended*** type in Delphi Pascal and let us assume that the number 0.1 has been read (in a console application) by the ***Readln*** (or ***Read***) standard procedure which assigns this number to a variable, say x, of the ***Extended*** type. If we use the ***Writeln*** (or ***Write***) standard procedure with parameter x:26, where 26 is the field width for which the maximum number of significant digits in the mantissa can be obtained, on the screen we will see quite a good result:

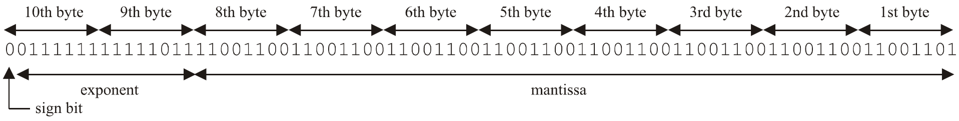
```
 1.0000000000000000E-0001                (1.11)
```

(with one space before the first digit). In computer memory the x variable occupies 10 bytes in which we have



```
11001101110011001100110011001100110011001100110011001100110011001111101100111111
```

In order to interpret these bytes we should reverse them:



From (1.10) it follows that the exact decimal value of the exponent is equal to −4. If we multiply the mantissa by 2^{-4}, in binary system we get

0.0001100110011001100110011001100110011001100110011001100110011001101

This binary number is equal to

0.100000000000000000001355252715606880542509316001087427139282226 5625

in decimal system. But this decimal number is not equal to 0.1 and we see an representation error. It can be checked that a lot of machine numbers, the ones that follow the machine representation of 0.1 (we present them in decimal notation):

0.100000000000000000000813151629364128325505589600652456283569335937 50
0.1000000000000000000014907779871675685967602476011961698532104492187 5

....

0.1000000000000000000049602249391211827855840965639799833297729492187 50
0.1000000000000000000050279875749015268127095623640343546867370605468 75

....

0.1000000000000000000054345633895835909754623571643605828285217285156 25

and the ones that precede this representation:

0.0999999999999999999994578989137572477829962735995650291442871093750 0
0.099999999999999999999878027255595380751174161559902131557464599609375

....

0.099999999999999999999539214076693660615546832559630274772644042968750
0.099999999999999999999471451440913316588421366759575903415679931640625

are all displayed by the ***Writeln*** (x :26) procedure in the form (1.11). Moreover, one can empirically prove (using Delphi Pascal) that there are 88 machine numbers of the ***Extended*** type that this procedure displays as (1.11). It is another matter that all underlined numbers (there are eight such machine numbers: seven that follow the machine representation of 0.1 and one that proceeds this representation) are wrong as a result of rounding by the ***Writeln*** procedure. ■

Modern computers usually implements four kinds of rounding: to the closest value (to the even number if the distance is the same), toward negative infinity, toward positive infinity, and toward zero rounding positive numbers down and negative numbers up. The roundings toward both infinities are very important from the point of view of floating-point interval arithmetic, since we use them in the definitions of elementary floating-point interval operations.

A *floating-point interval* (*machine interval*) is a real interval whose endpoints are floating-point numbers. The set of floating point intervals over R is defined as follows:

$$IR = \left\{ [x] \in \mathbf{IR}: \underline{x}, \overline{x} \in R \right\}.$$

The *elementary floating-point interval operations* are defined in such a way that for all $[x], [y] \in IR$ the resulting interval is the smallest machine interval which contains $[x] \circ [y]$, $\circ \in \{+, -, \cdot, /\}$, i.e.

$$\begin{aligned}
[x] + [y] &= \left[\nabla\left(\underline{x} + \underline{y}\right), \Delta\left(\overline{x} + \overline{y}\right) \right], \\
[x] - [y] &= \left[\nabla\left(\underline{x} - \overline{y}\right), \Delta\left(\overline{x} - \underline{y}\right) \right], \\
[x] \cdot [y] &= \left[\min\left\{ \nabla\left(\underline{x}\underline{y}\right), \nabla\left(\underline{x}\overline{y}\right), \nabla\left(\overline{x}\underline{y}\right), \nabla\left(\overline{x}\overline{y}\right) \right\}, \right. \\
&\qquad \left. \max\left\{ \Delta\left(\underline{x}\underline{y}\right), \Delta\left(\underline{x}\overline{y}\right), \Delta\left(\overline{x}\underline{y}\right), \Delta\left(\overline{x}\overline{y}\right) \right\} \right], \\
[x] / [y] &= \left[\min\left\{ \nabla\left(\underline{x} / \underline{y}\right), \nabla\left(\underline{x} / \overline{y}\right), \nabla\left(\overline{x} / \underline{y}\right), \nabla\left(\overline{x} / \overline{y}\right) \right\}, \right. \\
&\qquad \left. \max\left\{ \Delta\left(\underline{x} / \underline{y}\right), \Delta\left(\underline{x} / \overline{y}\right), \Delta\left(\overline{x} / \underline{y}\right), \Delta\left(\overline{x} / \overline{y}\right) \right\} \right], \quad 0 \notin [y],
\end{aligned} \tag{1.12}$$

where ∇ denotes the rounding toward negative infinity (down), and Δ toward positive infinity (up).

A *complex floating-point interval* is an interval whose real and imaginary parts are floating-point intervals. On the basis of (1.6) and (1.12) the floating-point arithmetic for complex intervals can be introduced very easily. We omit the relevant definitions, because in this book only real floating-point interval operations will be applied.

1.4. An Implementation of Floating-Point Interval Arithmetic in Delphi Pascal

Floating-point interval arithmetic can be implemented in almost any modern programming language. There are well-known three programming languages, developed at the Universität Karlsruhe (Germany), in which floating-point interval arithmetic is fully implemented: PASCAL-XSC [93], C-XSC [94, 102] and FORTRAN-XSC [178], where the abbreviation XSC stands for an eXtension for Scientific Computation. In this section we present an implementation of floating-point interval arithmetic in Delphi Pascal (previously Object Pascal), one of the most popular programming language developed by Borland Software Corporation (CodeGear at present).

The implementation of floating-point interval arithmetic has been written in the form of a unit called *IntervalArithmetic*. This unit takes advantage of the Delphi Pascal floating-point ***Extended*** type and makes it possible to:

- represent any input numerical data in the form of a machine interval (the ends of this interval are equal or are two subsequent machine numbers),
- perform all calculations in floating-point interval arithmetic,
- use some standard interval functions,
- give results in the form of proper intervals (if the ends of an interval are not the same machine numbers, one can see the difference in the output).

The current version of our *IntervalArithmetic* unit is as follows:

```
unit IntervalArithmetic;
// Version 2.13
// (C) Copyright 1998-2009 by Andrzej Marciniak
// Poznan University of Technology, Institute of Computing Science
interface
type interval = record
                  a, b : Extended
                end;
// Basic arithmetic operations
function iadd (const x, y : interval) : interval;
function isub (const x, y : interval) : interval;
function imul (const x, y : interval) : interval;
function idiv (const x, y : interval) : interval;

// Data reading functions
function int_read (const sa : AnsiString) : interval;
function left_read (const sa : AnsiString) : Extended;
function right_read (const sa : AnsiString) : Extended;
function int_width (const x : interval) : Extended;

// A procedure for transforming ends of intervals into strings
procedure iends_to_strings  (const x          : interval;
                             out left, right  : string);

// Basic functions
function isin  (const x  : interval;
                out st   : Integer) : interval;
function icos (const x   : interval;
                out st   : Integer) : interval;
```

```
function iexp  (const x  : interval;
                out st    : Integer) : interval;
function isqr (const x : interval;
               out st   : Integer) : interval;

// Interval constants
function isqrt2 : interval;
function isqrt3 : interval;
function isqrt5 : interval;
function isqrt6 : interval;
function isqrt7 : interval;
function isqrt8 : interval;
function isqrt10 : interval;
function ipi : interval;

implementation
  uses SysUtils, Math, Dialogs;
  type char_tab = array [1..80] of Char;
  const  bit  : array [0..7] of Byte = ($01, $02, $04, $08, $10, $20, $40, $80);
         ldi  : array [0..63] of string [65] =
    (’1.000000000000000000000000000000000000000000000000000000000000000’,
     ’0.500000000000000000000000000000000000000000000000000000000000000’,
     ’0.250000000000000000000000000000000000000000000000000000000000000’,
     ’0.125000000000000000000000000000000000000000000000000000000000000’,
     ’0.062500000000000000000000000000000000000000000000000000000000000’,
     ’0.031250000000000000000000000000000000000000000000000000000000000’,
     ’0.015625000000000000000000000000000000000000000000000000000000000’,
     ’0.007812500000000000000000000000000000000000000000000000000000000’,
     ’0.003906250000000000000000000000000000000000000000000000000000000’,
     ’0.001953125000000000000000000000000000000000000000000000000000000’,
     ’0.000976562500000000000000000000000000000000000000000000000000000’,
     ’0.000488281250000000000000000000000000000000000000000000000000000’,
     ’0.000244140625000000000000000000000000000000000000000000000000000’,
     ’0.000122070312500000000000000000000000000000000000000000000000000’,
     ’0.000061035156250000000000000000000000000000000000000000000000000’,
     ’0.000030517578125000000000000000000000000000000000000000000000000’,
     ’0.000015258789062500000000000000000000000000000000000000000000000’,
     ’0.000007629394531250000000000000000000000000000000000000000000000’,
     ’0.000003814697265625000000000000000000000000000000000000000000000’,
     ’0.000001907348632812500000000000000000000000000000000000000000000’,
     ’0.000000953674316406250000000000000000000000000000000000000000000’,
     ’0.000000476837158203125000000000000000000000000000000000000000000’,
     ’0.000000238418579101562500000000000000000000000000000000000000000’,
     ’0.000000119209289550781250000000000000000000000000000000000000000’,
     ’0.000000059604644775390625000000000000000000000000000000000000000’,
     ’0.000000029802322387695312500000000000000000000000000000000000000’,
     ’0.000000014901161193847656250000000000000000000000000000000000000’,
     ’0.000000007450580596923828125000000000000000000000000000000000000’,
     ’0.000000003725290298461914062500000000000000000000000000000000000’,
     ’0.000000001862645149230957031250000000000000000000000000000000000’,
     ’0.000000000931322574615478515625000000000000000000000000000000000’,
     ’0.000000000465661287307739257812500000000000000000000000000000000’,
     ’0.000000000232830643653869628906250000000000000000000000000000000’,
     ’0.000000000116415321826934814453125000000000000000000000000000000’,
     ’0.000000000058207660913467407226562500000000000000000000000000000’,
     ’0.000000000029103830456733703613281250000000000000000000000000000’,
     ’0.000000000014551915228366851806640625000000000000000000000000000’,
     ’0.000000000007275957614183425903320312500000000000000000000000000’,
     ’0.000000000003637978807091712951660156250000000000000000000000000’,
     ’0.000000000001818989403545856475830078125000000000000000000000000’,
     ’0.000000000000909494701772928237915039062500000000000000000000000’,
     ’0.000000000000454747350886464118957519531250000000000000000000000’,
     ’0.000000000000227373675443232059478759765625000000000000000000000’,
     ’0.000000000000113686837721616029739379882812500000000000000000000’,
     ’0.000000000000056843418860808014869689941406250000000000000000000’,
     ’0.000000000000028421709430404007434844970703125000000000000000000’,
     ’0.000000000000014210854715202003717422485351562500000000000000000’,
     ’0.000000000000007105427357601001858711242675781250000000000000000’,
     ’0.000000000000003552713678800500929355621337890625000000000000000’,
     ’0.000000000000001776356839400250464677810668945312500000000000000’,
     ’0.000000000000000888178419700125232338905334472656250000000000000’,
     ’0.000000000000000444089209850062616169452667236328125000000000000’,
     ’0.000000000000000222044604925031308084726333618164062500000000000’,
     ’0.000000000000000111022302462515654042363166809082031250000000000’,
     ’0.000000000000000055511151231257827021181583404541015625000000000’,
     ’0.000000000000000027755575615628913510590791702270507812500000000’,
     ’0.000000000000000013877787807814456755295395851135253906250000000’,
     ’0.000000000000000006938893903907228377647697925567626953125000000’,
     ’0.000000000000000003469446951953614188823848962783813476562500000’,
     ’0.000000000000000001734723475976807094411924481391906738281250000’,
     ’0.000000000000000000867361737988403547205962240695953369140625000’,
     ’0.000000000000000000433680868994201773602981120347976684570312500’,
     ’0.000000000000000000216840434497100886801490560173988342285156250’,
     ’0.000000000000000000108420217248550443400745280086994171142578125’);

  function iadd (const x, y : interval) : interval;
  begin
    SetRoundMode (rmDown);
    Result.a:=x.a+y.a;
    SetRoundMode (rmUp);
    Result.b:=x.b+y.b
  end {iadd};
```

```
function isub (const x, y : interval) : interval;
begin
  SetRoundMode (rmDown);
  Result.a:=x.a-y.b;
  SetRoundMode (rmUp);
  Result.b:=x.b-y.a
end {isub};

function imul (const x, y : interval) : interval;
var x1y1, x1y2, x2y1 : Extended;
begin
  SetRoundMode (rmDown);
  x1y1:=x.a*y.a;
  x1y2:=x.a*y.b;
  x2y1:=x.b*y.a;
  with Result do
    begin
      a:=x.b*y.b;
      if x2y1<a
        then a:=x2y1;
      if x1y2<a
        then a:=x1y2;
      if x1y1<a
        then a:=x1y1
    end;
  SetRoundMode (rmUp);
  x1y1:=x.a*y.a;
  x1y2:=x.a*y.b;
  x2y1:=x.b*y.a;
  with Result do
    begin
      b:=x.b*y.b;
      if x2y1>b
        then b:=x2y1;
      if x1y2>b
        then b:=x1y2;
      if x1y1>b
        then b:=x1y1
    end
end {imul};

function idiv (const x, y : interval) : interval;
var x1y1, x1y2, x2y1 : Extended;
begin
  if (y.a<=0) and (y.b>=0)
    then raise EZeroDivide.Create ('Division by an interval containing 0.')
    else  begin
            SetRoundMode (rmDown);
            x1y1:=x.a/y.a;
            x1y2:=x.a/y.b;
            x2y1:=x.b/y.a;
            with Result do
              begin
                a:=x.b/y.b;
                if x2y1<a
                  then a:=x2y1;
                if x1y2<a
                  then a:=x1y2;
                if x1y1<a
                  then a:=x1y1
              end;
            SetRoundMode (rmUp);
            x1y1:=x.a/y.a;
            x1y2:=x.a/y.b;
            x2y1:=x.b/y.a;
            with Result do
              begin
                b:=x.b/y.b;
                if x2y1>b
                  then b:=x2y1;
                if x1y2>b
                  then b:=x1y2;
                if x1y1>b
                  then b:=x1y1
              end
          end
end {idiv};

procedure to_fixed_point  (const awzi        : char_tab;
                            var significand  : AnsiString);
var  exponent               : Smallint;
     i, j, k, code          : Integer;
     remember, s1, s2, sum  : Byte;
     sumz                   : string [2];
begin
  exponent:=0;
  j:=1;
  for i:=16 downto 2 do
    begin
      if awzi[i]='1'
        then exponent:=exponent+j;
      j:=2*j
    end;
```

```
exponent:=exponent−16383;
for i:=80 downto 17 do
  if awzi[i]=’1’
    then  begin
            remember:=0;
            for j:=65 downto 3 do
              begin
                Val (significand[j], s1, code);
                Val (ldi[i−17,j], s2, code);
                sum:=s1+s2+remember;
                Str (sum, sumz);
                if sum>9
                  then  begin
                          significand[j]:=sumz[2];
                          Val (sumz[1], remember, code);
                          if j=3
                            then  begin
                                    Val (significand[1], s1, code);
                                    sum:=s1+remember;
                                    Str (sum, sumz);
                                    significand[1]:=sumz[1]
                                  end
                        end
                  else  begin
                          significand[j]:=sumz[1];
                          remember:=0
                        end
              end;
            Val (significand[1], s1, code);
            Val (ldi[i−17,1], s2, code);
            sum:=s1+s2;
            Str (sum, sumz);
            significand[1]:=sumz[1]
          end;
if exponent>0
  then  for i:=1 to exponent do
          begin
            j:=Length(significand);
            remember:=0;
            for k:=j downto j−62 do
              begin
                Val (significand[k], s1, code);
                sum:=2*s1+remember;
                Str (sum, sumz);
                if sum>9
                  then begin
```

```
                        significand[k]:=sumz[2];
                        Val (sumz[1], remember, code)
                      end
                else  begin
                        significand[k]:=sumz[1];
                        remember:=0
                      end
            end;
          for k:=j−64 downto 1 do
            begin
              Val (significand[k], s1, code);
              sum:=2*s1+remember;
              Str (sum, sumz);
              if sum>9
                then begin
                        significand[k]:=sumz[2];
                        Val (sumz[1], remember, code);
                        if k=1
                          then significand:=sumz[1]+significand
                      end
                else  begin
                        significand[k]:=sumz[1];
                        remember:=0
                      end
            end
        end
else if exponent<0
       then  for i:=1 to Abs(exponent) do
               begin
                 j:=Length(significand);
                 if significand[1]='1'
                   then  begin
                           significand[1]:='0';
                           remember:=10
                         end
                   else   remember:=0;
                 for k:=3 to j do
                   begin
                     Val (significand[k], s1, code);
                     sum:=remember+s1;
                     s1:=sum div 2;
                     Str (s1, sumz);
                     significand[k]:=sumz[1];
                     remember:=10*(sum mod 2);
                     if (k=j) and (remember<>0)
                       then significand:=significand+'5'
                       end
                   end;
  if awzi[1]=’1’
    then significand:=’-’+significand
    else significand:=’+’+significand;
  if DecimalSeparator=’,’
    then while (significand[Length(significand)]=’0’)
                 and (significand[Length(significand)-1]<>’,’) do
            significand:=Copy(significand, 1, Length(significand)-1)
    else  while (significand[Length(significand)]=’0’)
                 and (significand[Length(significand)-1]<>’.’) do
            significand:=Copy(significand, 1, Length(significand)-1)
end {to_fixed_point};

function int_read (const sa : AnsiString) : interval;
var  x, px, nx             : Extended;
     sa1, sx               : AnsiString;
     i, j                  : Integer;
     tab                   : array [1..10] of Byte absolute x;
     eps                   : array [1..10] of Byte;
     epsx                  : Extended absolute eps;
     epsw                  : Word absolute eps;
     digits, rev_digits    : char_tab;
     ix                    : interval;
     sep                   : Char;
begin
  sa1:=sa;
  if DecimalSeparator=’,’
    then sep:=’,’
    else sep:=’.’;
  if (Pos(’.’, sa1)>0) and (DecimalSeparator=’,’)
    then sa1[Pos(’.’, sa1)]:=’,’;
  x:=StrToFloat(sa1);
  if Pos(’e’, sa1)>0
    then sa1[Pos(’e’, sa1)]:=’E’;
  while sa1[1]=’ ’ do
    Delete (sa1, 1, 1);
  while sa1[Length(sa1)]=’ ’ do
    Delete (sa1, Length(sa1), 1);
  if (sa1[1]<>’-’) and (sa1[1]<>’+’)
    then Insert (’+’, sa1, 1);
  while (sa1[2]=’0’) and (Length(sa1)>2) and (sa1[3]<>’e’) and (sa1[3]<>'E')
        and (sa1[3]<>sep) do
    Delete (sa1, 2, 1);
  if (sa1[Length(sa1)]=’E’) or (sa1[Length(sa1)]=’+’) or (sa1[Length(sa1)]=’-’)
    then sa1:=sa1+’0’
  else  if Pos('E', sa1)=0
          then sa1:=sa1+'E0';
if Pos(sep, sa1)=0
  then Insert (sep+'0', sa1, Pos('E', sa1));
sx:=Copy(sa1, Pos('E', sa1)+1, Length(sa1)-Pos('E', sa1));
sa1:=Copy(sa1, 1, Pos('E', sa1)-1);
j:=StrToInt(sx);
if j>0
  then  for i:=1 to j do
          begin
            Insert (sep, sa1, Pos(sep, sa1)+2);
            Delete (sa1, Pos(sep, sa1), 1);
            if Pos(sep, sa1)=Length(sa1)
              then sa1:=sa1+'0'
          end
  else  if j<0
          then  for i:=j to -1 do
                  begin
                    Insert (sep, sa1, Pos(sep, sa1)-1);
                    Delete (sa1, Pos(sep, sa1)+2, 1);
                    if sa1[2]=sep
                      then Insert ('0', sa1, 2)
                  end;
while (sa1[Length(sa1)]='0') and (sa1[Length(sa1)-1]<>sep) do
  sa1:=Copy(sa1, 1, Length(sa1)-1);
for i:=1 to 10 do
  for j:=7 downto 0 do
    if tab[i] and bit[j] = bit[j]
      then digits[8*i-j]:='1'
      else digits[8*i-j]:='0';
for i:=1 to 10 do
  for j:=1 to 8 do
    rev_digits[8*(i-1)+j]:=digits[80-8*i+j];
sx:='0'+sep
   +'0000000000000000000000000000000000000000000000000000000000000000000';
to_fixed_point (rev_digits, sx);
if sa1=sx
  then  begin
          ix.a:=x;
          ix.b:=x
        end
  else  begin
          for i:=18 to 80 do
            rev_digits[i]:='0';
          rev_digits[17]:='1';
          rev_digits[1]:='0';
```

```
for i:=1 to 2 do
  begin
    eps[i]:=0;
    for j:=1 to 8 do
      if rev_digits[8*(i–1)+j]=’1’
        then eps[i]:=eps[i] or bit[8-j]
  end;
epsw:=Swap(epsw);
epsw:=epsw–63;
epsw:=Swap(epsw);
for i:=1 to 2 do
  for j:=7 downto 0 do
if eps[i] and bit[j] = bit[j]
  then rev_digits[8*i–j]:=’1’
  else rev_digits[8*i–j]:=’0’;
for i:=1 to 10 do
  for j:=1 to 8 do
    digits[8*(i–1)+j]:=rev_digits[80–8*i+j];
for i:=1 to 10 do
  begin
    eps[i]:=0;
    for j:=1 to 8 do
      if digits[8*(i–1)+j]=’1’
        then eps[i]:=eps[i] or bit[8–j]
  end;
px:=x–epsx;
nx:=x+epsx;
i:=Length(sa1)–Pos(sep, sa1);
j:=Length(sx)–Pos(sep, sx);
if j>i
  then i:=j;
while Length(sa1)–Pos(sep, sa1)<i do
  sa1:=sa1+’0’;
while Length(sx)–Pos(sep, sx)<i do
  sx:=sx+’0’;
i:=Pos(sep, sa1);
j:=Pos(sep, sx);
if j>i
  then i:=j;
while Pos(sep, sa1)<i do
  Insert (’0’, sa1, 2);
while Pos(sep, sx)<i do
  Insert (’0’, sx, 2);
if sx[1]=’+’
  then if sa1<sx
          then  begin
```

```
                               ix.a:=px;
                               ix.b:=x
                             end
                      else   begin
                               ix.a:=x;
                               ix.b:=nx
                             end
              else  if sa1<sx
                      then   begin
                               ix.a:=x;
                               ix.b:=nx
                             end
                      else   begin
                               ix.a:=px;
                               ix.b:=x
                             end
          end;
  Result.a:=ix.a;
  Result.b:=ix.b
end {int_read};

function left_read (const sa : AnsiString) : Extended;
var int_number : interval;
begin
  int_number:=int_read(sa);
  Result:=int_number.a
end {left_read};

function right_read (const sa : AnsiString) : Extended;
var int_number : interval;
begin
  int_number:=int_read(sa);
  Result:=int_number.b
end {right_read};

function int_width (const x : interval) : Extended;
begin
  if x.a=x.b
    then Result:=0
    else Result:=x.b−x.a
end {int_width};

procedure iends_to_strings (const x           : interval;
                               out left, right : string);
procedure modify_mantissa  (const i          : Integer;
                                var mantissa : string);
var s, s1 : string;
begin
```

```
  if i>=0
    then Insert (’+’, mantissa, 21)
    else Insert (’−’, mantissa, 21);
  Str (Abs(i), s1);
  if i<10
    then s:=’000’+s1
    else  if i<100
           then s:=’00’+s1
           else if i<1000
                  then s:=’0’+s1
                  else s:=s1;
  Insert (s, mantissa, 22)
end;
function take_up (var fl_str : string) : string;
var  s, s1        : string;
     code, i, k : Integer;
     finished   : Boolean;
begin
  finished:=False;
  k:=19;
  repeat
    s:=Copy(fl_str, k, 1);
    Delete (fl_str, k, 1);
    Val (s, i, code);
    i:=i+1;
    if i<10
      then  begin
              Str (i, s);
              Insert (s, fl_str, k);
              finished:=True
            end
      else  begin
              Insert (’0’, fl_str, k);
              k:=k−1
            end
  until finished or (k<4);
  if not finished
    then begin
            s:=Copy(fl_str, 2, 1);
            Delete (fl_str, 2, 1);
            Val (s, i, code);
            i:=i+1;
            if i<10
              then begin
                     Str (i, s);
```

```
                      Insert (s, fl_str, 2)
                    end
              else  begin
                      Insert (’1’, fl_str, 2);
                      s:=’0’;
                      for k:=4 to 19 do
                        begin
                          s1:=Copy(fl_str, k, 1);
                          Delete (fl_str, k, 1);
                          Insert (s, fl_str, k);
                          s:=s1
                        end;
                      s:=Copy(fl_str, 21, 5);
                      Delete (fl_str, 21, 5);
                      Val (s, i, code);
                      i:=i-1;
                      modify_mantissa (i, fl_str)
                    end
          end;
  Result:=fl_str
end;
function take_down (var fl_str : string) : string;
var  s          : string;
     code, i, k : Integer;
     finished   : Boolean;
begin
  finished:=False;
  k:=19;
  repeat
    s:=Copy(fl_str, k, 1);
    Delete (fl_str, k, 1);
    Val (s, i, code);
    i:=i-1;
    if i>-1
      then  begin
              Str (i, s);
              Insert (s, fl_str, k);
              finished:=True
            end
      else  begin
              Insert (’9’, fl_str, k);
              k:=k-1
            end
  until finished or (k<4);
  if not finished
    then  begin
```

```
      s:=Copy(fl_str, 2, 1);
      Delete (fl_str, 2, 1);
      Val (s, i, code);
      i:=i−1;
      if i>0
       then begin
               Str (i, s);
               Insert (s, fl_str, 2)
             end
       else  begin
               s:=Copy(fl_str, 4, 1);
               Insert (s, fl_str, 2);
               for k:=4 to 18 do
                 begin
                   s:=Copy(fl_str, k+1, 1);
                   Delete (fl_str, k+1, 1);
                   Insert (s, fl_str, k)
                 end;
               Delete (fl_str, 19, 1);
               Insert (’9’, fl_str, 19);
               s:=Copy(fl_str, 21, 5);
               Delete (fl_str, 21, 5);
               Val (s, i, code);
               i:=i−1;
               modify_mantissa (i, fl_str)
             end
    end;
 Result:=fl_str
end;
var code        : Integer;
    y, z        : Extended;
    fixed_number: AnsiString;
begin
 if x.a<=x.b
  then if x.a>=0
         then begin
                Str (x.a:26, left);
                Delete (left, 20, 1);
                Val (left, z, code);
                if x.a<z
                  then left:=take_down(left);
                Str (x.b:25, right);
                fixed_number:=right;
                y:=left_read(fixed_number);
                Val (right, z, code);
```

```
                    if (x.b>=z) and (x.a<>x.b) and (y<>x.b)
                      then right:=take_up(right)
                  end
            else  if x.b<=0
                    then begin
                            Str (x.a:25, left);
                            fixed_number:=left;
                            y:=right_read(fixed_number);
                            Val (left, z, code);
                            if (x.a<=z) and (x.a<>x.b) and (y<>x.a)
                              then left:=take_up(left);
                            Str (x.b:26, right);
                            Delete (right, 20, 1);
                            Val (right, z, code);
                            if x.b>z
                              then right:=take_down(right)
                          end
                    else  begin
                            Str (x.a:25, left);
                            fixed_number:=left;
                            y:=right_read(fixed_number);
                            Val (left, z, code);
                            if (x.a<=z) and (y<>x.a)
                              then left:=take_up(left);
                            Str (x.b:25, right);
                            fixed_number:=right;
                            y:=left_read(fixed_number);
                            Val (right, z, code);
                            if (x.b>=z) and (y<>x.b)
                              then right:=take_up(right)
                          end
end {iends_to_strings};

function isin  (const x  : interval;
                out st   : Integer) : interval;
var  is_even, finished : Boolean;
     k                 : Integer;
     d, s, w, w1, x2   : interval;
begin
  if x.a>x.b
    then st:=1
    else  begin
            s:=x;
            w:=x;
            x2:=imul(x,x);
            k:=1;
            is_even:=True;
```

```
finished:=False;
st:=0;
repeat
  d.a:=(k+1)*(k+2);
  d.b:=d.a;
  s:=imul(s,idiv(x2,d));
  if is_even
    then  w1:=isub(w,s)
    else  w1:=iadd(w,s);
  if (w.a<>0) and (w.b<>0)
    then  if (Abs(w.a-w1.a)/Abs(w.a)<1e-18)
               and (Abs(w.b-w1.b)/Abs(w.b)<1e-18)
            then finished:=True
            else
    else  if (w.a=0) and (w.b<>0)
            then  if (Abs(w.a-w1.a)<1e-18)
                       and (Abs(w.b-w1.b)/Abs(w.b)<1e-18)
                     then finished:=True
                     else
            else  if w.a<>0
                     then  if (Abs(w.a-w1.a)/Abs(w.a)<1e-18)
                                and (Abs(w.b-w1.b)<1e-18)
                              then finished:=True
                              else
                     else  if (Abs(w.a-w1.a)<1e-18)
                                and (Abs(w.b-w1.b)<1e-18)
                              then finished:=True;
  if finished
    then  begin
            if w1.b>1
              then begin
                      w1.b:=1;
                      if w1.a>1
                        then w1.a:=1
                    end;
            if w1.a<-1
              then begin
                      w1.a:=-1;
                      if w1.b<-1
                        then w1.b:=-1
                    end;
            Result:=w1
          end
    else  begin
            w:=w1;
            k:=k+2;
```

```
                        is_even:=not is_even
                      end
            until finished or (k>MaxInt/2);
            if not finished
              then st:=2
          end
end {isin};

function icos (const x : interval;
                out st  : Integer) : interval;
var is_even, finished : Boolean;
     k                 : Integer;
     d, c, w, w1, x2   : interval;
begin
  if x.a>x.b
    then st:=1
    else begin
            c.a:=1;
            c.b:=1;
            w:=c;
            x2:=imul(x,x);
            k:=1;
            is_even:=True;
            finished:=False;
            st:=0;
            repeat
              d.a:=k*(k+1);
              d.b:=d.a;
              c:=imul(c,idiv(x2,d));
              if is_even
                then w1:=isub(w,c)
                else w1:=iadd(w,c);
              if (w.a<>0) and (w.b<>0)
                then if (Abs(w.a-w1.a)/Abs(w.a)<1e-18)
                          and (Abs(w.b-w1.b)/Abs(w.b)<1e-18)
                        then finished:=True
                        else
                else if (w.a=0) and (w.b<>0)
                        then if (Abs(w.a-w1.a)<1e-18)
                                  and (Abs(w.b-w1.b)/Abs(w.b)<1e-18)
                                then finished:=True
                                else
                        else if w.a<>0
                                then if (Abs(w.a-w1.a)/Abs(w.a)<1e-18)
                                          and (Abs(w.b-w1.b)<1e-18)
                                        then finished:=True
                                        else
                                else  if (Abs(w.a-w1.a)<1e-18)
                                          and (Abs(w.b-w1.b)<1e-18)
                                        then finished:=True;
              if finished
                then  begin
                        if w1.b>1
                          then begin
                                  w1.b:=1;
                                  if w1.a>1
                                    then w1.a:=1
                                end;
                        if w1.a<-1
                          then begin
                                  w1.a:=-1;
                                  if w1.b<-1
                                    then w1.b:=-1
                                end;
                        Result:=w1
                      end
                else  begin
                        w:=w1;
                        k:=k+2;
                        is_even:=not is_even
                      end
            until finished or (k>MaxInt/2);
            if not finished
              then st:=2
          end
end {icos};

function iexp (const x  : interval;
                out st    : Integer) : interval;
var  finished     : Boolean;
     k            : Integer;
     d, e, w, w1 : interval;
begin
  if x.a>x.b
    then  st:=1
    else  begin
            e.a:=1;
            e.b:=1;
            w:=e;
            k:=1;
            finished:=False;
            st:=0;
            repeat
              d.a:=k;
              d.b:=k;
              e:=imul(e,idiv(x,d));
              w1:=iadd(w,e);
              if (Abs(w.a-w1.a)/Abs(w.a)<1e-18)
                  and (Abs(w.b-w1.b)/Abs(w.b)<1e-18)
                then begin
                       finished:=True;
                       Result:=w1
                     end
                else begin
                       w:=w1;
                       k:=k+1
                     end
            until finished or (k>MaxInt/2);
            if not finished
              then st:=2
          end
end {iexp};

function isqr  (const x  : interval;
                out st   : Integer) : interval;
var minx, maxx : Extended;
begin
  if x.a>x.b
    then st:=1
    else begin
           st:=0;
           if (x.a<=0) and (x.b>=0)
             then minx:=0
             else if x.a>0
                    then minx:=x.a
                    else  minx:=x.b;
           if Abs(x.a)>Abs(x.b)
             then maxx:=Abs(x.a)
             else maxx:=Abs(x.b);
           SetRoundMode (rmDown);
           Result.a:=minx*minx;
           SetRoundMode (rmUp);
           Result.b:=maxx*maxx
         end
end {isqr};

function isqrt2 : interval;
var i2 : AnsiString;
begin
  i2:='1.414213562373095048';
  Result.a:=left_read(i2);
  i2:=’1.414213562373095049’;
  Result.b:=right_read(i2)
end {isqrt2};

function isqrt3 : interval;
var i3 : AnsiString;
begin
  i3:=’1.732050807568877293’;
  Result.a:=left_read(i3);
  i3:=’1.732050807568877294’;
  Result.b:=right_read(i3)
end {isqrt3};

function isqrt5 : interval;
var i5 : AnsiString;
begin
  i5:=’2.236067977499789696’;
  Result.a:=left_read(i5);
  i5:=’2.236067977499789697’;
  Result.b:=right_read(i5)
 end {isqrt5};

function isqrt6 : interval;
var i6 : AnsiString;
begin
  i6:=’2.449489742783178098’;
  Result.a:=left_read(i6);
  i6:=’2.449489742783178099’;
  Result.b:=right_read(i6)
end {isqrt6};

function isqrt7 : interval;
var i7 : AnsiString;
begin
  i7:=’2.645751311064590590’;
  Result.a:=left_read(i7);
  i7:=’2.645751311064590591’;
  Result.b:=right_read(i7)
end {isqrt7};

function isqrt8 : interval;
var i8 : AnsiString;
begin
  i8:=’2.828427124746190097’;
  Result.a:=left_read(i8);
  i8:=’2.828427124746190098’;
  Result.b:=right_read(i8)
end {isqrt8};
```

```
  function isqrt10 : interval;
  var i10 : AnsiString;
  begin
    i10:='3.162277660168379331';
    Result.a:=left_read(i10);
    i10:='3.162277660168379332';
    Result.b:=right_read(i10)
  end {isqrt10};

  function ipi : interval;
  var ipistr : AnsiString;
  begin
    ipistr:='3.141592653589793238';
    Result.a:=left_read(ipistr);
    ipistr:='3.141592653589793239';
    Result.b:=right_read(ipistr)
  end {ipi};

initialization
  ;
finalization
  SetRoundMode (rmNearest)
end.
```

The *IntervalArithmetic* unit needs some comments. It is obvious that the functions *iadd*, *isub*, *imul* and *idiv* perform elementary floating-point interval operations (addition, subtraction, multiplication and division, respectively). Within these functions the standard Delphi Pascal procedure ***SetRoundMode*** is used, which sets the floating-point unit (FPU) rounding mode either toward negative infinity (down, by the predefined constant ***mrDown***) or toward positive infinity (up, by the predefined constant ***mrUp***). In both these cases the ***SetRoundMode*** procedure changes the 10-th and 11-th bits of the FPU control register (word) substituting 01 for rounding down and 10 for rounding up (see Figure 1.5).

In all graphical user interface (GUI) applications numerical data are entered in the form of strings. Thus, it is necessary to convert these strings to numerical values. In the Delphi Pascal programming language there are a number of functions and procedures to perform such conversions. In the *IntervalArithmetic* unit to obtain a machine interval from a string containing a numerical value we have implemented the function *int_read*. This function is rather complicated, although it executes an algorithm which can be described in a quite accessible way.

Let *a* be a number one enters into a GUI application and let *x* be a variable of the ***Extended*** type which stores *a* within the application. Let us denote by *sa* a variable of long string type (***AnsiString***) which stores the sequence of characters entered for the number *a*. Let *sx* be a long string type variable which stores the fixed-point decimal machine representation of *a*, i.e. the value of *x* obtained from its internal (machine) representation. If the number *a* is entered in the floating-point notation, then we should convert *sa* to the fixed-point notation. Note that taking into account the range of the ***Extended*** numbers, the strings *sa* and *sx* can be very long, but they will have less than 5,000 characters, which is not too many in comparison to the maximum number of characters in such strings. In order to perform the conversions into strings containing the fixed-point numbers, within the function *int_read* the array *ldi* of constant strings is used (see the listing of *IntervalArithmetic* unit). This array contains decimal values (in the form of strings) of all possible powers of 2 for the mantissa of the ***Extended*** type numbers (from 2^0 to 2^{-63}). Of course, these powers could be calculated within the function *int_read*, but using the constant strings makes the execution of this function faster.

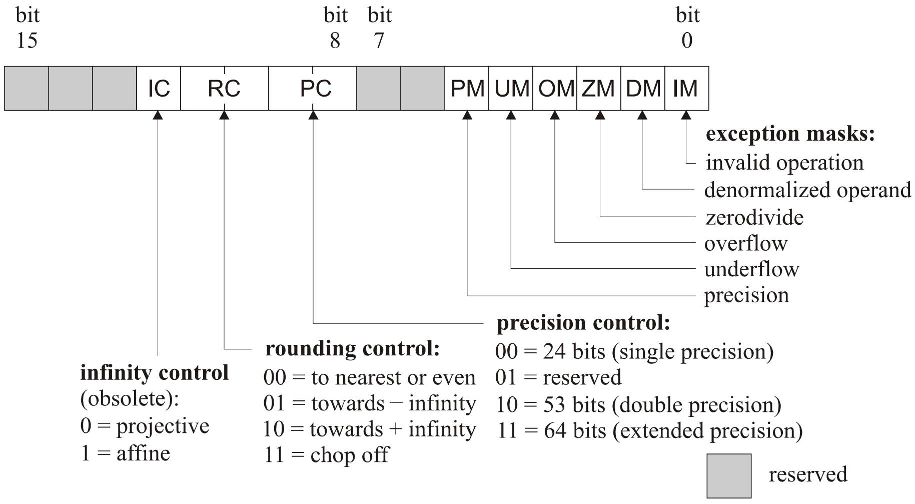


Figure 1.5. FPU control register

If *sa* = *sx*, then the number *a* is exactly represented in the computer memory and the width of machine interval for this number is equal to 0 ([*x*, *x*] is such an interval).

If *sa* ≠ *sx*, then the machine representation of *a*, i.e. *x*, differs from *a*. From an analysis of the internal representation of the ***Extended*** type numbers it follows that if the exponents are equal, then two subsequent machine numbers differ in 2^{-63}. Thus, adding 1 to the last bit of the internal representation of the mantissa of *x*, and performing possibly a modification of the exponent, we obtain the next subsequent machine number with reference to *x*. Similarly, if we subtract 1 from the last bit of the internal representation of the mantissa of x, we get the previous subsequent machine number with reference to x[1].

We assign the obtained values to the following variables:

nx – in the case of the next subsequent machine number,
px – in the case of the previous subsequent machine number.

Moreover, let us introduce:

snx – the long string type variable containing decimal fixed-point value of nx,
spx – the long string type variable containing decimal fixed-point value of px.

Thus, we have four long strings: sa, sx, snx and spx. If a positive number is entered, it is sufficient to check which of the following string relation pairs are fulfilled:

$$spx < sa < sx \text{ or } sx < sa < snx.$$

If the first pair of the inequality is fulfilled, then the interval $[px, x]$ is the floating--point interval representing the given (non-machine) number a. In the second case we have to choose the interval $[x, nx]$.

Let us note that for a positive number a the inequalities

$$spx < sa \text{ and } sa < snx$$

are always fulfilled (if the introduced number is exactly represented in a computer, then $sa = sx$, and moreover we have $spx < sx <snx$). Thus, in practice it is sufficient to determine only the long strings sa and sx, and to check the correctness of one of the following inequalities:

$$sa < sx \text{ or } sx < sa.$$

In the case of a negative number a, if the first inequality is true, then the interval $[x, nx]$ should be taken as the floating-point interval representation of the non-machine number a, and if the second inequality is fulfilled we should take the interval $[px, x]$.

In many problems the input data are not given in a form of real numbers, but in a form of real intervals. For instance, such a situation occurs if the data are obtained from some measurements. In such a case it is necessary to represent a real interval,

[1] Instead of these operations we can add a variable eps to x and subtract eps from x, where the value of eps is determined by the internal (normalized) form of the mantissa equals 1 and by the exponent smaller in 63 from the exponent of the internal representation of x. The determination of the value of eps is possible for each variable x of the ***Extended*** type which absolute value is in 2^{63} (approximately $5{\cdot}10^{20}$) greater from the minimal positive number within this type. Taking into account that this number is approximately equal to $3.6{\cdot}10^{-4951}$, it is enough to assume 10^{-4930} as the lowest range of considered values.

say $[a, b]$ $(a < b)$, by a floating-point interval, say $[x, y]$, where x and y are machine numbers $(x < y)$, and where $x \leq a$ and $b \leq y$. In the *IntervalArithmetic* unit we have the function *left_read*, which for a given real number a, entered as a string, finds the largest machine number x of the ***Extended*** type for which $x \leq a$. We also have the function *right_read*, which for a given real number b, also entered in a form of a string, finds the smallest machine number y of the same ***Extended*** type for which $b \leq y$. Thus, the obtained machine interval $[x, y]$ contains the given real interval $[a, b]$.

The function *int_width*, present in the *IntervalArithmetic* unit, is a simple function for calculating the width of a given interval.

When for the output one uses standard Delphi Pascal functions or procedures, the obtained (on the screen or printer) floating-point values are rounded. If such functions or procedures are used for presenting left and right ends of intervals one can see that both the ends are equal, although in fact they differ on positions that are not displayed. In the *IntervalArithmetic* unit we have the procedure called *iends_to_strings* which prevents such a situation. For a given machine interval this procedure gives the left end and the right end of the interval in the form of strings. If the ends of the interval are not the same machine numbers, one can see the difference in the displayed string.

Example 1.8

Although the procedure *iends_to_strings* is designed primarily for using in GUI applications, one can just as well use it in console applications.

Let us assume that x is a variable of the *interval* type, and let *sa* be an ***AnsiString*** string. After executing the instructions:

```
sa:='0.1';
x:=int_read(sa);
```

the variable x stores the interval containing the real number 0.1 (the ends of this interval are two subsequent machine numbers). If we use the standard Delphi Pascal ***Str*** procedure to get the strings containing ends of this interval, i.e. execute the following instructions:

```
Str (x.a:25, left);
Str (x.b:25, right);
Writeln ('x = [', left, ', ', right, ']');
```

(*left* and *right* are variables of the **string** type), on the display we obtain

```
x = [ 1.00000000000000000E-0001, 1.00000000000000000E-0001]
```

and one can expect that the real number 0.1 are exactly represented in computer memory. Using the procedure *iends_to_string* and the function *int_width* (to display the interval width), i.e. executing the instructions:

```
   iends_to_strings (x, left, right);
   Writeln ('x = [', left, ', ', right, ']');
   Writeln ('width = ', int_width(x):10);
```

on the display we get

```
   x = [ 9.9999999999999999E-0002,  1.0000000000000001E-0001]
   width =   6.8E-0021
```

and we see that the machine interval is included within the interval displayed. ■

In the *IntervalArithmetic* unit there are also three standard (basic) Delphi Pascal functions for calculating the interval sine, the interval cosine and the interval exponential function. All these algorithms are based on the relevant Taylor series, i.e.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \pm \ldots ,$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots + (-1)^n \frac{x^{2n}}{(2n)!} \pm \ldots ,$$

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + \ldots ,$$

where $|x| < \infty$. In the functions *isin*, *icos* and *iexp* the argument x is an interval. The subsequent interval terms of the series are added until the relative error between the sum calculated so far and the next term is less than 10^{-18} or the number of terms added are greater than ***MaxInt***/2, where ***MaxInt*** is a constant predefined in the Delphi Pascal language. In the second case the output parameter *st* is equal to 2. In a similar way one can write other standard interval functions.

The function *isqr* calculates the square of a given interval on the basis of (1.8).

At the end of the unit presented there are given the functions *isqrt2*, *isqrt3*, *isqrt5*, *isqrt6*, *isqrt7*, *isqrt8*, *isqrt10* and *ipi* that return intervals containing $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{6}$, $\sqrt{7}$, $\sqrt{8}$, $\sqrt{10}$ and π, respectively.

Chapter 2

The Initial Value Problem

The *initial value problem* consists in finding the function $y = y(t)$, such that

$$y' = f(t, y(t)), \quad y(0) = y_0, \tag{2.1}$$

where $t \in [0, a]$, $y \in \mathbf{R}^N$ and f: $[0, a] \times \mathbf{R}^N \to \mathbf{R}^N$. If $t \in [t_0, t_0 + a]$, then introducing a new independent variable $\tau = t - t_0$ we have

$$y' = \frac{dy}{dt} = \frac{dy}{d\tau}\frac{d\tau}{dt} = \frac{dy}{d\tau} = f(\tau, y(\tau)),$$

where $\tau \in [0, a]$. It means that in such a case we obtain an equation of the form (2.1). Further, we will assume that the solution to the initial value problem (2.1) exists and is unique.

From the theory of ordinary differential equations it is known that the solution to (2.1) exists and is unique if the function f is determined and continuous with respect to t and there exists a constant $L > 0$, called the Lipschitz constant, such that for each $t \in [0, a]$ and all $y_1, y_2 \in \mathbf{R}^N$ we have

$$\| f(t, y_1) - f(t, y_2) \| \le L \| y_1 - y_2 \|.$$

In a lot of monographs and lecture books on differential equations, also in those concerning numerical methods for solving the initial value problem, other well-known theorems are presented that guarantee the existence and uniqueness of the solution to (2.1) [38, 62].

Below we present a few examples of the initial value problem that will be used to verify the interval methods presented in the next chapters.

Example 2.1

The commonly used test problem is the following:

$$y' = \lambda y, \quad y(0) = 1. \tag{2.2}$$

This problem has the exact solution of the form

$$y = \exp(\lambda t).$$

For $\lambda = 0.5$ and $t = 0.1 \cdot i$, where $i = 1, 2, \ldots, 10$, the numerical values of the solution are given in Table 2.1. A graph of the solution is presented in Figure 2.1. ■

Table 2.1. The approximate (with 15 digits after decimal point) exact solution to (2.2) with λ = 0.5

t	$y(t)$	t	$y(t)$
0.1	1.051271096376024	0.6	1.348858807576003
0.2	1.105170918075648	0.7	1.419067548593257
0.3	1.161834242728283	0.8	1.491824697641270
0.4	1.221402758160170	0.9	1.568312185490169
0.5	1.284025416687742	1.0	1.648721270700128

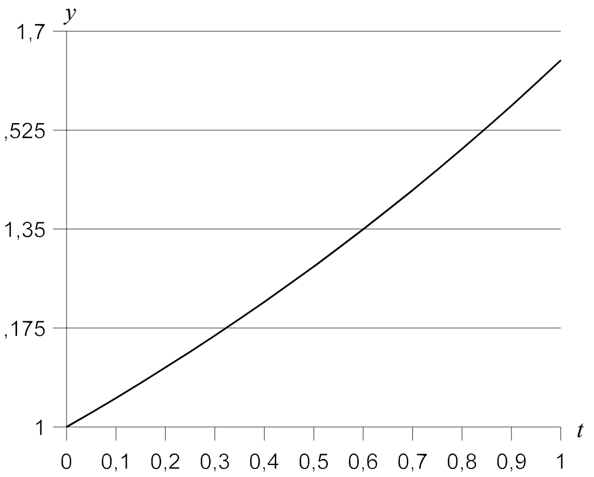


Figure 2.1. The solution to (2.2) with λ = 0.5

Example 2.2

The motion of the Moon in a rotating coordinate system is described by the Hill equations of the form [37, 69, 117, 180]

$$\begin{aligned}
&\frac{dy_l}{d\tau} = y_{l+3}, \quad l = 1, 2, 3,\\
&\frac{dy_4}{d\tau} = 2My_5 - \left(\frac{\kappa}{r^3} - 3M^2\right) y_1,\\
&\frac{dy_5}{d\tau} = -2My_4 - \frac{\kappa}{r^3} y_2,\\
&\frac{dy_6}{d\tau} = -\left(\frac{\kappa}{r^3} + M^2\right) y_3,
\end{aligned} \tag{2.3}$$

where

$$r = \sqrt{y_1^2 + y_2^2 + y_3^2}, \quad \tau = (\nu - \nu')(t - t_0),$$
$$M = \frac{\nu'}{\nu - \nu'}, \quad \kappa = G\frac{m_0 + m_1}{(\nu - \nu')^2},$$

and where ν is the mean motion of the Moon, ν' – the mean motion of the Sun, G – the gravitational constant, m_0 – the mass of the Earth, m_1 is the mass of the Moon, and t_0 is an initial moment. For the equations (2.3) we can formulate the initial conditions as follows:

$$y_l(\tau_0) = y_l^0, \quad l = 1, 2, \ldots, 6. \tag{2.4}$$

Of course, we can assume that $\tau_0 = 0$ and consider the equation (2.3) in an interval $[0, T]$.

If we consider the motion on a plane, then the equations (2.3) – (2.4) are as follows:

$$\begin{aligned}
&\frac{dy_l}{d\tau} = y_{l+2}, \quad l = 1, 2,\\
&\frac{dy_3}{d\tau} = 2My_4 - \left(\frac{\kappa}{r^3} - 3M^2\right) y_1,\\
&\frac{dy_4}{d\tau} = -2My_3 - \frac{\kappa}{r^3} y_2,\\
&y_l(\tau_0) = y_l^0, \quad l = 1, 2, 3, 4.
\end{aligned} \tag{2.5}$$

If we take $M = 0$, then the initial value problem (2.5) has the following solution:

$$\begin{aligned}
y_1 &= \sqrt[3]{\kappa}\cos\tau, \quad y_2 = \sqrt[3]{\kappa}\sin\tau,\\
y_3 &= -\sqrt[3]{\kappa}\sin\tau, \quad y_4 = \sqrt[3]{\kappa}\cos\tau.
\end{aligned} \tag{2.6}$$

For $\kappa = 1$, $\tau_0 = 0, y_1(0) = 1, y_2(0) = 0, y_3(0) = 0$ and $y_4(0) = 1$ the numerical values of the solution at some moments τ are given in Table 2.2. The obtained orbit is presented in Figure 2.2. ■

Table 2.2. The approximate (with 15 digits after decimal point) exact solution of (2.5) with $M = 0$ and $\kappa = 1$

τ	$y_1(\tau) = y_4(\tau)$	$y_2(\tau) = -y_3(\tau)$
0.05	0.998750260394966	0.049979169270678
0.1	0.995004165278026	0.099833416646828
1.0	0.540302305868140	0.841470984807897
2.0	−0.416146836547142	0.909297426825682
3.0	−0.989992496600446	0.141120008059867
4.0	−0.653643620863612	−0.756802495307928
5.0	0.283662185463226	−0.958924274663139
6.0	0.960170286650366	−0.279415498198926

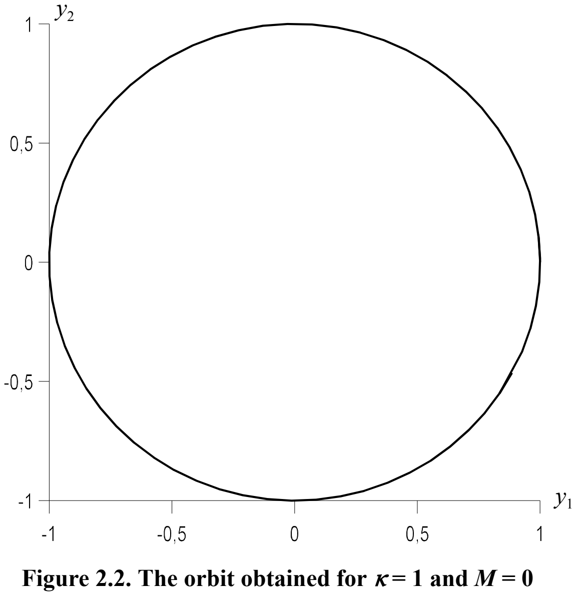


Figure 2.2. The orbit obtained for $\kappa = 1$ and $M = 0$

Example 2.3

Let us consider the following system of first-order differential equations:

$$y_1' = 3y_1 + 2y_2, \quad y_2' = 4y_1 + y_2, \tag{2.7}$$

with the initial conditions

$$y_1(0) = 0, \quad y_2(0) = 1. \tag{2.8}$$

The solution of the problem (2.7) – (2.8) is of the form

$$y_1(t) = \frac{\exp(5t) - \exp(-t)}{3}, \quad y_2(t) = \frac{\exp(5t) + 2\exp(-t)}{3}. \tag{2.9}$$

Numerical values of the functions y_1(t) and y_2(t) for some t-values are given in Table 2.3, and a graph of both functions are presented in Figure 2.3. ■

Table 2.3. The approximate (with 15 digits after decimal point) exact solution of the initial value problem (2.7) – (2.8)

t	$y_1(t)$	$y_2(t)$
0.05	0.110931997395676	1.062161421896390
0.10	0.247961284221390	1.152798702257349
0.15	0.418764013395872	1.279471989820930
0.20	0.633183691793688	1.451914444871670

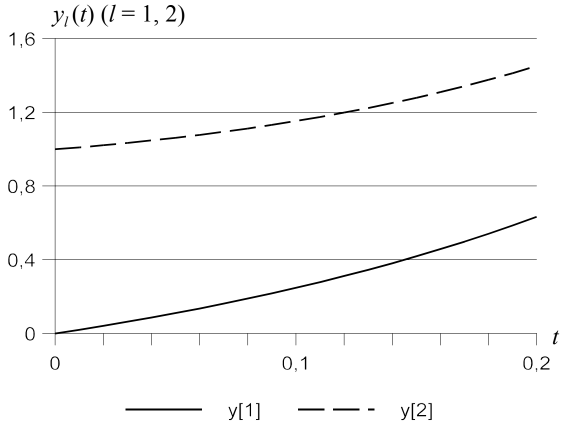


Figure 2.3. The solutions y[1] ≡ y_1(t) and y[2] ≡ y_2(t) of the problem (2.7) – (2.8)

Example 2.4

Let us consider the motion of a simple pendulum described by the equation

$$\varphi'' + u^2 \sin\varphi = 0, \tag{2.10}$$

where $\varphi = \varphi(t)$, $u = \sqrt{g / L}$, and where g is the gravitational acceleration at the Earth's surface and L denotes the pendulum length. If we assume that the angle φ is small, i.e. $\sin\varphi \approx \varphi$, then the equation (2.10) can be reduced to the equation of simple harmonic motion

$$\varphi'' + u^2\varphi = 0 \tag{2.11}$$

with the solution $\varphi(t) = \varphi_0 \cos(ut)$, where φ_0 is an initial angle. Denoting $y_1 = \varphi'$, $y_2 = \varphi$ and assuming that $\varphi'(0) = 0$, $\varphi(0) = \varphi_0$, we can transform (2.11) into the following systems of differential equations of the first order:

$$y_1' = -u^2 y_2, \quad y_2' = y_1 \tag{2.12}$$

with the initial conditions

$$y_1(0) = 0, \quad y_2(0) = \varphi_0. \tag{2.13}$$

For $g = 9.80665$, $L = 1$ and $\varphi_0 = \pi / 6$ the exact solution is presented in Table 2.4 and in Figure 2.4. ■

Table 2.4. The approximate (with 15 digits after decimal point) exact solution of the initial value problem (2.12) – (2.13)

t	$y_1(t)$	$y_2(t)$
0.05	−0.255689725696726	0.517193440672640
0.1	−0.505123598987129	0.498134152516948
0.5	−1.639658832231953	0.002627285350445
1.0	−0.016454781143167	−0.523572409500308
1.5	1.639493700427002	−0.007881591454600
2.0	0.032907905107624	0.523493313861694

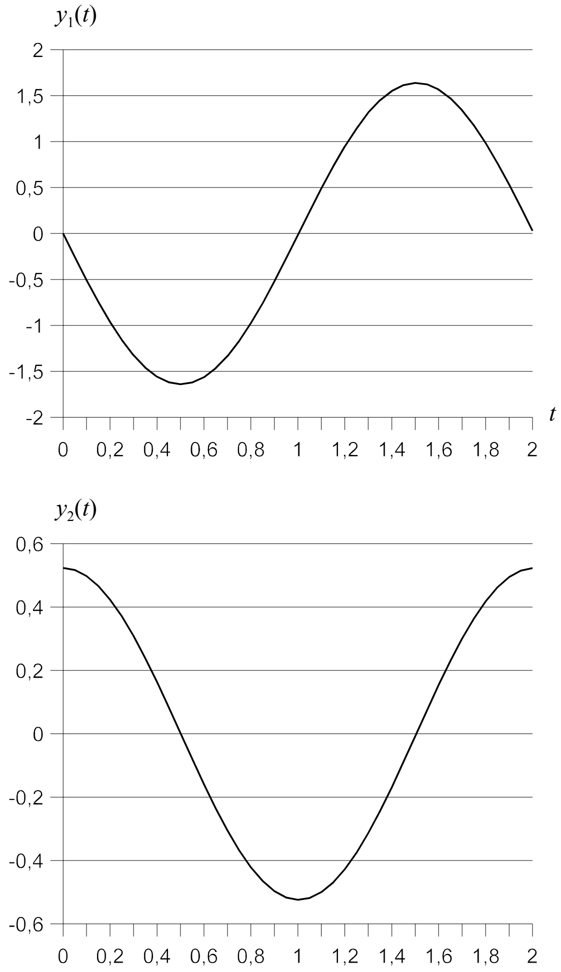


Figure 2.4. The solution of the problem (2.12) – (2.13)

Chapter 3

Interval Methods of Runge-Kutta Type

3.1. Conventional Runge-Kutta Methods

Runge-Kutta methods have been originally developed by C. Runge [165] towards the end of the nineteenth century and generalized by W. Kutta [101] in the early twentieth century. Although early studies were devoted entirely to the so-called *explicit Runge-Kutta methods*, further interest extended to *implicit* methods which are recognized as appropriate for so-called *stiff differential equations* [38, 63].

In order to construct Runge-Kutta methods one should consider the relation

$$y(t) = y(t_k) + \int_{t_k}^{t} f(\tau, y(\tau))d\tau, \quad t > t_k, \tag{3.1}$$

which is equivalent to the differential equation

$$y' = f(t, y)$$

for $t > t_k$. If in (3.1) we take $t_{k+1} = t_k + h$ instead of t, we get

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))d\tau. \tag{3.2}$$

Changing variables, from (3.2) we have

$$y(t_{k+1}) = y(t_k) + h\int_{0}^{1} f(t_k + ch, y(t_k + ch))dc,$$

where $h = t_{k+1} - t_k$. If we substitute the integral else for a sum we obtain

$$y(t_{k+1}) = y(t_k) + h\sum_{i=1}^{m} w_i f(t_k + c_i h, y(t_k + c_i h)) + E_m(h), \tag{3.3}$$

where w_i and c_i are some coefficients and $E_m(h)$ is an approximation error. If we assume that $c_1 = 0$, then the first term in the sum occurring in (3.3) is of the form $w_1 f(t_k, y(t_k))$. Taking into account the second term with $f(t_k + c_2 h, y(t_k + c_2 h))$ and the fact that

$$y(t_k + c_2 h) = y(t_k) + hc_2 f(t_k, y(t_k)) + \ldots,$$

we can substitute $y(t_k + c_2 h)$ else for $y(t_k) + hc_2 f(t_k, y(t_k))$.

We can apply the same procedure to the next terms of the sum occurring in (3.3). If we denote

$$\kappa_1(h) = f(t_k, y(t_k)),$$
$$\kappa_i(h) = f\left(t_k + c_i h, y(t_k) + h\sum_{j=1}^{i-1} a_{ij}\kappa_j(h)\right), \quad i > 1,$$

where

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i > 1, \quad c_1 = 0, \tag{3.4}$$

or, in another way, $a_{i1} = c_2$, $a_{i2} = c_3 - c_2$, ... , $a_{i,i-1} = c_i - c_{i-1}$, then the equation (3.3) can be rewritten in the form

$$y(t_{k+1}) = y(t_k) + h\sum_{i=1}^{m} w_i \kappa_i(h) + R_m(h), \tag{3.5}$$

where the error $R_m(h)$ includes $E_m(h)$ and the errors following from the approximations of $y(t_k + c_i h)$ by the first two terms of the Taylor series. If we omit $R_m(h)$ and substitute an approximation y_k for the exact value $y(t_k)$, then from (3.5) we can determine y_{k+1} which is an approximation of $y(t_{k+1})$. This leads to the *explicit m-stage Runge-Kutta methods* given by the following formula:

$$y_{k+1} = y_k + h\sum_{i=1}^{m} w_i \kappa_{ik}, \tag{3.6}$$

where

$$\kappa_{1k} = f(t_k, y_k),$$

$$\kappa_{ik} = f\left(t_k + c_i h, y_k + h \sum_{j=1}^{i-1} a_{ij} \kappa_{jk} \right), \quad i = 2, 3, \ldots, m, \tag{3.7}$$

and where the coefficients w_i, c_i and a_{ij} are some parameters. It is convenient to present these coefficients in a form of an array, called the *Butcher table*:

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & & & \\
c_m & a_{m1} & a_{m2} & \cdots & a_{m,\, m-1} & \\
\hline
 & w_1 & w_2 & \cdots & w_{m-1} & w_m
\end{array}$$

If we do not assume that $c_1 = 0$, then from (3.3) we can get more general, *implicit m-stage Runge-Kutta methods* in which

$$\kappa_{ik} = f\left(t_k + c_i h, y_k + h \sum_{j=1}^{m} a_{ij} \kappa_{jk} \right), \quad i = 1, 2, \ldots, m, \tag{3.8}$$

where

$$c_i = \sum_{j=1}^{m} a_{ij}. \tag{3.9}$$

In this case the Butcher table is of the following form:

$$\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1m} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2m} \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
c_m & a_{m1} & a_{m2} & \cdots & a_{mm} \\
\hline
 & w_1 & w_2 & \cdots & w_m
\end{array}$$

The *local truncation error* of step $k + 1$ for a Runge-Kutta method (explicit and implicit) of order p can be written in the form [38, 41, 62, 77, 97]

$$\begin{aligned} r_{k+1}(h) &= \psi(t_k, y(t_k))h^{p+1} + O(h^{p+2}) = \\ &= r_{k+1}^{(p+1)}(0)\frac{h^{p+1}}{(p+1)!} + r_{k+1}^{(p+1)}(\theta h)\frac{h^{p+2}}{(p+2)!}, \quad 0 < \theta < 1. \end{aligned} \tag{3.10}$$

From the conditions $r_{k+1}^{(l)}(0) = 0$ (for $l = 1, 2, \ldots, p$) follow the equations for determining the coefficients w_i, c_i and a_{ij}. Unfortunately, there are fewer equations than the number of unknowns, and usually we consider some special cases.

It can be proved [38, 62, 77] that if $p_{\max}(m)$ denotes the maximum order of the m-stage explicit Runge-Kutta method, then we have

$$\begin{aligned} p_{\max}(m) &= m, \quad m = 1, 2, 3, 4, \\ p_{\max}(m) &= m - 1, \quad m = 5, 6, 7, \\ p_{\max}(m) &= m - 2, \quad m = 8, 9, \\ p_{\max}(m) &\le m - 2, \quad m \ge 10. \end{aligned}$$

In the case of implict Runge-Kutta methods for each m there exists a method with maximum order $p = 2m$.

The simplest explicit Runge-Kutta method we get for $m = 1$. In this case we have $w_1 = 1$ and the method is of the form

$$y_{k+1} = y_k + hf(t_k, y_k), \tag{3.11}$$

which is called the *Euler method*.

If $m = 2$, then we have the following equations for the coefficients [38, 62, 77, 97]:

$$w_1 + w_2 = 1, \quad w_2 c_2 = \frac{1}{2}.$$

It is obvious that the cases $w_2 = 0$ and $c_2 = 0$ are impossible. If we take $c_2 \neq 0$ as a parameter, then

$$w_1 = \frac{2c_2 - 1}{2c_2}, \quad w_2 = \frac{1}{2c_2}. \tag{3.12}$$

Thus, we have the infinite number of two-stage methods of the second order. Two most popular methods are as follows:

- $c_2 = \frac{1}{2}$

$$\begin{aligned} &y_{k+1} = y_k + h\kappa_{2k}, \\ &\kappa_{1k} = f(t_k, y_k), \quad \kappa_{2k} = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}\kappa_{1k}\right), \end{aligned} \tag{3.13}$$

which is called the *Euler improved method*,

- $c_2 = 1$

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{2}(\kappa_{1k} + \kappa_{2k}), \\ \kappa_{1k} &= f(t_k, y_k), \quad \kappa_{2k} = f(t_k + h, y_k + h\kappa_{1k}), \end{aligned} \tag{3.14}$$

which is called the *Euler-Cauchy method*.

If $m = 4$, then we get the following system of equations for the parameters (see e.g. [97]):

$$\begin{aligned} & w_1 + w_2 + w_3 + w_4 = 1, \\ & w_2c_2 + w_3c_3 + w_4c_4 = \frac{1}{2}, \\ & w_2c_2^2 + w_3c_3^2 + w_4c_4^2 = \frac{1}{3}, \\ & w_3a_{32}c_2 + w_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{6}, \\ & w_2c_2^3 + w_3c_3^3 + w_4c_4^3 = \frac{1}{4}, \\ & w_3c_2c_3a_{32} + w_4c_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{8}, \\ & w_3a_{32}c_2^2 + w_4(a_{42}c_2^2 + a_{43}c_3^2) = \frac{1}{12}, \\ & w_4a_{43}a_{32}c_2 = \frac{1}{24}. \end{aligned}$$

It is self-evident that the coefficients w_4, a_{43}, a_{32} and c_2 must be different from zero. If we take c_2 and c_3 as parameters, then we get a two-parameter family of solutions. If we assume that $c_2 = c_3$ and $c_4 = 1$, then we obtain a family with the following Butcher table:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & \frac{3t-1}{6t} & \frac{1}{6t} & & \\ 1 & 0 & 1-3t & 3t & \\ \hline & \frac{1}{6} & \frac{2}{3}-t & t & \frac{1}{6} \end{array}$$

For $t = 1/3$ we obtain one of the most popular Runge-Kutta methods of the fourth order, simply called the *Runge-Kutta method*:

$$
\begin{aligned}
&y_{k+1} = y_k + \frac{h}{6}(\kappa_{1k} + 2\kappa_{2k} + 2\kappa_{3k} + \kappa_{4k}),\\
&\kappa_{1k} = f(t_k, y_k), \quad \kappa_{2k} = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}\kappa_{1k}\right),\\
&\kappa_{3k} = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}\kappa_{2k}\right), \quad \kappa_{4k} = f(t_k + h, y_k + h\kappa_{3k}).
\end{aligned}
\tag{3.15}
$$

In order to reduce the number of equations for coefficients in the case of implicit Runge-Kutta methods, one can consider the following methods:

- *semi-implicit*

$$
\begin{array}{c|cccc}
c_1 = a_{11} & a_{11} & 0 & \dots & 0 \\
c_2 & a_{21} & a_{22} & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots \\
c_m & a_{m1} & a_{m2} & \dots & a_{mm} \\
\hline
 & w_1 & w_2 & \dots & w_m
\end{array}
$$

- *diagonally implicit*

$$
\begin{array}{c|cccc}
c_1 = a_{11} & a_{11} & 0 & \dots & 0 \\
c_2 & a_{21} & a_{22} = a_{11} & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots \\
c_m & a_{m1} & a_{m2} & \dots & a_{mm} = a_{11} \\
\hline
 & w_1 & w_2 & \dots & w_m
\end{array}
$$

- *symmetric* $(a_{m-i+1,\, m-j+1} + a_{ij} = w_{m-j+1} = w_j)$

$$
\begin{array}{c|ccccc}
c_1 & a_{11} & a_{12} = w_2 - a_{m,\, m-1} & \dots & a_{1,\, m-1} = w_2 - a_{m2} & a_{1m} = w_1 - a_{m1} \\
c_2 & a_{21} & a_{22} & \dots & a_{2,\, m-1} = w_2 - a_{m-1,\, 2} & a_{2m} = w_1 - a_{m-1,\, 1} \\
\vdots & \vdots & \vdots & \dots & \vdots & \vdots \\
c_m & a_{m1} & a_{m2} & \dots & a_{m,\, m-1} & a_{mm} \\
\hline
 & w_1 & w_2 & \dots & w_{m-1} = w_2 & w_m = w_1
\end{array}
$$

The simplest implicit Runge-Kutta method we obtain when m =1. In this case $w_1 = 1, c_1 = a_{11} = 1/2$ and we get the method of the second order, called the *implicit midpoint rule*:

$$
\begin{aligned}
&y_{k+1} = y_k + h\kappa_{1k},\\
&\kappa_{1k} = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}\kappa_{1k}\right).
\end{aligned}
\tag{3.16}
$$

If we take m = 2, then from the conditions $r_{k+1}^{(l)}(0) = 0$ for l = 1, 2, 3, 4 (see (3.10)) we get the following equations to determine the coefficients (see e.g. [97]):

$$
\begin{aligned}
&w_1 + w_2 = 1,\\
&w_1c_1 + w_2c_2 = \frac{1}{2},\\
&w_1c_1^2 + w_2c_2^2 = \frac{1}{3},\\
&w_1(a_{11}c_1 + a_{12}c_2) + w_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{6},\\
&w_1c_1^3 + w_2c_2^3 = \frac{1}{4},\\
&w_1c_1(a_{11}c_1 + a_{12}c_2) + w_2c_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{8},\\
&w_1(a_{11}c_1^2 + a_{12}c_2^2) + w_2(a_{21}c_1^2 + a_{22}c_2^2) = \frac{1}{12},\\
&(w_1a_{11} + w_2a_{21})(a_{11}c_1 + a_{12}c_2) + (w_1a_{12} + w_2a_{22})(a_{21}c_1 + a_{22}c_2) = \frac{1}{24}.
\end{aligned}
$$

Solving this equations we get an implicit Runge-Kutta method of the fourth order, called the *Hammer-Hollingsworth method*, with the Butcher table of the form:

$$
\begin{array}{c|cc}
\frac{1}{2} \mp \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} \mp \frac{\sqrt{3}}{6}\\
\frac{1}{2} \pm \frac{\sqrt{3}}{6} & \frac{1}{4} \pm \frac{\sqrt{3}}{6} & \frac{1}{4}\\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
\tag{3.17}
$$

If we consider the case of $m = 2$ and $p = 3$, then for the semi-implicit methods we get the following equations for determining the coefficients:

$$
\begin{aligned}
&w_1 + w_2 = 1,\\
&w_1c_1 + w_2c_2 = \frac{1}{2},\\
&w_1c_1^2 + w_2c_2^2 = \frac{1}{3},\\
&w_1a_{11}c_1 + w_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{6},\\
&c_1 = a_{11},\\
&c_2 = a_{21} + a_{22},\\
&a_{12} = 0.
\end{aligned}
\tag{3.18}
$$

Assuming that c_1 is a parameter and $2c_1 - 1 \neq 0$, $c_1 - c_2 \neq 0$ we get a one-parameter family of solutions with the Butcher table of the form

$$
\begin{array}{c|cc}
c_1 & c_1 & 0 \\
\frac{1}{2} - \frac{1}{6(2c_1 - 1)} & \frac{1}{3(1 - 2c_1)} & \frac{1}{2} - \frac{1}{2(2c_1 - 1)} \\
\hline
 & 1 - \dfrac{2c_1 - 1}{2\left(c_1 - \frac{1}{2} - \frac{1}{6(2c_1 - 1)}\right)} & \dfrac{2c_1 - 1}{2\left(c_1 - \frac{1}{2} - \frac{1}{6(2c_1 - 1)}\right)}
\end{array}
$$

Taking $c_1 = 1$ we have the following semi-implicit method of the third order:

$$
\begin{aligned}
&y_{k+1} = y_k + \frac{h}{4}(\kappa_{1k} + 3\kappa_{2k}),\\
&\kappa_{1k} = f(t_k + h, y_k + h\kappa_{1k}), \quad \kappa_{2k} = f\left(t_k + \frac{h}{3}, y_k - \frac{h}{3}\kappa_{1k} + \frac{2h}{3}\kappa_{2k}\right).
\end{aligned}
\tag{3.19}
$$

Adding to (3.18) the equation

$$a_{22} = a_{11}$$

we get two diagonally implicit methods of the third order (called the *Alexander methods*) with the following Butcher table:

$$\begin{array}{c|cc}
\frac{1}{2} \pm \frac{\sqrt{3}}{6} & \frac{1}{2} \pm \frac{\sqrt{3}}{6} & 0 \\[2ex]
\frac{1}{2} \mp \frac{\sqrt{3}}{6} & \mp \frac{\sqrt{3}}{3} & \frac{1}{2} \pm \frac{\sqrt{3}}{6} \\[1ex]
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array} \tag{3.20}$$

In order to obtain two-stage ($m = 2$) symmetric Runge-Kutta methods of the third order ($p = 3$) we should solve the system of equations of the form:

$$\begin{aligned}
&w_1 + w_2 = 1,\\
&w_1c_1 + w_2c_2 = \frac{1}{2},\\
&w_1c_1^2 + w_2c_2^2 = \frac{1}{3},\\
&w_1(a_{11}c_1 + a_{12}c_2) + w_2(a_{21}c_1 + a_{22}c_2) = \frac{1}{6},\\
&c_1 = a_{11} + a_{12} = a_{11} + w_1 - a_{21},\\
&c_2 = a_{21} + a_{22},\\
&w_1 = w_2.
\end{aligned}$$

from which we get the method with the following Butcher table:

$$\begin{array}{c|cc}
\frac{1}{2} \pm \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} \pm \frac{\sqrt{3}}{6} \\[2ex]
\frac{1}{2} \mp \frac{\sqrt{3}}{6} & \frac{1}{4} \mp \frac{\sqrt{3}}{6} & \frac{1}{4} \\[1ex]
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array} \tag{3.21}$$

Let us note that in fact the method (3.21) is the Hammer-Hollingsworth method (3.17) of the fourth order.

An example of a three-stage semi-implict method of the fourth order is the Butcher method with the following table:

$$\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\
0 & 0 & 1 & 0 \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array} \tag{3.22}$$

For $m = 3$ and $p = 4$ we can obtain *diagonally implicit methods of Alexander* with the Butcher table of the form (see e.g. [38]):

$$\begin{array}{c|ccc}
\frac{1}{2}+\frac{\sqrt{3}}{3}\zeta & \frac{1}{2}+\frac{\sqrt{3}}{3}\zeta & 0 & 0 \\
\frac{1}{2} & -\frac{\sqrt{3}}{3}\zeta & \frac{1}{2}+\frac{\sqrt{3}}{3}\zeta & 0 \\
\frac{1}{2}-\frac{\sqrt{3}}{3}\zeta & 1+\frac{2\sqrt{3}}{3}\zeta & -1-\frac{4\sqrt{3}}{3}\zeta & \frac{1}{2}+\frac{\sqrt{3}}{3}\zeta \\
\hline
 & \frac{1}{8\zeta^2} & 1-\frac{1}{4\zeta^2} & \frac{1}{8\zeta^2}
\end{array} \tag{3.23}$$

where $\zeta = \cos 10°, -\cos 50°$ or $-\cos 70°$.

Also known are *symplectic Runge-Kutta methods*. In general, symplectic methods (not only the Runge-Kutta methods) concern the Hamilton equations

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \quad \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \quad i = 1, 2, \ldots, N,$$

where $H = H(p_1, \ldots, p_N, q_1, \ldots, q_N)$. The coefficients of symplectic Runge-Kutta methods fulfill the conditions [166]

$$w_i a_{ij} + w_j a_{ji} = w_i w_j, \quad i, j = 1, 2, \ldots, m. \tag{3.24}$$

Taking into account (3.24) we can significantly reduce the number of equations for finding the coefficients. It should be noted that all symplectic Runge-Kutta methods are implicit and that there is no relation between symmetric and symplectic Runge--Kutta methods.

The implicit midpoint method (3.16) and the Hammer-Hollingsworth method (3.17) are examples of symplectic Runge-Kutta methods.

The function $\psi(t, y) \equiv \psi(t, y(t))$, occurring in (3.10), depends on coefficients w_i, c_i, a_{ij} and on partial derivatives of $f(t, y)$. The form of $\psi(t, y)$ is very complicated and cannot be written in a general form for an arbitrary p. Since this form is very important from the point of view of interval methods developed, below we present adequate formulas.

For the explicit Runge-Kutta methods we have:

- $p = m = 1$

$$\psi_s(t, y) = \frac{1}{2} y_s'', \tag{3.25}$$

- $p = m = 2$

$$\psi_s(t, y) = \frac{1}{6}\left(1 - 3w_2c_2^2\right) y_s''' + \frac{1}{2} w_2c_2^2 \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k'', \tag{3.26}$$

- $p = m = 3$

$$\begin{aligned}\psi_s(t, y) = {} & \frac{1}{24}\left[1 - 4\left(w_2c_2^3 + w_3c_3^3\right)\right] y_s^{\mathrm{IV}} + \\ & + \frac{1}{6}\left(w_2c_2^3 + w_2c_3^3 - 3w_2c_3^2a_{32}\right) \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k''' + \\ & + \frac{1}{2}\left(w_2c_2^3 + w_2c_3^3 - 2w_3c_2c_3a_{32}\right) \sum_{k=1}^{N} \left(\frac{\partial^2 f_s}{\partial t\, \partial y_k} + \sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k\, \partial y_l} f_l \right) y_k'' + \\ & + \frac{1}{2} w_3c_2^2a_{32} \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} \sum_{l=1}^{N} \frac{\partial f_k}{\partial y_l} y_l'',\end{aligned}$$

- $p = m = 4$

$$\begin{aligned}\psi_s(t, y) = {} & \frac{1}{120}\left[1 - 5\left(w_2c_2^4 + w_3c_3^4 + w_4c_4^4\right)\right] y_s^{\mathrm{V}} + \\ & + \frac{1}{24}\left[w_2c_2^4 + w_3c_3^4 + w_4c_4^4 - 4w_3c_2^3a_{32} - 4w_4\left(c_2^3a_{42} + c_3^3a_{43}\right)\right] \times \\ & \times \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_s^{\mathrm{IV}} +\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{6}\left[w_2c_2^4+w_3c_3^4+w_4c_4^4-3w_3c_2^2c_3a_{32}-3w_4c_4\left(c_2^2a_{42}+c_3^2a_{43}\right)\right]\times\\
&\quad\times\sum_{k=1}^{N}\left(\frac{\partial^2 f_s}{\partial t\,\partial y_k}+\sum_{l=1}^{N}\frac{\partial^2 f_s}{\partial y_k\,\partial y_l}f_l\right)y_k'''+\\
&+\frac{1}{6}\left[w_3c_2^3a_{32}+w_4\left(c_2^3a_{42}+c_3^3a_{43}\right)-3w_4c_2^2a_{32}a_{43}\right]\times\\
&\quad\times\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}\sum_{l=1}^{N}\frac{\partial f_k}{\partial y_l}y_l'''+\\
&+\frac{1}{4}\left[w_2c_2^4+w_3c_3^4+w_4c_4^4-2w_3c_2c_3^2a_{32}-2w_4c_4^2\left(c_2a_{42}+c_3a_{43}\right)\right]\times\\
&\quad\times\sum_{k=1}^{N}\left[\frac{\partial^3 f_s}{\partial t^2\,\partial y_k}+\sum_{l=1}^{N}\left(2\frac{\partial^3 f_s}{\partial t\,\partial y_k\,\partial y_l}+\sum_{p=1}^{N}\frac{\partial^3 f_s}{\partial y_k\,\partial y_l\,\partial y_p}f_p\right)f_l\right]y_k''+\\
&+\frac{1}{2}\left[w_3c_2^3a_{32}+w_4c_2^3a_{42}+w_4c_3a_{43}\left(c_3^2-2c_2a_{32}\right)\right]\times\\
&\quad\times\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}\sum_{l=1}^{N}\left(\frac{\partial^2 f_k}{\partial t\,\partial y_l}y_l''+\sum_{p=1}^{N}\frac{\partial^2 f_k}{\partial y_l\,\partial y_p}y_p''f_l\right)+\\
&+\frac{1}{2}\left[w_3c_2^2c_3a_{32}+w_4c_2^2c_4a_{42}+w_4c_4a_{43}\left(c_3^2-2c_2a_{32}\right)\right]\times\\
&\quad\times\sum_{k=1}^{N}\left(\frac{\partial^2 f_s}{\partial t\,\partial y_k}\sum_{l=1}^{N}\frac{\partial f_k}{\partial y_l}y_l''+\sum_{l=1}^{N}\frac{\partial^2 f_s}{\partial y_k\,\partial y_l}\sum_{p=1}^{N}\frac{\partial f_l}{\partial y_p}y_p''f_k\right)+\\
&+\frac{1}{8}\left[w_2c_2^4+w_3c_3^4+w_4c_4^4-4w_3c_2^2a_{32}^2-4w_4\left(c_2a_{42}+c_3a_{43}\right)^2\right]\times\\
&\quad\times\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{\partial^2 f_s}{\partial y_k\,\partial y_l}y_l''y_k''+\\
&+\frac{1}{2}w_4c_2^2a_{32}a_{43}\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}\sum_{l=1}^{N}\frac{\partial f_k}{\partial y_l}\sum_{p=1}^{N}\frac{\partial f_l}{\partial y_p}y_p'',
\end{aligned}
\qquad (3.27)
$$

where

$$y_s'' = \frac{\partial f_s}{\partial t} + \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} f_k,$$

$$y_s''' = \frac{\partial^2 f_s}{\partial t^2} + 2\sum_{k=1}^{N} \frac{\partial^2 f_s}{\partial t\,\partial y_k} f_k + \sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k\,\partial y_l} f_l\, f_k + \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k'',$$

$$\begin{aligned} y_s^{\mathrm{IV}} = {} & \frac{\partial^3 f_s}{\partial t^3} + 3\sum_{k=1}^{N} \frac{\partial^3 f_s}{\partial t^2\,\partial y_k} f_k + 3\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^3 f_s}{\partial t\,\partial y_k\,\partial y_l} f_l\, f_k + \\ & + \sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{p=1}^{N} \frac{\partial^3 f_s}{\partial y_k\,\partial y_l\,\partial y_p} f_p\, f_l\, f_k + 3\sum_{k=1}^{N} \frac{\partial^2 f_s}{\partial t\,\partial y_k} y_k + \\ & + 3\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k\,\partial y_l} f_l\, y_k'' + \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k''', \end{aligned}$$

$$\begin{aligned} y_s^{\mathrm{V}} = {} & \frac{\partial^4 f_s}{\partial t^4} + 4\sum_{k=1}^{N} \frac{\partial^3 f_s}{\partial t^3\,\partial y_k} f_k + 6\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^4 f_s}{\partial t^2\,\partial y_k\,\partial y_l} f_l\, f_k + \\ & + 4\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{p=1}^{N} \frac{\partial^4 f_s}{\partial t\,\partial y_k\,\partial y_l\,\partial y_p} f_p\, f_l\, f_k + \\ & + \sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N} \frac{\partial^4 f_s}{\partial y_k\,\partial y_l\,\partial y_p\,\partial y_q} f_q\, f_p\, f_l\, f_k + \\ & + 12\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^3 f_s}{\partial t\,\partial y_k\,\partial y_l} y_l'' f_k + 6\sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{p=1}^{N} \frac{\partial^3 f_s}{\partial y_k\,\partial y_l\,\partial y_p} y_p''\, f_l\, f_k + \\ & + 6\sum_{k=1}^{N} \frac{\partial^3 f_s}{\partial t^2\,\partial y_k} y_k'' + 3\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k\,\partial y_l} y_l'' y_k'' + 4\sum_{k=1}^{N} \frac{\partial^2 f_s}{\partial t\,\partial y_k} y_k''' + \\ & + 4\sum_{k=1}^{N}\sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k\,\partial y_l} y_l''' f_k + \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k^{\mathrm{IV}}, \end{aligned}$$

$$f_s = f_s(t, y) = f_s(t, y_1(t), y_2(t), \ldots, y_N(t)), \quad s = 1, 2, \ldots, N.$$

For the implicit Runge-Kutta methods the function $\psi(t, y)$ is of the form:

- $p = 2, m = 1$

$$\psi_s(t, y) = \frac{1}{6}\left(1 - 3w_1c_1^2\right)y_s''' - \frac{1}{2}w_1c_1^2\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}y_k'', \tag{3.28}$$

- $p = 2, m = 2$

$$\psi_s(t, y) = \frac{1}{6}\left[1 - 3\left(w_1c_1^2 + w_2c_2^2\right)\right]y_s''' +$$
$$+ \frac{1}{2}\left\{w_1\left[c_1^2 - 2\left(c_1a_{11} + c_2a_{12}\right)\right] + w_2\left[c_2^2 - 2\left(c_1a_{21} + c_2a_{22}\right)\right]\right\}\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}y_k'',$$

- $p = 3, m = 2, 3$

$$\begin{aligned}\psi_s(t, y) = {} & \frac{\alpha_m}{24}y_s^{\mathrm{IV}} + \frac{\beta_m}{6}\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}y_k''' + \\ & + \frac{\chi_m}{2}\sum_{k=1}^{N}\left(\frac{\partial^2 f_s}{\partial t\,\partial y_k} + \sum_{l=1}^{N}\frac{\partial^2 f_s}{\partial y_k\,\partial y_l}f_l\right)y_k'' + \\ & + \frac{\delta_m}{2}\sum_{k=1}^{N}\frac{\partial f_s}{\partial y_k}\sum_{l=1}^{N}\frac{\partial f_k}{\partial y_l}y_l'',\end{aligned} \tag{3.29}$$

where

$$\alpha_m = 1 - 4\sum_{i=1}^{m}w_ic_i^3,$$

$$\beta_m = \sum_{i=1}^{m}w_i\left(c_i^3 - 3\sum_{j=1}^{m}c_j^2a_{ij}\right),$$

$$\chi_m = \sum_{i=1}^{m}w_ic_i\left(c_i^2 - 2\sum_{j=1}^{m}c_ja_{ij}\right),$$

$$\delta_m = \sum_{i=1}^{m}w_i\sum_{j=1}^{m}a_{ij}\left(c_j^2 - 2\sum_{k=1}^{m}c_ka_{jk}\right),$$

● $p = 4$, $m = 2, 3, 4$

$$\begin{aligned}
\psi_s(t, y) = {} & \frac{\varepsilon_m}{120} y_s^{\mathrm{V}} + \frac{\varphi_m}{24} \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} y_k^{\mathrm{IV}} + \\
& + \frac{\gamma_m}{6} \sum_{k=1}^{N} \left(\frac{\partial^2 f_s}{\partial t \, \partial y_k} + \sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k \, \partial y_l} f_l \right) y_k''' + \\
& + \frac{\eta_m}{6} \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} \sum_{l=1}^{N} \frac{\partial f_k}{\partial y_l} y_l''' + \\
& + \frac{\lambda_m}{4} \sum_{k=1}^{N} \left[\frac{\partial^3 f_s}{\partial t^2 \, \partial y_k} + \sum_{l=1}^{N} \left(2 \frac{\partial^3 f_s}{\partial t \, \partial y_k \, \partial y_l} + \right. \right. \\
& \qquad \left. \left. + \sum_{p=1}^{N} \frac{\partial^3 f_s}{\partial y_k \, \partial y_l \, \partial y_p} f_p \right) f_l \right] y_k'' + \\
& + \frac{\mu_m}{2} \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} \sum_{l=1}^{N} \left(\frac{\partial^2 f_k}{\partial t \, \partial y_l} y_l'' + \sum_{p=1}^{N} \frac{\partial^2 f_k}{\partial y_l \, \partial y_p} y_p'' f_l \right) + \\
& + \frac{\nu_m}{2} \sum_{k=1}^{N} \left(\frac{\partial^2 f_s}{\partial t \, \partial y_k} \sum_{l=1}^{N} \frac{\partial f_k}{\partial y_l} y_l'' + \sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k \, \partial y_l} \sum_{p=1}^{N} \frac{\partial f_l}{\partial y_p} y_p'' f_k \right) + \\
& + \frac{\theta_m}{2} \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{\partial^2 f_s}{\partial y_k \, \partial y_l} y_l'' y_k'' + \frac{\rho_m}{2} \sum_{k=1}^{N} \frac{\partial f_s}{\partial y_k} \sum_{l=1}^{N} \frac{\partial f_k}{\partial y_l} \sum_{p=1}^{N} \frac{\partial f_l}{\partial y_p} y_p'',
\end{aligned} \tag{3.30}$$

where

$$\varepsilon_m = 1 - 5 \sum_{i=1}^{m} w_i c_i^4,$$

$$\varphi_m = \sum_{i=1}^{m} w_i \left(c_i^4 - 4 \sum_{j=1}^{m} c_j^3 a_{ij} \right),$$

$$\gamma_m = \sum_{i=1}^{m} w_i c_i \left(c_i^3 - 3 \sum_{j=1}^{m} c_j^2 a_{ij} \right),$$

$$\eta_m = \sum_{i=1}^{m} w_i \sum_{j=1}^{m} a_{ij} \left(c_j^3 - 3 \sum_{k=1}^{m} c_k^2 a_{jk} \right),$$

$$\lambda_m = \sum_{i=1}^{m} w_i c_i^2 \left(c_i^2 - 2 \sum_{j=1}^{m} c_j a_{ij} \right),$$

$$\mu_m = \sum_{i=1}^{m} w_i \sum_{j=1}^{m} c_j a_{ij} \left(c_j^2 - 2 \sum_{k=1}^{m} c_k a_{jk} \right),$$

$$\nu_m = \sum_{i=1}^{m} w_i c_i \sum_{j=1}^{m} a_{ij} \left(c_j^2 - 2 \sum_{k=1}^{m} c_k a_{jk} \right),$$

$$\theta_m = \sum_{i=1}^{m} w_i \left[c_i^4 - 4 \left(\sum_{j=1}^{m} c_j a_{ij} \right)^2 \right],$$

$$\rho_m = \sum_{i=1}^{m} w_i \sum_{j=1}^{m} a_{ij} \sum_{k=1}^{m} a_{jk} \left(c_k^2 - 2 \sum_{l=1}^{m} c_l a_{kl} \right).$$

3.2. Explicit Methods

3.2.1. Basic Formulas

Let us denote:

- Δ_t and Δ_y – bounded sets in which the function $f(t, y)$, occurring in (2.1), is defined, i.e.

$$\Delta_t = \{t \in \mathbf{R}\colon\ 0 \le t \le a\},$$
$$\Delta_y = \{y = (y_1, y_2, \ldots, y_N)^{\mathrm{T}} \in \mathbf{R}^N\colon\ \underline{b}_i \le y_i \le \overline{b}_i,\quad i = 1, 2, \ldots, N\},$$

- F(T, Y) – an interval extension of $f(t, y)$, where an interval extension of the function

$$f\colon\ \mathbf{R} \times \mathbf{R}^N \supset \Delta_t \times \Delta_y \to \mathbf{R}^N$$

we call a function

$$F\colon\ \mathbf{IR}\times\mathbf{IR}^N\supset \mathbf{I}\Delta_t\times\mathbf{I}\Delta_y\to\mathbf{IR}^N$$

such that

$$(t,y)\in(T,Y)\Rightarrow f(t,y)\in F(T,Y),$$

and where **IR** and $\mathbf{IR}^N$ denote the space of real intervals, and the space of N-dimensional real interval vectors, respectively,

- $\Psi(T, Y)$ – an interval extension of $\psi(t, y)$ (see (3.10)),

and let us assume that:

- the function $F(T, Y)$ is defined and continuous for all $T\subset\Delta_t$ and $Y\subset\Delta_y$,
- 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 $Y\subset\Delta_y$ there exists a constant $\Lambda>0$ such that

$$w(F(T,Y))\le\Lambda(w(T)+w(Y)),\tag{3.31}$$

where $w(A)$ denotes the width of the interval A (if $A=(A_1,A_2,\ldots,A_N)^T$, then the number $w(A)$ is defined by $w(A)=\max\limits_{i=1,2,\ldots,N}w(A_i)$),

- the function $\Psi(T, Y)$ is defined for all $T\subset\Delta_t$ and $Y\subset\Delta_y$,
- the function $\Psi(T, Y)$ is monotonic with respect to inclusion.

For $t_0=0$ and $y_0\in Y(0)=Y_0$, where the interval Y_0 is given, the *explicit m-stage interval method of Runge-Kutta type* is defined as follows [87, 167]:

$$Y_{k+1}=Y_k+h\sum_{i=1}^{m}w_iK_{ik}+\big(\Psi(T_k,Y_k)+[-\alpha,\alpha]\big)h^{p+1},\qquad k=0,1,\ldots,n-1,\tag{3.32}$$

where $Y_k=Y(t_k)$ and Y_k depends also on n, $K_{ik}=K_{ik}(h)$,

$$\begin{aligned}K_{1k}&=F(T_k,Y_k),\\K_{ik}&=F\left(T_k+c_ih,\,Y_k+h\sum_{j=1}^{i-1}a_{ij}K_{jk}\right),\quad i=2,3,\ldots,m,\end{aligned}\tag{3.33}$$

α is a constant such that

$$\alpha=Mh_0,$$

where h_0 is a given initial step size, and (see (3.10))

$$\left|\frac{r_{k+1}^{(p+2)}(\theta h)}{(p+2)!}\right| \le M, \quad 0 < \theta < 1. \tag{3.34}$$

The step size h of the method (3.32), which fulfills the condition $0 < h \le h_0$, is given by

$$h = \frac{\xi_m^*}{n}, \tag{3.35}$$

where

$$\xi_m^* = \min\{\xi_0, \xi_2, \dots, \xi_m\}, \tag{3.36}$$

and where for $Y_0 \subset \varDelta_y$ and $y_0 \in Y_0$ the numbers $\xi_2 > 0$, $\xi_3 > 0$, ... , $\xi_m > 0$ are such that

$$Y_0 + \xi_i c_i F(\varDelta_t, \varDelta_y) \subset \varDelta_y, \quad i = 2, 3, \dots, m, \tag{3.37}$$

and the number $\xi_0 > 0$ fulfills the condition

$$Y_0 + \xi_0 \sum_{i=1}^{m} w_i F(\varDelta_t, \varDelta_y) + (\varPsi(\varDelta_t, \varDelta_y) + [-\alpha, \alpha]) h_0^p \subset \varDelta_y. \tag{3.38}$$

We divide the interval $[0, \xi_m^*]$ into n parts by the points $t_k = kh$ ($k = 0, 1, \dots, n$), whereas the intervals T_k, which appear in the methods (3.32) – (3.33), are selected in such a way that

$$t_k = kh \in T_k \subset [0, \xi_m^*].$$

On the basis of (3.32) – (3.33) we can present interval methods corresponding with the conventional explicit Runge-Kutta methods (3.11) and (3.13) – (3.15):

- the interval version of Euler's method (3.11)

$$\begin{gathered} Y_{k+1} = Y_k + hF(T_k, Y_k) + \big(\varPsi(T_k, Y_k) + [-\alpha, \alpha]\big)h^2, \\ k = 0, 1, \dots, n-1, \end{gathered} \tag{3.39}$$

  where $\varPsi(T, Y) = (\varPsi_1(T, Y), \varPsi_2(T, Y), \dots, \varPsi_N(T, Y))$ is an interval extension of $\psi(t, y) = (\psi_1(t, y), \psi_2(t, y), \dots, \psi_N(t, y))$ with $\psi_s(t, y)$ ($s = 1, 2, \dots, N$) given by (3.25),

- the interval version of Euler's improved method (3.13)

$$\begin{gathered} Y_{k+1} = Y_k + hK_{2k} + \big(\varPsi(T_k, Y_k) + [-\alpha, \alpha]\big)h^3, \\ K_{1k} = F(T_k, Y_k), \quad K_{2k} = F\left(T_k + \frac{h}{2}, Y_k + \frac{h}{2}K_{1k}\right), \\ k = 0, 1, \dots, n-1, \end{gathered} \tag{3.40}$$

where $\Psi(T, Y)$ is an interval extension of the function $\psi(t, y)$, of which components are given by (3.26),

- the interval version of the Euler-Cauchy method (3.14)

$$\begin{gathered} Y_{k+1} = Y_k + \frac{h}{2}(K_{1k} + K_{2k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^3, \\ K_{1k} = F(T_k, Y_k), \quad K_{2k} = F(T_k + h, Y_k + hK_{1k}), \\ k = 0, 1, \ldots, n-1, \end{gathered} \tag{3.41}$$

where $\Psi(T, Y)$ is an interval extension of (3.26),

- the interval version of the Runge-Kutta method (3.15)

$$\begin{gathered} Y_{k+1} = Y_k + \frac{h}{6}(K_{1k} + 2K_{2k} + 2K_{3k} + K_{4k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^5, \\ K_{1k} = F(T_k, Y_k), \quad K_{2k} = F\left(T_k + \frac{h}{2}, Y_k + \frac{h}{2}K_{1k}\right), \\ K_{3k} = F\left(T_k + \frac{h}{2}, Y_k + \frac{h}{2}K_{2k}\right), \quad K_{4k} = F(T_k + h, Y_k + hK_{3k}), \\ k = 0, 1, \ldots, n-1, \end{gathered} \tag{3.42}$$

where $\Psi(T, Y)$ is an interval extension of the function $\psi(t, y)$, of which components are given by (3.27).

From the classical theory of the Runga-Kutta methods it is known [38, 41, 50, 62, 77, 97] that for a given number m of stages and a given order p from the equations involving the coefficients w_i, c_i and a_{ij} we can obtain one- or multi-parameter families of solutions. If these equations are solved in floating-point interval arithmetic, then the number of possible families will be significantly greater. Below we explain this fact on the basis of the two-stage explicit method.

If we assume that $c_2 \neq 0$ is a parameter, then the one-parameter family of two--stage ($m = 2$) explicit Runge-Kutta method (with the maximum order $p = 2$) is given by (3.12). The same family we obtain if we take $w_1 \neq 1$ or $w_2 \neq 0$ as parameters. Now, let us assume that all calculations are carried out in floating-point interval arithmetic and consider the evaluation of $w_2 = \left[\underline{w}_2, \overline{w}_2\right]$. On the basis of (3.12) we have

$$\left[\underline{w}_2, \overline{w}_2\right] = \frac{1}{2\left[\underline{c}_2, \overline{c}_2\right]} = \frac{1}{\left[\nabla\left(2\underline{c}_2\right), \Delta\left(2\overline{c}_2\right)\right]} = \left[\nabla\left(\frac{1}{\Delta\left(2\overline{c}_2\right)}\right), \Delta\left(\frac{1}{\nabla\left(2\underline{c}_2\right)}\right)\right]. \tag{3.43}$$

If we take w_2 as a parameter, then on the basis of the same equations we get

$$c_2 = \frac{1}{2w_2} \underset{(3.14)}{=} \frac{1}{2\frac{1}{2c_2}} = c_2.$$

But in floating-point interval arithmetic we have

$$\left[\underline{c}_2', \overline{c}_2'\right] = \frac{1}{2\left[\underline{w}_2, \overline{w}_2\right]} = \frac{1}{\left[\nabla\left(2\underline{w}_2\right), \Delta\left(2\overline{w}_2\right)\right]} =$$

$$\underset{(3.43)}{=} \frac{1}{\left[\nabla\left(2\nabla\left(\frac{1}{\Delta\left(2\overline{c}_2\right)}\right)\right), \Delta\left(2\Delta\left(\frac{1}{\nabla\left(2\underline{c}_2\right)}\right)\right)\right]} =$$

$$= \left[\nabla\left(\frac{1}{\Delta\left(2\Delta\left(\frac{1}{\nabla\left(2\underline{c}_2\right)}\right)\right)}\right), \Delta\left(\frac{1}{\nabla\left(2\nabla\left(\frac{1}{\Delta\left(2\overline{c}_2\right)}\right)\right)}\right)\right],$$

and we see that in general $\underline{c}_2' \neq \underline{c}_2$ and $\overline{c}_2' \neq \overline{c}_2$.

In Table 3.1 we present the numbers of one-parameter families in floating-point interval arithmetic.

Table 3.1. The numbers of explicit Runge-Kutta one-parameter families

m	*Conventional Runge-Kutta methods*	*Runge-Kutta methods in floating-point interval arithmetic*
2	1	3
3	2	7
4	4	21

3.2.2. The Exact Solution vs. Interval Solutions

For the explicit m-stage interval method of Runge-Kutta type we can prove

Theorem 3.1. *For the exact solution* $y(t)$ *of the initial value problem* (2.1) *we have* $y(t_k) \in Y_k$ $(k = 0, 1, \ldots, n)$*, where* Y_k *are obtained from the method* (3.32) – (3.33).

Proof (the mathematical induction with respect to k). The assumption of the method (3.32) – (3.33) states that $y_0 = y(0) = y(t_0) \in Y_0$, i.e. the case $k = 0$ is true. Let us assume that $y(t_k) \in Y_k$. According to (3.5) and (3.10) we have

$$y(t_{k+1}) = y(t_k) + h\sum_{i=1}^{m} w_i\kappa_i(h) + R_m(h),$$

where

$$R_m(h) = \left[\ \psi(t_k, y(t_k)) + \frac{r_{k+1}^{(p+2)}(\theta h)h}{(p+1)!}\right]h^{p+1} \tag{3.44}$$

is a summarized error of interpolation and integration. But $\psi(t_k, y(t_k)) \in \Psi(T_k, Y_k)$, and from the assumption (3.34) about the method considered it follows that

$$\left|\frac{r_{k+1}^{(p+2)}(\theta h)h}{(p+2)!}\right| \le Mh \le Mh_0 = \alpha.$$

This implies that

$$\frac{r_{k+1}^{(p+2)}(\theta h)h}{(p+2)!} \in [-\alpha, \alpha].$$

Hence, taking into account (3.44), we have

$$R_m(h) \in \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^{p+1}.$$

Moreover, $f(t, y) \in F(T, Y)$ for each $t \in \Delta_t$ and $y \in \Delta_y$, and from the induction assumption we have $y(t_k) \in Y_k$. Thus, we get

$$y(t_{k+1}) \in Y_k + h\sum_{i=1}^{m} w_i K_{ik} + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^{p+1}.$$

But on the basis of (3.34) the interval on the right-hand side of membership operation is equal to Y_{k+1}. ■

3.2.3. Widths of Interval Solutions

Before we estimate the widths of interval solutions obtained by the method (3.32), let us consider the widths of intervals K_{ik} given by (3.33). From this formula and the property (3.31) of the function F it follows that

$$w(K_{ik}) \le \Lambda\left[w(T_k) + w(Y_k) + h\sum_{j=1}^{i-1} \left|a_{ij}\right| w(K_{jk}) \right].$$

Hence,

$$\begin{aligned}
w(K_{1k}) &\le \Lambda(w(T_k) + w(Y_k)),\\
w(K_{2k}) &\le \Lambda(w(T_k) + w(Y_k) + h\left|a_{21}\right| w(K_{1k})) \le\\
&\le \Lambda(w(T_k) + w(Y_k))(1 + \left|a_{21}\right| h\Lambda),\\
w(K_{3k}) &\le \Lambda(w(T_k) + w(Y_k) + h\left|a_{31}\right| w(K_{1k}) + h\left|a_{32}\right| w(K_{2k})) \le\\
&\le \Lambda(w(T_k) + w(Y_k))[1 + h\left|a_{31}\right|\Lambda + h\left|a_{32}\right|(1 + h\left|a_{21}\right|\Lambda)\Lambda] =\\
&\le \Lambda(w(T_k) + w(Y_k))[1 + (\left|a_{31}\right| + \left|a_{31}\right|)h\Lambda + \left|a_{32}\right|\left|a_{21}\right|(h\Lambda)^2],
\end{aligned}$$

. . . .

Thus, for each $i = 1, 2, \ldots, m$ we have

$$w(K_{ik}) \le \Lambda(w(T_k) + w(Y_k))\sum_{j=0}^{i-1} \mu_{ij}(h\Lambda)^j, \tag{3.45}$$

where μ_{ij} denote some constants.

Now, we can prove

Theorem 3.2. *If* Y_k ($k = 1, 2, \ldots, n$) *are obtained from* (3.32) – (3.33)*, then*

$$w(Y_k) \le Qh^p + Rw(Y_0) + S \max_{l = 0, 1, \ldots, k-1} w(T_l), \tag{3.46}$$

where Q, R and S denote some nonnegative constants.

Proof. From (3.32) we get

$$w(Y_{k+1}) \le w(Y_k) + h\sum_{i=1}^{m} \left| w_i \right| w(K_{ik}) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}. \tag{3.47}$$

The insertion of (3.45) into (3.47) yields

$$w(Y_{k+1}) \le w(Y_k) + h\Lambda(w(T_k) + w(Y_k))\sum_{i=1}^{m} |w_i| \sum_{j=1}^{i-1} \mu_{ij}(h\Lambda)^j +$$

$$+ [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1} \le$$

$$\le w(Y_k)\left[1 + h\Lambda\sum_{i=1}^{m} |w_i| \sum_{j=0}^{i-1} \mu_{ij}(h\Lambda)^j\right] +$$

$$+ h\Lambda w(T_k)\left[1 + h\Lambda\sum_{i=1}^{m} |w_i| \sum_{j=0}^{i-1} \mu_{ij}(h\Lambda)^j\right] + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}.$$

Since $h < h_0$, then

$$\sum_{i=1}^{m} |w_i| \sum_{j=0}^{i-1} \mu_{ij}(h\Lambda)^j \le \sum_{i=1}^{m} |w_i| \sum_{j=0}^{i-1} \mu_{ij}(h_0\Lambda)^j = \gamma_m.$$

Thus,

$$\begin{gathered} w(Y_{k+1}) \le w(Y_k)(1 + \gamma_m h\Lambda) + \gamma_m h\Lambda w(T_k) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}, \\ k = 0, 1, \ldots, n-1. \end{gathered} \tag{3.48}$$

From (3.48) we get

$$w(Y_1) \le w(Y_0)(1 + \gamma_m h\Lambda) + \gamma_m h\Lambda w(T_0) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1},$$

$$w(Y_2) \le w(Y_1)(1 + \gamma_m h\Lambda) + \gamma_m h\Lambda w(T_1) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1} \le$$

$$\le w(Y_0)(1 + \gamma_m h\Lambda)^2 +$$

$$+ \left\{\gamma_m h\Lambda \max_{l=0,1} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}\right\}[1 + (1 + \gamma_m h\Lambda)],$$

$$w(Y_3) \le w(Y_2)(1 + \gamma_m h\Lambda) + \gamma_m h\Lambda w(T_2) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1} \le$$

$$\le w(Y_0)(1 + \gamma_m h\Lambda)^3 +$$

$$+ \left\{\gamma_m h\Lambda \max_{l=0,1,2} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}\right\} \times$$

$$\times [1 + (1 + \gamma_m h\Lambda) + (1 + \gamma_m h\Lambda)^2],$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

i.e. for each $k = 1, 2, \ldots, n$ we have

$$w(Y_k) \le w(Y_0)(1 + \gamma_m h\Lambda)^k +$$
$$+ \left\{ \gamma_m h\Lambda \max_{l = 0, 1, \ldots, k-1} w(T_i) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha] h^{p+1} \right\} \times$$
$$\times \sum_{i=0}^{k-1} (1 + \gamma_m h\Lambda)^i =$$
$$= w(Y_0)(1 + \gamma_m h\Lambda)^k + \gamma_m h\Lambda \max_{l = 0, 1, \ldots, k-1} w(T_i) \sum_{i=0}^{k-1} (1 + \gamma_m h\Lambda)^i +$$
$$+ [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha] h^{p+1} \sum_{i=0}^{k-1} (1 + \gamma_m h\Lambda)^i .$$

Since

$$\sum_{i=0}^{k-1} (1 + \gamma_m h\Lambda)^i = \frac{(1 + \gamma_m h\Lambda)^k - 1}{\gamma_m h\Lambda} \le \frac{\exp(\gamma_m k h\Lambda) - 1}{\gamma_m h\Lambda} \le$$
$$\le \frac{\exp(\gamma_m n h\Lambda) - 1}{\gamma_m h\Lambda} \le \frac{\exp(\gamma_m \xi_m^* \Lambda) - 1}{\gamma_m h\Lambda},$$

then

$$w(Y_k) \le w(Y_0) \exp(\gamma_m \xi_m^* \Lambda) + [\exp(\gamma_m \xi_m^* \Lambda) - 1] \max_{l = 0, 1, \ldots, k-1} w(T_l) +$$
$$+ [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha] \frac{\exp(\gamma_m \xi_m^* \Lambda)}{\gamma_m \Lambda} h^p \le$$
$$\le R w(Y_0) + S \max_{l = 0, 1, \ldots, k-1} w(T_l) + Q h^p,$$

where

$$Q = [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha] \frac{\exp(\gamma_m \xi_m^* \Lambda) - 1}{\gamma_m \Lambda}, \quad R = \exp(\gamma_m \xi_m^* \Lambda), \quad S = R - 1. \qquad \blacksquare$$

3.3. Implicit Methods

3.3.1. Basic Formulas

Let $F(T, Y)$ and $\Psi(T, Y)$ be interval extensions of $f(t, y)$ and $\psi(t, y)$, respectively, and fulfill the same assumptions as in Section 3.2. For $t_0 = 0$ and $y_0 \in Y_0$, where the interval Y_0 is given, an *implicit m-stage interval method of Runge-Kutta type*, which solves the initial value problem (2.1), is given by [54 – 57, 121, 122, 125]

$$Y_{k+1} = Y_k + h\sum_{i=1}^{m} w_i K_{ik} + \left(\Psi(T_k, Y_k) + [-\alpha, \alpha]\right)h^{p+1}, \qquad k = 0, 1, \ldots, n-1, \tag{3.49}$$

where

$$K_{ik} = F\left(T_k + c_i h,\ Y_k + h\sum_{j=1}^{m} a_{ij}K_{jk}\right), \qquad \alpha = Mh_0, \quad 0 < h \le h_0, \tag{3.50}$$

and where h_0 denotes a given number (initial value of step size).

To find h we apply the following formulas:

$$h = \frac{\eta_m^*}{n}, \quad \eta_m^* = \min\{\eta_0, \eta_1, \ldots, \eta_m\}, \tag{3.51}$$

where the numbers $\eta_1 > 0,\ \eta_2 > 0,\ \ldots,\ \eta_m > 0$ should be evaluated in such a way that

$$Y_0 + \eta_i c_i F(\Delta_t, \Delta_y) \subset \Delta_y, \quad i = 1, 2, \ldots, m, \tag{3.52}$$

and the number $\eta_0 > 0$ should fulfill the following condition:

$$Y_0 + \eta_0 \sum_{i=1}^{m} w_i F(\Delta_t, \Delta_y) + \left(\Psi(\Delta_t, \Delta_y) + [-\alpha, \alpha]\right)h_0^p \subset \Delta_y, \tag{3.53}$$

and where $Y_0 \subset \Delta_y$ and $y_0 \in Y_0$.

The interval $[0, \eta_m^*]$ is then divided into n parts by the points $t_k = kh$, where $k = 0, 1, \ldots, n$, and the intervals T_k occurring in the method should be chosen in such a way that

$$t_k = kh \in T_k \subset [0, \eta_m^*].$$

From (3.50) it follows that in each step k we have to solve a nonlinear equation of the form

$$X = G(T, X),$$

where

$$T \in \mathbf{I}\Delta_t \subset \mathbf{IR}, \quad X = (X_1, X_2, \dots, X_N)^{\mathrm{T}} \in \mathbf{I}\Delta_y \subset \mathbf{IR}^N,$$
$$G\colon\ \mathbf{I}\Delta_t \times \mathbf{I}\Delta_y \to \mathbf{IR}^N.$$

If we assume that G is a contraction mapping, then the well-known fixed-point theorem implies that the iteration

$$X^{(l+1)} = G(T, X^{(l)}), \quad l = 0, 1, \dots, \tag{3.54}$$

is convergent to X^*, i.e. $\lim\limits_{l \to \infty} X^{(l)} = X^*$, for an arbitrary choice of $X^{(0)} \in \mathbf{I}\Delta_y$. Let us recall that G is called a contraction mapping if

$$d(G(T, X_{(1)}), G(T, X_{(2)})) \le \alpha\, d(X_{(1)}, X_{(2)}),$$

where d is a metric[1] and $\alpha < 1$ denotes a constant.

For the equation (3.50) the process (3.54) is of the form

$$K_{ik}^{(l+1)} = F\left(T_k + c_i h,\ Y_k + h\sum_{j=1}^{m} a_{ij} K_{jk}^{(l)}\right),$$
$$i = 1, 2, \dots, m, \quad k = 0, 1, \dots, n-1, \quad l = 0, 1, \dots, \tag{3.55}$$

where

$$K_{ik}^{(0)} = F(T_k + c_i h, Y_k).$$

The process (3.55) may be modified to the following form:

[1] In the space **IR** the metric is defined by (1.3). In the space $\mathbf{IR}^N$ the metric can be defined by the formula

$$d(A, B) = \max_{i = 1, 2, \dots, N} d(A_i, B_i),$$

where $A = (A_1, A_2, \dots, A_N)^{\mathrm{T}} \in \mathbf{IR}^N$, $B = (B_1, B_2, \dots, B_N)^{\mathrm{T}} \in \mathbf{IR}^N$, $A_i = \left[\underline{a}_i, \overline{a}_i\right]$ and $B_i = \left[\underline{b}_i, \overline{b}_i\right]$.

$$K_{ik}^{(l+1)} = F\left[T_k + c_i h, Y_k + h\left(\sum_{j=1}^{i-1} a_{ij} K_{jk}^{(l+1)} + \sum_{j=i}^{m} a_{ij} K_{jk}^{(l)}\right)\right], \tag{3.56}$$

which should reduce the number of iterations.

Interval methods corresponding with the conventional implicit Runge-Kutta methods (3.16), (3.17), (3.19), (3.20), (3.22) and (3.23) are as follows [55, 175]:

- the interval version of the implicit midpoint rule (3.16)

$$\begin{aligned} Y_{k+1} &= Y_k + hK_{1k} + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^3, \\ K_{1k} &= F\left(T_k + \frac{h}{2}, Y_k + \frac{h}{2}K_{1k}\right), \\ k &= 0, 1, \ldots, n-1, \end{aligned} \tag{3.57}$$

  where $\Psi(T, Y) = (\Psi_1(T, Y), \Psi_2(T, Y), \ldots, \Psi_N(T, Y))$ is an interval extension of $\psi(t, y) = (\psi_1(t, y), \psi_2(t, y), \ldots, \psi_N(t, y))$ with $\psi_s(t, y)$ $(s = 1, 2, \ldots, N)$ given by (3.28),

- the interval version of the Hammer-Hollingsworth method (3.17)

$$\begin{aligned} Y_{k+1} &= Y_k + \frac{h}{2}(K_{1k} + K_{2k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^5, \\ K_{1k} &= F\left[T_k + \left(\frac{1}{2} \mp \frac{\sqrt{3}}{6}\right)h, Y_k + \frac{h}{4}K_{1k} + \left(\frac{1}{4} \mp \frac{\sqrt{3}}{6}\right)hK_{2k}\right], \\ K_{2k} &= F\left[T_k + \left(\frac{1}{2} \pm \frac{\sqrt{3}}{6}\right)h, Y_k + \left(\frac{1}{4} \pm \frac{\sqrt{3}}{6}\right)hK_{1k} + \frac{h}{4}K_{2k}\right], \\ k &= 0, 1, \ldots, n-1, \end{aligned} \tag{3.58}$$

  where $\Psi(T, Y)$ is an interval extension of $\psi(t, y)$, whose components are given by (3.30) with $m = 2$,

- the interval version of the semi-implicit method (3.19)

$$\begin{aligned} Y_{k+1} &= Y_k + \frac{h}{4}(K_{1k} + 3K_{2k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^4, \\ K_{1k} &= F(T_k + h, Y_k + hK_{1k}), \\ K_{2k} &= F\left(T_k + \frac{h}{3}, Y_k - \frac{h}{3}K_{1k} + \frac{2h}{3}K_{2k}\right), \\ k &= 0, 1, \ldots, n-1, \end{aligned} \tag{3.59}$$

where $\Psi(T, Y)$ is an interval extension of $\psi(t, y)$, whose components are given by (3.29) with $m = 2$,

- the interval versions of two diagonally implicit methods given by (3.20)

$$\begin{aligned}
Y_{k+1} &= Y_k + \frac{h}{2}(K_{1k} + K_{2k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^4,\\
K_{1k} &= F\left[T_k + \left(\frac{1}{2} \pm \frac{\sqrt{3}}{6}\right)h,\ Y_k + \left(\frac{1}{2} \pm \frac{\sqrt{3}}{6}\right)hK_{1k}\right],\\
K_{2k} &= F\left[T_k + \left(\frac{1}{2} \mp \frac{\sqrt{3}}{6}\right)h,\ Y_k \mp \frac{\sqrt{3}}{3}hK_{1k} + \left(\frac{1}{2} \pm \frac{\sqrt{3}}{6}\right)hK_{2k}\right],\\
k &= 0, 1, \ldots, n-1,
\end{aligned} \tag{3.60}$$

where $\Psi(T, Y)$ is the same function as in (3.59),

- the interval version of the Butcher semi-implicit method (3.22)

$$\begin{aligned}
Y_{k+1} &= Y_k + \frac{h}{6}(K_{1k} + 4K_{2k} + K_{3k}) + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^5,\\
K_{1k} &= F(T_k, Y_k),\\
K_{2k} &= F\left(T_k + \frac{h}{2},\ Y_k + \frac{h}{4}(K_{1k} + K_{2k})\right),\\
K_{3k} &= F(T_k, Y_k + hK_{2k}),\\
k &= 0, 1, \ldots, n-1,
\end{aligned} \tag{3.61}$$

where $\Psi(T, Y)$ is an interval extension of $\psi(t, y)$, whose components are given by (3.30) with $m = 3$,

- the interval versions of Alexander's diagonally implicit methods of the form (3.23)

$$\begin{aligned}
Y_{k+1} &= Y_k + h\left[\frac{1}{8\zeta^2}K_{1k} + \left(1 - \frac{1}{4\zeta^2}\right)K_{2k} + \frac{1}{8\zeta^2}K_{3k}\right] +\\
&\quad + \big(\Psi(T_k, Y_k) + [-\alpha, \alpha]\big)h^5,\\
K_{1k} &= F\left[T_k + \left(\frac{1}{2} + \frac{\sqrt{3}}{3}\zeta\right)h,\ Y_k + \left(\frac{1}{2} + \frac{\sqrt{3}}{3}\zeta\right)hK_{1k}\right],\\
K_{2k} &= F\left[T_k + \frac{h}{2},\ Y_k - \frac{\sqrt{3}}{3}\zeta hK_{1k} + \left(\frac{1}{2} + \frac{\sqrt{3}}{3}\zeta\right)hK_{2k}\right],
\end{aligned} \tag{3.62}$$

$$K_{3k} = F\left[T_k + \left(\frac{1}{2} - \frac{\sqrt{3}}{3}\zeta\right)h,\right.$$

$$\left. Y_k + \left(\frac{1}{2} + \frac{2\sqrt{3}}{3}\zeta\right)hK_{1k} - \left(\frac{1}{2} + \frac{4\sqrt{3}}{3}\zeta\right)hK_{2k} + \left(\frac{1}{2} + \frac{\sqrt{3}}{3}\zeta\right)hK_{3k}\right],$$

where $k = 0, 1, \ldots, n - 1$, $\zeta = \cos 10°, -\cos 50°$ or $-\cos 70°$, and where $\Psi(T, Y)$ is the same function as in (3.61).

3.3.2. On an Inclusion of the Exact Solution within Interval Solutions

For the methods (3.49) – (3.50) we can prove that the exact solution of the initial value problem (2.1) belongs to the intervals obtained by these methods.

Theorem 3.3. *For the exact solution $y(t)$ of the initial value problem* (2.1) *we have $y(t_k) \in Y_k$ ($k = 0, 1, \ldots, n$), where Y_k are obtained from the method* (3.49) – (3.50).

The proof of this theorem is exactly the same as the proof of Theorem 3.1.

3.3.3. Estimations of the Widths of Interval Solutions

Before we estimate the widths of interval solutions obtained by the methods of the form (3.49), let us consider the widths of intervals K_{ik} given by (3.50). From this formula and properties of the function F it follows that [54, 122]

$$w(K_{ik}) \le \Lambda[w(T_k) + w(Y_k)] + h\Lambda\sum_{j=1}^{m}|a_{ij}|w(K_{jk}), \tag{3.63}$$

where $i = 1$ for $m = 1$, $i = 1, 2$ for $m = 2$, etc. The inequalities (3.63) are of the form

$$x_i \le \beta + \sum_{j=1}^{m} \alpha_{ij}x_j, \quad i = 1, 2, \ldots, m,$$

where α_{ij} and β are some constants. These inequalities can be written as

$$(1 - \alpha_{ii})x_i - \sum_{\substack{j=1\\ j \ne i}}^{m} \alpha_{ij}x_j \le \beta, \quad i = 1, 2, \ldots, m. \tag{3.64}$$

For $m = 1, 2, 3$ and 4 we get the following solutions of (3.64):

- $m = 1$

$$x_1 \le \frac{\beta}{1-\alpha_{11}}, \tag{3.65}$$

where

$$1-\alpha_{11} > 0,$$

- $m = 2$

$$\begin{aligned} x_1 &\le \frac{(1+\alpha_{12}-\alpha_{22})\beta}{(1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21}}, \\ x_2 &\le \frac{(1+\alpha_{21}-\alpha_{11})\beta}{(1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21}}, \end{aligned} \tag{3.66}$$

where

$$1-\alpha_{11} > 0, \quad 1-\alpha_{22} > 0, \quad (1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21} > 0,$$

- $m = 3$

$$\begin{aligned} x_1 \le \frac{\beta}{W_3}&[(1-\alpha_{22})(1-\alpha_{33})+(1+\alpha_{23}-\alpha_{33})\alpha_{12}+ \\ &+(1+\alpha_{32}-\alpha_{22})\alpha_{13}-\alpha_{23}\alpha_{32}], \\ x_2 \le \frac{\beta}{W_3}&[(1-\alpha_{11})(1-\alpha_{33})+(1+\alpha_{13}-\alpha_{33})\alpha_{21}+ \\ &+(1+\alpha_{31}-\alpha_{11})\alpha_{23}-\alpha_{13}\alpha_{31}], \\ x_1 \le \frac{\beta}{W_3}&[(1-\alpha_{11})(1-\alpha_{22})+(1+\alpha_{12}-\alpha_{22})\alpha_{31}+ \\ &+(1+\alpha_{21}-\alpha_{11})\alpha_{32}-\alpha_{12}\alpha_{21}], \end{aligned} \tag{3.67}$$

where

$$\begin{aligned} W_3 = {} & (1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{33})-\alpha_{12}\alpha_{23}\alpha_{31}-\alpha_{13}\alpha_{21}\alpha_{32}- \\ & -(1-\alpha_{11})\alpha_{23}\alpha_{32}-(1-\alpha_{22})\alpha_{13}\alpha_{31}-(1-\alpha_{33})\alpha_{12}\alpha_{21}, \end{aligned}$$

and where

$$\begin{gathered} 1-\alpha_{11} > 0, \quad 1-\alpha_{22} > 0, \quad 1-\alpha_{33} > 0, \\ (1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21} > 0, \quad (1-\alpha_{11})(1-\alpha_{33})-\alpha_{13}\alpha_{31} > 0, \\ (1-\alpha_{22})(1-\alpha_{33})-\alpha_{23}\alpha_{32} > 0, \quad W_3 > 0, \end{gathered}$$

- $m = 4$

$$
\begin{aligned}
x_1 \le \frac{\beta}{W_4}\{&(1-\alpha_{22})(1-\alpha_{33})(1-\alpha_{44})+\\
&+[1+(1+\alpha_{34}-\alpha_{44})\alpha_{23}+(1+\alpha_{43}-\alpha_{33})\alpha_{24}-\\
&\quad-(1-\alpha_{44})\alpha_{33}-\alpha_{34}\alpha_{43}-\alpha_{44}]\alpha_{12}+\\
&+[1+(1+\alpha_{24}-\alpha_{44})\alpha_{32}+(1+\alpha_{42}-\alpha_{22})\alpha_{34}-\\
&\quad-(1-\alpha_{44})\alpha_{22}-\alpha_{24}\alpha_{42}-\alpha_{44}]\alpha_{13}+\\
&+[1+(1+\alpha_{23}-\alpha_{33})\alpha_{42}+(1+\alpha_{32}-\alpha_{22})\alpha_{43}-\\
&\quad-(1-\alpha_{33})\alpha_{22}-\alpha_{23}\alpha_{32}-\alpha_{33}]\alpha_{14}-\\
&-(1-\alpha_{44})\alpha_{23}\alpha_{32}-(1-\alpha_{33})\alpha_{24}\alpha_{42}-(1-\alpha_{22})\alpha_{34}\alpha_{43}-\\
&-\alpha_{23}\alpha_{34}\alpha_{42}-\alpha_{24}\alpha_{32}\alpha_{43}\},\\
x_2 \le \frac{\beta}{W_4}\{&(1-\alpha_{11})(1-\alpha_{33})(1-\alpha_{44})+\\
&+[1+(1+\alpha_{34}-\alpha_{44})\alpha_{13}+(1+\alpha_{43}-\alpha_{33})\alpha_{14}-\\
&\quad-(1-\alpha_{44})\alpha_{33}-\alpha_{34}\alpha_{43}-\alpha_{44}]\alpha_{21}+\\
&+[1+(1+\alpha_{14}-\alpha_{44})\alpha_{31}+(1+\alpha_{41}-\alpha_{11})\alpha_{34}-\\
&\quad-(1-\alpha_{44})\alpha_{11}-\alpha_{14}\alpha_{41}-\alpha_{44}]\alpha_{23}+\\
&+[1+(1+\alpha_{13}-\alpha_{33})\alpha_{41}+(1+\alpha_{31}-\alpha_{11})\alpha_{43}-\\
&\quad-(1-\alpha_{33})\alpha_{11}-\alpha_{13}\alpha_{31}-\alpha_{33}]\alpha_{24}-\\
&-(1-\alpha_{44})\alpha_{13}\alpha_{31}-(1-\alpha_{33})\alpha_{14}\alpha_{41}-(1-\alpha_{11})\alpha_{34}\alpha_{43}-\\
&-\alpha_{13}\alpha_{34}\alpha_{41}-\alpha_{14}\alpha_{31}\alpha_{43}\},\\
x_3 \le \frac{\beta}{W_4}\{&(1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{44})+\\
&+[1+(1+\alpha_{24}-\alpha_{44})\alpha_{12}+(1+\alpha_{42}-\alpha_{22})\alpha_{14}-\\
&\quad-(1-\alpha_{44})\alpha_{22}-\alpha_{24}\alpha_{42}-\alpha_{44}]\alpha_{31}+\\
&+[1+(1+\alpha_{14}-\alpha_{44})\alpha_{21}+(1+\alpha_{41}-\alpha_{11})\alpha_{24}-\\
&\quad-(1-\alpha_{44})\alpha_{11}-\alpha_{14}\alpha_{41}-\alpha_{44}]\alpha_{32}+\\
&+[1+(1+\alpha_{12}-\alpha_{22})\alpha_{41}+(1+\alpha_{21}-\alpha_{11})\alpha_{42}-\\
&\quad-(1-\alpha_{11})\alpha_{22}-\alpha_{12}\alpha_{21}-\alpha_{11}]\alpha_{34}-\\
&-(1-\alpha_{44})\alpha_{12}\alpha_{21}-(1-\alpha_{11})\alpha_{24}\alpha_{42}-(1-\alpha_{22})\alpha_{14}\alpha_{41}-\\
&-\alpha_{12}\alpha_{24}\alpha_{41}-\alpha_{14}\alpha_{21}\alpha_{42}\},\\
x_4 \le \frac{\beta}{W_4}\{&(1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{33})+
\end{aligned}
\tag{3.68}
$$

$$
\begin{aligned}
&+[1+(1+\alpha_{23}-\alpha_{33})\alpha_{12}+(1+\alpha_{32}-\alpha_{22})\alpha_{12}-\\
&\quad -(1-\alpha_{33})\alpha_{22}-\alpha_{23}\alpha_{32}-\alpha_{33}]\alpha_{41}+\\
&+[1+(1+\alpha_{13}-\alpha_{33})\alpha_{21}+(1+\alpha_{31}-\alpha_{11})\alpha_{23}-\\
&\quad -(1-\alpha_{33})\alpha_{11}-\alpha_{13}\alpha_{31}-\alpha_{33}]\alpha_{42}+\\
&+[1+(1+\alpha_{12}-\alpha_{22})\alpha_{31}+(1+\alpha_{21}-\alpha_{11})\alpha_{32}-\\
&\quad -(1-\alpha_{11})\alpha_{22}-\alpha_{12}\alpha_{21}-\alpha_{11}]\alpha_{43}-\\
&-(1-\alpha_{11})\alpha_{23}\alpha_{32}-(1-\alpha_{22})\alpha_{13}\alpha_{31}-(1-\alpha_{33})\alpha_{12}\alpha_{21}-\\
&-\alpha_{12}\alpha_{23}\alpha_{31}-\alpha_{13}\alpha_{21}\alpha_{32}\},
\end{aligned}
$$

where

$$
\begin{aligned}
W_4 = {}&(1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{33})(1-\alpha_{44})-\\
&-(1-\alpha_{44})[(1-\alpha_{11})\alpha_{23}\alpha_{32}+(1-\alpha_{22})\alpha_{13}\alpha_{31}+(1-\alpha_{33})\alpha_{12}\alpha_{21}]-\\
&-(1-\alpha_{11})(\alpha_{23}\alpha_{34}\alpha_{42}+\alpha_{24}\alpha_{32}\alpha_{43})-\\
&-(1-\alpha_{22})(\alpha_{13}\alpha_{34}\alpha_{41}+\alpha_{14}\alpha_{31}\alpha_{43})-\\
&-(1-\alpha_{33})(\alpha_{12}\alpha_{24}\alpha_{41}+\alpha_{14}\alpha_{21}\alpha_{42})-\\
&-(1-\alpha_{44})(\alpha_{12}\alpha_{23}\alpha_{31}+\alpha_{13}\alpha_{21}\alpha_{32})-\\
&-\alpha_{14}\alpha_{41}[(1-\alpha_{22})(1-\alpha_{33})-\alpha_{23}\alpha_{32}]-\\
&-\alpha_{24}\alpha_{42}[(1-\alpha_{11})(1-\alpha_{33})-\alpha_{13}\alpha_{31}]-\\
&-\alpha_{34}\alpha_{43}[(1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21}]-\\
&-\alpha_{12}(\alpha_{23}\alpha_{34}\alpha_{41}+\alpha_{24}\alpha_{31}\alpha_{43})-\alpha_{13}(\alpha_{21}\alpha_{34}\alpha_{42}+\alpha_{24}\alpha_{32}\alpha_{41})-\\
&-\alpha_{14}(\alpha_{21}\alpha_{32}\alpha_{43}+\alpha_{23}\alpha_{31}\alpha_{42}),
\end{aligned}
$$

and where

$$
\begin{gathered}
1-\alpha_{11}>0,\quad 1-\alpha_{22}>0,\quad 1-\alpha_{33}>0,\quad 1-\alpha_{44}>0,\\
(1-\alpha_{11})(1-\alpha_{22})-\alpha_{12}\alpha_{21}>0,\quad (1-\alpha_{11})(1-\alpha_{33})-\alpha_{13}\alpha_{31}>0,\\
(1-\alpha_{11})(1-\alpha_{44})-\alpha_{14}\alpha_{41}>0,\quad (1-\alpha_{22})(1-\alpha_{33})-\alpha_{23}\alpha_{32}>0,\\
(1-\alpha_{22})(1-\alpha_{44})-\alpha_{14}\alpha_{41}>0,\quad (1-\alpha_{33})(1-\alpha_{44})-\alpha_{34}\alpha_{43}>0,\\
(1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{33})-(1-\alpha_{11})\alpha_{23}\alpha_{32}-(1-\alpha_{22})\alpha_{13}\alpha_{31}-\\
-(1-\alpha_{33})\alpha_{12}\alpha_{21}-\alpha_{12}\alpha_{23}\alpha_{31}-\alpha_{13}\alpha_{21}\alpha_{32}>0,\\
(1-\alpha_{11})(1-\alpha_{22})(1-\alpha_{44})-(1-\alpha_{11})\alpha_{24}\alpha_{42}-(1-\alpha_{22})\alpha_{14}\alpha_{41}-\\
-(1-\alpha_{44})\alpha_{12}\alpha_{21}-\alpha_{12}\alpha_{24}\alpha_{41}-\alpha_{14}\alpha_{21}\alpha_{42}>0,\\
(1-\alpha_{11})(1-\alpha_{33})(1-\alpha_{44})-(1-\alpha_{11})\alpha_{34}\alpha_{43}-(1-\alpha_{33})\alpha_{14}\alpha_{41}-\\
-(1-\alpha_{44})\alpha_{13}\alpha_{31}-\alpha_{13}\alpha_{34}\alpha_{41}-\alpha_{14}\alpha_{31}\alpha_{43}>0,
\end{gathered}
$$

$$(1-\alpha_{22})(1-\alpha_{33})(1-\alpha_{44})-(1-\alpha_{22})\alpha_{34}\alpha_{43}-(1-\alpha_{33})\alpha_{24}\alpha_{42}-$$
$$-(1-\alpha_{44})\alpha_{23}\alpha_{32}-\alpha_{23}\alpha_{34}\alpha_{42}-\alpha_{24}\alpha_{32}\alpha_{43}>0,$$
$$W_4>0.$$

Using the above results we can estimate the widths of interval solutions for implicit one-, two-, three- and four-stage methods of Runge-Kutta type [54, 55, 122, 175].

Theorem 3.4. *If* Y_k ($k = 1, 2, ..., n$) *are obtained from* (3.49) – (3.50) *with* $m = 1$, *then for* $h_0 < 2/\Lambda$ *we have*

$$w(Y_k)\le Qh^2+Rw(Y_0)+S\max_{l=1,2,\ldots,n}w(T_l), \tag{3.69}$$

where Q, R and S denote some nonnegative constants.

Proof. From (3.49) for $m = 1$ we have

$$w(Y_{k+1})\le w(Y_k)+hw(K_{1k})+[w(\Psi(\Delta_t,\Delta_y))+2\alpha]h^3. \tag{3.70}$$

Applying (3.65) to (3.63) we get

$$w(K_{1k})\le\frac{\Lambda[w(T_k)+w(Y_k)]}{1-\dfrac{h\Lambda}{2}}, \tag{3.71}$$

if

$$1-\frac{h\Lambda}{2}>0. \tag{3.72}$$

Since $h\le h_0$, then from the assumption that $h_0<2/\Lambda$ it follows the inequality (3.72), and also (3.71). For $h\le h_0$ from (3.71) we have

$$w(K_{1k})\le\frac{2\Lambda}{2-h_0\Lambda}[w(T_k)+w(Y_k)].$$

The insertion of this estimate into (3.70) yields

$$w(Y_{k+1})\le w(Y_k)+\frac{2h\Lambda}{2-h_0\Lambda}[w(T_k)+w(Y_k)]+[w(\Psi(\Delta_t,\Delta_y))+2\alpha]h^3.$$

Denoting

$$\nu_1=\frac{2\Lambda}{2-h_0\Lambda},$$

we can write the last inequality in the form

$$w(Y_{k+1}) \le w(Y_k)(1+\nu_1 h\Lambda) + \nu_1 h\Lambda w(T_k) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3, \qquad k = 0, 1, \ldots, n-1. \tag{3.73}$$

From (3.73) it follows that

$$\begin{aligned}
&w(Y_1) \le w(Y_0)(1+\nu_1 h\Lambda) + \nu_1 h\Lambda w(T_0) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3,\\
&w(Y_2) \le w(Y_1)(1+\nu_1 h\Lambda) + \nu_1 h\Lambda w(T_1) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3 \le\\
&\quad \le \left(w(Y_0)(1+\nu_1 h\Lambda) + \nu_1 h\Lambda w(T_0) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3\right)(1+\nu_1 h\Lambda) +\\
&\quad + \nu_1 h\Lambda w(T_1)[w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3 \le\\
&\quad \le w(Y_0)(1+\nu_1 h\Lambda)^2 +\\
&\quad + \left(\nu_1 h\Lambda \max_{l=0,1} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3\right)[1 + (1+\nu_1 h\Lambda)],\\
&w(Y_2) \le w(Y_2)(1+\nu_1 h\Lambda) + \nu_1 h\Lambda w(T_2) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3 \le\\
&\quad \le \Bigg[w(Y_0)(1+\nu_1 h\Lambda)^2 +\\
&\quad + \left(\nu_1 h\Lambda \max_{l=0,1} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3\right)(1 + (1+\nu_1 h\Lambda))\Bigg](1+\nu_1 h\Lambda) +\\
&\quad + \nu_1 h\Lambda w(T_2) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3 \le\\
&\quad \le w(Y_0)(1+\nu_1 h\Lambda)^3 +\\
&\quad + \left(\nu_1 h\Lambda \max_{l=0,1,2} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3\right) \times\\
&\quad \times \left(1 + (1+\nu_1 h\Lambda)^2 + (1+\nu_1 h\Lambda)^3\right)
\end{aligned}$$

....

Thus, for each $k = 1, 2, \ldots, n$ we have

$$\begin{aligned}
&w(Y_k) \le w(Y_0) + (1+\nu_1 h\Lambda)^k +\\
&\quad + \left(\nu_1 h\Lambda \max_{l=0,1,\ldots,k-1} w(T_l) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^3\right)\sum_{i=0}^{k-1}(1+\nu_1 h\Lambda)^i.
\end{aligned}$$

But

$$\sum_{i=0}^{k-1}(1+\nu_i h\Lambda)^i = \frac{(1+\nu_i h\Lambda)^k - 1}{\nu_i h\Lambda} \le \frac{\exp(\nu_i kh\Lambda)}{\nu_i h\Lambda} \le$$

$$\le \frac{\exp(\nu_i nh\Lambda) - 1}{\nu_i h\Lambda} = \frac{\exp(\nu_i \eta_1^* \Lambda) - 1}{\nu_i h\Lambda},$$

where, according to (3.51),

$$\eta_1^* = \min\{\eta_0, \eta_1\}.$$

Hence

$$w(Y_k) \le Rw(Y_0) + S \max_{l=0,1,\ldots,k} w(T_l) + Qh^2, \tag{3.74}$$

where

$$R = \exp(\nu_1 \eta_1^* \Lambda), \quad S = R - 1, \quad Q = \frac{\exp(\nu_1 \eta_1^* \Lambda) - 1}{\nu_1 \Lambda}[w(\Psi(\Delta_t, \Delta_y)) + 2\alpha].$$

Taking into account that $T_0 = [0, 0]$, i.e. $w(T_0) = 0$, the inequality (3.70) follows immediately from (3.74). ■

For the two-stage implicit interval method of Runge-Kutta type we can prove

Theorem 3.5. *If* Y_k $(k = 0, 1, \ldots, n)$ *are obtained on the basis of the method* (3.49) – (3.50) *with* $m = 2$*, then for* h_0 *such that*

$$h_0 < \min\left\{1, \frac{1}{\Lambda|a_{11}|}, \frac{1}{\Lambda|a_{22}|}, \frac{1}{\Lambda\left(|a_{11}| + |a_{22}|\right) + \Lambda^2|a_{12}||a_{21}|}\right\} \tag{3.75}$$

we have

$$w(Y_k) \le Qh^p + Rw(Y_0) + S \max_{l=1,2,\ldots,n} w(T_l), \tag{3.76}$$

where $p \le 4$*, and Q, R and S denote some nonnegative constants.*

Proof. For $m = 2$ the formula (3.49) yields

$$\begin{aligned} w(Y_{k+1}) \le w(Y_k) &+ \\ &+ h\left(|w_1|w(K_{1k}) + |w_2|w(K_{2k})\right) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}, \end{aligned} \tag{3.77}$$

where $p \le 4$ and $k = 0, 1, \ldots, n-1$. On the basis of (3.63) we have

$$\begin{aligned} w(K_{1k}) &\le \Lambda[w(T_k)+w(Y_k)]+h\Lambda\left|a_{11}\right|w(K_{1k})+h\Lambda\left|a_{12}\right|w(K_{2k}),\\ w(K_{2k}) &\le \Lambda[w(T_k)+w(Y_k)]+h\Lambda\left|a_{21}\right|w(K_{1k})+h\Lambda\left|a_{22}\right|w(K_{2k}). \end{aligned}\tag{3.78}$$

From (3.66) it follows that the solution of the inequalities (3.78) is of the form

$$\begin{aligned} w(K_{1k}) &\le \frac{1-h\Lambda\left|a_{22}\right|+h\Lambda\left|a_{12}\right|}{\left(1-h\Lambda\left|a_{11}\right|\right)\left(1-h\Lambda\left|a_{22}\right|\right)-h^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|}\Lambda[w(T_k)+w(Y_k)],\\ w(K_{2k}) &\le \frac{1-h\Lambda\left|a_{11}\right|+h\Lambda\left|a_{21}\right|}{\left(1-h\Lambda\left|a_{11}\right|\right)\left(1-h\Lambda\left|a_{22}\right|\right)-h^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|}\Lambda[w(T_k)+w(Y_k)], \end{aligned}\tag{3.79}$$

if

$$h<\frac{1}{\Lambda\left|a_{11}\right|},\quad h<\frac{1}{\Lambda\left|a_{22}\right|},$$

$$1-h\Lambda\left(\left|a_{11}\right|+\left|a_{22}\right|\right)+h^2\Lambda^2\left(\left|a_{11}\right|\left|a_{22}\right|-\left|a_{12}\right|\left|a_{21}\right|\right)>0.$$

The first two inequalities are fulfilled from the assumption (3.75) and because of $h\le h_0$. The third inequality also follows from (3.75), because for $h\le h_0$ we have

$$h<\frac{1}{\Lambda\left(\left|a_{11}\right|+\left|a_{22}\right|\right)+\Lambda^2\left|a_{12}\right|\left|a_{21}\right|},$$

i.e.

$$1-h\Lambda\left(\left|a_{11}\right|+\left|a_{22}\right|\right)-h\Lambda^2\left|a_{12}\right|\left|a_{21}\right|>0.\tag{3.80}$$

Since $h<1$ (as a consequence of $h_0<1$), then $h^2<h$. Thus, from (3.80) it follows that

$$1-h\Lambda\left(\left|a_{11}\right|+\left|a_{22}\right|\right)-h^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|>0,$$

and hence, obviously,

$$1-h\Lambda\left(\left|a_{11}\right|+\left|a_{22}\right|\right)-h^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|+h^2\Lambda^2\left|a_{11}\right|\left|a_{22}\right|>0.$$

Taking into account that $h\le h_0$, from (3.79) we get

$$\begin{aligned} w(K_{1k}) &\le \frac{1+h_0\Lambda\left|a_{12}\right|}{\left(1-h_0\Lambda\left|a_{11}\right|\right)\left(1-h_0\Lambda\left|a_{22}\right|\right)-h_0^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|}\Lambda[w(T_k)+w(Y_k)],\\ w(K_{2k}) &\le \frac{1+h_0\Lambda\left|a_{21}\right|}{\left(1-h_0\Lambda\left|a_{11}\right|\right)\left(1-h_0\Lambda\left|a_{22}\right|\right)-h_0^2\Lambda^2\left|a_{12}\right|\left|a_{21}\right|}\Lambda[w(T_k)+w(Y_k)]. \end{aligned}$$

Using these estimates, from the inequality (3.77) we obtain

$$w(Y_{k+1}) \le w(Y_k)(1 + \nu_2 h\Lambda) + \nu_2 h\Lambda w(T_k) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}, \quad k = 0, 1, \ldots, n-1, \tag{3.81}$$

where

$$\nu_2 = \frac{|w_1|(1 + h_0\Lambda|a_{12}|) + |w_2|(1 + h_0\Lambda|a_{21}|)}{(1 - h_0\Lambda|a_{11}|)(1 - h_0\Lambda|a_{22}|) - h_0^2\Lambda^2|a_{12}||a_{21}|}.$$

Proceeding further as in the proof of the previous theorem, from (3.81) we get

$$w(Y_k) \le Rw(Y_0) + S \max_{l = 0, 1, \ldots, k} w(T_l) + Qh^p, \tag{3.82}$$

where

$$R = \exp(\nu_2 \eta_2^* \Lambda), \quad S = R - 1, \quad Q = \frac{\exp(\nu_2\eta_2^*\Lambda) - 1}{\nu_2\Lambda}[w(\Psi(\Delta_t, \Delta_y)) + 2\alpha],$$
$$\eta_2^* = \min\{\eta_0, \eta_1, \eta_2\}.$$

Since $w(T_0) = 0$, the inequality (3.76) is an obvious consequence of (3.82). ■

For the implicit three-stage method we have

Theorem 3.6. *If* Y_k $(k = 1, 2, \ldots, n)$ *are obtained on the basis of the method* (3.49) – (3.50) *with* $m = 3$, *then for* h_0 *such that*

$$h_0 < \min\left\{1, \frac{1}{\Lambda|a_{11}|}, \frac{1}{\Lambda|a_{22}|}, \frac{1}{\Lambda|a_{33}|}, \right.$$
$$\frac{1}{\Lambda(|a_{11}| + |a_{22}|) + \Lambda^2|a_{12}||a_{21}|},$$
$$\frac{1}{\Lambda(|a_{11}| + |a_{33}|) + \Lambda^2|a_{13}||a_{31}|}, \tag{3.83}$$
$$\frac{1}{\Lambda(|a_{22}| + |a_{33}|) + \Lambda^2|a_{23}||a_{32}|},$$
$$\left.\frac{1}{\Lambda\Omega_1 + \Lambda^2\Omega_2 + \Lambda^3\Omega_3}\right\},$$

where

$$\Omega_1 = |a_{11}| + |a_{22}| + |a_{33}|,$$

$$\Omega_2 = |a_{12}||a_{21}| + |a_{13}||a_{31}| + |a_{23}||a_{32}|,$$
$$\Omega_3 = |a_{11}||a_{22}||a_{33}| + |a_{12}||a_{23}||a_{31}| + |a_{13}||a_{21}||a_{32}|,$$

we have

$$w(Y_k) \le Qh^p + Rw(Y_0) + S \max_{l = 1, 2, \ldots, n} w(T_l), \tag{3.84}$$

where Q, R and S denote some nonnegative constants, and $p \le 6$.

Proof. From (3.49) for $m = 3$ we get

$$\begin{aligned} w(Y_{k+1}) = w(Y_k) = h\big(|w_1|w(K_{1k}) + |w_2|w(K_{2k}) + |w_3|w(K_{3k})\big) + \\ + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1}, \end{aligned} \tag{3.85}$$

where $p \le 6$, $k = 0, 1, \ldots, n-1$. On the basis of (3.50) we have

$$\begin{aligned} w(K_{ik}) \le \Lambda[w(T_k) + w(Y_k)] + \\ + h\Lambda\big(|a_{i1}|w(K_{1k}) + |a_{i2}|w(K_{2k}) + |a_{i3}|w(K_{3k})\big), \\ i = 1, 2, 3. \end{aligned} \tag{3.86}$$

From (3.67) it follows that if

$$\begin{aligned} & 1 - h\Lambda|a_{ii}| > 0, \quad i = 1, 2, 3, \\ & \big(1 - h\Lambda|a_{ii}|\big)\big(1 - h\Lambda|a_{jj}|\big) - h^2\Lambda^2|a_{ij}||a_{ji}| > 0, \quad i, j = 1, 2, 3, \quad i \ne j, \\ & \big(1 - h\Lambda|a_{11}|\big)\big(1 - h\Lambda|a_{22}|\big)\big(1 - h\Lambda|a_{33}|\big) - \\ & \quad - h^2\Lambda^2\big(|a_{12}||a_{21}| + |a_{13}||a_{31}| + |a_{23}||a_{32}|\big) - \\ & \quad - h^3\Lambda^3\big(|a_{12}||a_{23}||a_{31}| + |a_{13}||a_{21}||a_{32}| - \\ & \qquad - |a_{11}||a_{23}||a_{32}| - |a_{13}||a_{22}||a_{31}| - |a_{12}||a_{21}||a_{33}|\big) > 0, \end{aligned} \tag{3.87}$$

then the solution of inequalities (3.86) is of the form

$$w(K_{ik}) \le \frac{U_i}{U}\Lambda[w(T_k) + w(Y_k)], \quad i = 1, 2, 3, \tag{3.88}$$

where

$$U_i = \big(1 - h\Lambda|a_{jj}|\big)\big(1 - h\Lambda|a_{kk}|\big) + \big(1 + h\Lambda|a_{jk}| - h\Lambda|a_{kk}|\big)h\Lambda|a_{ij}| +$$

$$+\left(1+h\Lambda|a_{kj}|-h\Lambda|a_{jj}|\right)h\Lambda|a_{ik}|-h^2\Lambda^2|a_{jk}||a_{kj}|,$$
$$i, k, j = 1, 2, 3, \quad i \neq k, \quad k \neq j, \quad j \neq i,$$
$$U = \left(1-h\Lambda|a_{11}|\right)\left(1-h\Lambda|a_{22}|\right)\left(1-h\Lambda|a_{33}|\right)-$$
$$-h^2\Lambda^2\Big[\left(1-h\Lambda|a_{11}|\right)|a_{23}||a_{32}|+\left(1-h\Lambda|a_{22}|\right)|a_{13}||a_{31}|+$$
$$+\left(1-h\Lambda|a_{33}|\right)|a_{12}||a_{21}|\Big]-$$
$$-h^3\Lambda^3\left(|a_{12}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|\right).$$

Taking into account the assumption (3.83), the first six inequalities in (3.87) are self-evident. The last inequality in (3.87) is also an implication of this assumption, because from the inequality

$$h_0 < \frac{1}{\Lambda\Omega_1 + \Lambda^2\Omega_2 + \Lambda^3\Omega_3}$$

it follows that for $h \leq h_0$ we have

$$1-h\Lambda\left(|a_{11}|+|a_{22}|+|a_{33}|\right)-h\Lambda^2\left(|a_{12}||a_{21}|+|a_{13}||a_{31}|+|a_{23}||a_{32}|\right)-$$
$$-h\Lambda^3\left(|a_{11}||a_{22}||a_{33}|+|a_{13}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|\right)>0.$$

Since $h < 1$ (as a consequence of $h_0 < 1$), then $h^2 < h$ and $h^3 < h$. Thus, from the last inequality it follows that

$$1-h\Lambda\left(|a_{11}|+|a_{22}|+|a_{33}|\right)-h^2\Lambda^2\left(|a_{12}||a_{21}|+|a_{13}||a_{31}|+|a_{23}||a_{32}|\right)-$$
$$-h^3\Lambda^3\left(|a_{11}||a_{22}||a_{33}|+|a_{13}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|\right)>0,$$

and hence — all the more —

$$1-h\Lambda\left(|a_{11}|+|a_{22}|+|a_{33}|\right)-h^2\Lambda^2\left(|a_{12}||a_{21}|+|a_{13}||a_{31}|+|a_{23}||a_{32}|\right)+$$
$$+h^2\Lambda^2\left(|a_{11}||a_{22}|+|a_{11}||a_{33}|+|a_{22}||a_{33}|\right)-$$
$$-h^3\Lambda^3\left(|a_{11}||a_{22}||a_{33}|+|a_{13}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|\right)+$$
$$+h^3\Lambda^3\left(|a_{11}||a_{23}||a_{32}|+|a_{22}||a_{13}||a_{31}|+|a_{33}||a_{12}||a_{21}|\right)>0.$$

But the left-hand side of this inequality is equal to the left-hand side of the last inequality in (3.87). Taking into account that $h \leq h_0$, from (3.88) we get

$$w(K_{1k}) \le \frac{1 + h_0\Big[|a_{12}|\big(1 + h_0\Lambda|a_{23}|\big) + |a_{13}|\big(1 + h_0\Lambda|a_{32}|\big) + h_0^2\Lambda^2|a_{22}||a_{33}|\Big]}{\overline{U}} \times$$
$$\times\ \Lambda[w(T_k) + w(Y_k)],$$
$$w(K_{2k}) \le \frac{1 + h_0\Big[|a_{21}|\big(1 + h_0\Lambda|a_{13}|\big) + |a_{23}|\big(1 + h_0\Lambda|a_{31}|\big) + h_0^2\Lambda^2|a_{11}||a_{33}|\Big]}{\overline{U}} \times$$
$$\times\ \Lambda[w(T_k) + w(Y_k)],$$
$$w(K_{3k}) \le \frac{1 + h_0\Big[|a_{31}|\big(1 + h_0\Lambda|a_{12}|\big) + |a_{32}|\big(1 + h_0\Lambda|a_{21}|\big) + h_0^2\Lambda^2|a_{11}||a_{22}|\Big]}{\overline{U}} \times$$
$$\times\ \Lambda[w(T_k) + w(Y_k)],$$

where

$$\overline{U} = \big(1 - h_0\Lambda|a_{11}|\big)\big(1 - h_0\Lambda|a_{22}|\big)\big(1 - h_0\Lambda|a_{33}|\big) -$$
$$- h_0^2\Lambda^2\big(|a_{12}||a_{21}| + |a_{13}||a_{31}| + |a_{23}||a_{32}|\big) -$$
$$- h_0^3\Lambda^3\big(|a_{12}||a_{23}||a_{31}| + |a_{13}||a_{21}||a_{32}|\big).$$

Using these estimates, from the inequality (3.85) we obtain

$$w(Y_{k+1}) \le w(Y_k)(1 + \nu_3 h\Lambda) + \nu_3 h\Lambda w(T_k) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]h^{p+1},$$
$$k = 0, 1, \ldots, n-1,$$

where

$$\nu_3 = \frac{1}{\overline{U}}\Big\{|w_1|\Big(1 + h_0\Lambda\Big[|a_{12}|\big(1 + h_0\Lambda|a_{23}|\big) + |a_{13}|\big(1 + h_0\Lambda|a_{32}|\big)\Big] + h_0^2\Lambda^2|a_{22}||a_{33}|\Big) +$$
$$+ |w_2|\Big(1 + h_0\Lambda\Big[|a_{21}|\big(1 + h_0\Lambda|a_{13}|\big) + |a_{23}|\big(1 + h_0\Lambda|a_{31}|\big)\Big] + h_0^2\Lambda^2|a_{11}||a_{33}|\Big) +$$
$$+ |w_3|\Big(1 + h_0\Lambda\Big[|a_{31}|\big(1 + h_0\Lambda|a_{12}|\big) + |a_{32}|\big(1 + h_0\Lambda|a_{21}|\big)\Big] + h_0^2\Lambda^2|a_{11}||a_{22}|\Big)\Big\}.$$

Proceeding further, as in the proof of the theorem 3.4, we get

$$w(Y_k) \le Rw(Y_0) + S \max_{l = 0, 1, \ldots, k} w(T_l) + Qh^{p+1}, \qquad (3.89)$$

where

$$R = \exp(\nu_3\overset{*}{\eta}_3\Lambda), \quad S = R - 1, \quad Q = \frac{\exp(\nu_3\overset{*}{\eta}_3\Lambda) - 1}{\nu_3\Lambda}[w(\Psi(\Delta_t, \Delta_y)) + 2\alpha].$$

Taking into account that $w(T_0) = 0$, from (3.89) the inequality (3.84) follows immediately. ■

Finally, for the implicit four-stage method we can prove

Theorem 3.7. *If* Y_k ($k = 1, 2, \ldots, n$) *are obtained on the basis of the method* (3.49) – (3.50) *with* $m = 4$, *then for* h_0 *such that*

$$\begin{aligned} h_0 < \min\Bigg\{ & 1, \frac{1}{\Lambda|a_{11}|}, \frac{1}{\Lambda|a_{22}|}, \frac{1}{\Lambda|a_{33}|}, \frac{1}{\Lambda|a_{44}|}, \\ & \frac{1}{\Lambda(|a_{11}|+|a_{22}|)+\Lambda^2|a_{12}||a_{21}|}, \\ & \frac{1}{\Lambda(|a_{11}|+|a_{33}|)+\Lambda^2|a_{13}||a_{31}|}, \\ & \frac{1}{\Lambda(|a_{11}|+|a_{44}|)+\Lambda^2|a_{14}||a_{41}|}, \\ & \frac{1}{\Lambda(|a_{22}|+|a_{33}|)+\Lambda^2|a_{23}||a_{32}|}, \\ & \frac{1}{\Lambda(|a_{22}|+|a_{44}|)+\Lambda^2|a_{24}||a_{42}|}, \\ & \frac{1}{\Lambda(|a_{33}|+|a_{44}|)+\Lambda^2|a_{34}||a_{43}|}, \\ & \frac{1}{\Lambda\Theta_{11}+\Lambda^2\Theta_{12}+\Lambda^3\Theta_{13}}, \frac{1}{\Lambda\Theta_{21}+\Lambda^2\Theta_{22}+\Lambda^3\Theta_{23}}, \\ & \frac{1}{\Lambda\Theta_{31}+\Lambda^2\Theta_{32}+\Lambda^3\Theta_{33}}, \frac{1}{\Lambda\Theta_{41}+\Lambda^2\Theta_{42}+\Lambda^3\Theta_{43}}, \\ & \frac{1}{\Lambda\Xi_1+\Lambda^2\Xi_2+\Lambda^3\Xi_3+\Lambda^4\Xi_4}\Bigg\}, \end{aligned} \tag{3.90}$$

where

$$\Theta_{11} = |a_{11}|+|a_{22}|+|a_{33}|,$$

$$\Theta_{12} = |a_{12}||a_{21}|+|a_{13}||a_{31}|+|a_{23}||a_{32}|,$$

$$\Theta_{13} = |a_{11}||a_{22}||a_{33}|+|a_{12}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|,$$

$$
\begin{aligned}
&\Theta_{21} = |a_{11}| + |a_{22}| + |a_{44}|,\\
&\Theta_{22} = |a_{12}||a_{21}| + |a_{14}||a_{41}| + |a_{24}||a_{42}|,\\
&\Theta_{23} = |a_{11}||a_{22}||a_{44}| + |a_{12}||a_{24}||a_{41}| + |a_{14}||a_{21}||a_{42}|,\\
&\Theta_{31} = |a_{11}| + |a_{33}| + |a_{44}|,\\
&\Theta_{32} = |a_{13}||a_{31}| + |a_{14}||a_{41}| + |a_{34}||a_{43}|,\\
&\Theta_{33} = |a_{11}||a_{33}||a_{44}| + |a_{13}||a_{34}||a_{41}| + |a_{13}||a_{31}||a_{43}|,\\
&\Theta_{41} = |a_{22}| + |a_{33}| + |a_{44}|,\\
&\Theta_{42} = |a_{23}||a_{32}| + |a_{24}||a_{42}| + |a_{34}||a_{43}|,\\
&\Theta_{43} = |a_{22}||a_{33}||a_{44}| + |a_{23}||a_{34}||a_{42}| + |a_{24}||a_{32}||a_{43}|,\\
&\Xi_1 = |a_{11}| + |a_{22}| + |a_{33}| + |a_{44}|,\\
&\Xi_2 = |a_{12}||a_{21}| + |a_{13}||a_{31}| + |a_{14}||a_{41}| +\\
&\qquad + |a_{23}||a_{32}| + |a_{24}||a_{42}| + |a_{34}||a_{43}|,\\
&\Xi_3 = |a_{11}||a_{22}||a_{33}| + |a_{11}||a_{22}||a_{44}| + |a_{11}||a_{33}||a_{44}| +\\
&\qquad + |a_{12}||a_{23}||a_{31}| + |a_{12}||a_{24}||a_{41}| + |a_{13}||a_{21}||a_{32}| +\\
&\qquad + |a_{13}||a_{34}||a_{41}| + |a_{14}||a_{21}||a_{42}| + |a_{14}||a_{31}||a_{43}| +\\
&\qquad + |a_{22}||a_{33}||a_{44}| + |a_{23}||a_{34}||a_{42}| + |a_{24}||a_{32}||a_{43}|,\\
&\Xi_4 = |a_{11}||a_{22}||a_{34}||a_{43}| + |a_{11}||a_{23}||a_{32}||a_{44}| +\\
&\qquad + |a_{11}||a_{24}||a_{33}||a_{42}| + |a_{12}||a_{21}||a_{33}||a_{44}| +\\
&\qquad + |a_{12}||a_{23}||a_{34}||a_{41}| + |a_{12}||a_{24}||a_{31}||a_{43}| +\\
&\qquad + |a_{13}||a_{21}||a_{34}||a_{42}| + |a_{13}||a_{22}||a_{31}||a_{44}| +\\
&\qquad + |a_{13}||a_{24}||a_{32}||a_{41}| + |a_{14}||a_{21}||a_{32}||a_{43}| +\\
&\qquad + |a_{14}||a_{22}||a_{33}||a_{41}| + |a_{14}||a_{23}||a_{31}||a_{42}|,
\end{aligned}
$$

we have

$$
w(Y_k) \le Qh^p + Rw(Y_0) + S \max_{l=1,2,\ldots,n} w(T_l), \tag{3.91}
$$

where Q, R and S denote some nonnegative constants, and $p \le 8$.

Proof. For $m = 4$ the formula (3.49) yields

$$w(Y_{k+1}) \le w(Y_k) + h\sum_{i=1}^{4} \left| w_i \right| w(K_{ik}) + [w(\varPsi(\varDelta_t, \varDelta_y)) + 2\alpha]h^{p+1}, \tag{3.92}$$

where $p \le 8$, $k = 0, 1, \ldots, n-1$. On the basis of (3.50) we have

$$\begin{gathered} w(K_{ik}) \le \varLambda[w(T_k) + w(Y_k)] + h\varLambda\sum_{j=1}^{4} \left| a_{ij} \right| w(K_{jk}), \\ i = 1, 2, 3, 4. \end{gathered} \tag{3.93}$$

From (3.68) it follows that if

$$\begin{aligned} & 1 - h\varLambda\left| a_{ii} \right| > 0, \quad i = 1, 2, 3, 4, \\ & \left(1 - h\varLambda\left| a_{ii} \right|\right)\left(1 - h\varLambda\left| a_{jj} \right|\right) - h^2\varLambda^2\left| a_{ij} \right|\left| a_{ji} \right|, \quad i, j = 1, 2, 3, 4, \quad i \ne j, \\ & \left(1 - h\varLambda\left| a_{ii} \right|\right)\left(1 - h\varLambda\left| a_{jj} \right|\right)\left(1 - h\varLambda\left| a_{kk} \right|\right) - \\ & \quad - h^2\varLambda^2\Big[\left(1 - h\varLambda\left| a_{ii} \right|\right)\left| a_{jk} \right|\left| a_{kj} \right| + \left(1 - h\varLambda\left| a_{jj} \right|\right)\left| a_{ik} \right|\left| a_{ki} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{kk} \right|\right)\left| a_{ij} \right|\left| a_{ji} \right|\Big] - \\ & \quad - h^3\varLambda^3\left(\left| a_{ij} \right|\left| a_{jk} \right|\left| a_{ki} \right| + \left| a_{ik} \right|\left| a_{ji} \right|\left| a_{kj} \right|\right) > 0, \\ & \quad i, j, k = 1, 2, 3, 4, \quad i \ne j, \quad j \ne k, \quad k \ne i, \\ & V \equiv \left(1 - h\varLambda\left| a_{11} \right|\right)\left(1 - h\varLambda\left| a_{22} \right|\right)\left(1 - h\varLambda\left| a_{33} \right|\right)\left(1 - h\varLambda\left| a_{44} \right|\right) - \\ & \quad - h^2\varLambda^2\Big[\left(1 - h\varLambda\left| a_{11} \right|\right)\left(1 - h\varLambda\left| a_{22} \right|\right)\left| a_{34} \right|\left| a_{43} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{11} \right|\right)\left(1 - h\varLambda\left| a_{33} \right|\right)\left| a_{24} \right|\left| a_{42} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{11} \right|\right)\left(1 - h\varLambda\left| a_{44} \right|\right)\left| a_{23} \right|\left| a_{32} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{22} \right|\right)\left(1 - h\varLambda\left| a_{33} \right|\right)\left| a_{14} \right|\left| a_{41} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{22} \right|\right)\left(1 - h\varLambda\left| a_{44} \right|\right)\left| a_{13} \right|\left| a_{31} \right| + \\ & \qquad + \left(1 - h\varLambda\left| a_{33} \right|\right)\left(1 - h\varLambda\left| a_{44} \right|\right)\left| a_{12} \right|\left| a_{21} \right|\Big] - \\ & \quad - h^3\varLambda^3\Big[\left(1 - h\varLambda\left| a_{11} \right|\right)\left(\left| a_{23} \right|\left| a_{34} \right|\left| a_{41} \right| + \left| a_{24} \right|\left| a_{32} \right|\left| a_{43} \right|\right) + \\ & \qquad + \left(1 - h\varLambda\left| a_{22} \right|\right)\left(\left| a_{13} \right|\left| a_{34} \right|\left| a_{41} \right| + \left| a_{14} \right|\left| a_{31} \right|\left| a_{43} \right|\right) + \\ & \qquad + \left(1 - h\varLambda\left| a_{33} \right|\right)\left(\left| a_{12} \right|\left| a_{24} \right|\left| a_{41} \right| + \left| a_{14} \right|\left| a_{21} \right|\left| a_{42} \right|\right) + \end{aligned} \tag{3.94}$$

$$
\begin{aligned}
&+\left(1-h\Lambda|a_{44}|\right)\left(|a_{12}||a_{23}||a_{31}|+|a_{13}||a_{21}||a_{32}|\right)\Big]-\\
&-h^4\Lambda^4\Big[|a_{12}|\left(|a_{23}||a_{34}||a_{41}|+|a_{24}||a_{31}||a_{43}|-|a_{21}||a_{34}||a_{43}|\right)+\\
&\qquad+|a_{13}|\left(|a_{21}||a_{34}||a_{42}|+|a_{24}||a_{32}||a_{41}|-|a_{24}||a_{31}||a_{42}|\right)+\\
&\qquad+|a_{14}|\left(|a_{21}||a_{32}||a_{43}|+|a_{23}||a_{31}||a_{42}|-|a_{23}||a_{32}||a_{41}|\right)\Big]>0,
\end{aligned}
$$

then

$$
w(K_{ik})\le\frac{V_i}{V}\Lambda[w(T_k)+w(Y_k)],\quad i=1,2,3,4, \tag{3.95}
$$

where

$$
\begin{aligned}
V_i={}&\left(1-h\Lambda|a_{jj}|\right)\left(1-h\Lambda|a_{kk}|\right)\left(1-h\Lambda|a_{ll}|\right)+\\
&+\left[1+h\Lambda\left(|a_{jk}|+|a_{jl}|\right)-h\Lambda\left(|a_{kk}|+|a_{ll}|\right)\right]h\Lambda|a_{ij}|+\\
&+\left[1+h\Lambda\left(|a_{kj}|+|a_{kl}|\right)-h\Lambda\left(|a_{jj}|+|a_{ll}|\right)\right]h\Lambda|a_{ik}|+\\
&+\left[1+h\Lambda\left(|a_{lj}|+|a_{lk}|\right)-h\Lambda\left(|a_{jj}|+|a_{kk}|\right)\right]h\Lambda|a_{il}|-\\
&-\left(1+h\Lambda|a_{ij}|-h\Lambda|a_{jj}|\right)h^2\Lambda^2|a_{kl}||a_{lk}|-\\
&-\left(1+h\Lambda|a_{ik}|-h\Lambda|a_{kk}|\right)h^2\Lambda^2|a_{jl}||a_{lj}|-\\
&-\left(1+h\Lambda|a_{il}|-h\Lambda|a_{ll}|\right)h^2\Lambda^2|a_{jk}||a_{kj}|+\\
&+h^3\Lambda^3\Big\{|a_{ij}|\Big[|a_{jk}|\left(|a_{kl}|-|a_{ll}|\right)+|a_{jl}|\left(|a_{kl}|-|a_{kk}|\right)+|a_{kk}||a_{ll}|\Big]+\\
&\qquad+|a_{ik}|\Big[|a_{kl}|\left(|a_{lj}|-|a_{jj}|\right)+|a_{kj}|\left(|a_{jl}|-|a_{ll}|\right)+|a_{jj}||a_{kk}|\Big]+\\
&\qquad+|a_{il}|\Big[|a_{lj}|\left(|a_{jk}|-|a_{kk}|\right)+|a_{lk}|\left(|a_{kj}|-|a_{jj}|\right)+|a_{jj}||a_{kk}|\Big]\Big\}-\\
&-h^3\Lambda^3\left(|a_{jk}||a_{kl}||a_{lj}|+|a_{jl}||a_{kj}||a_{lk}|\right),\\
&\quad i,j,k,l=1,2,3,4,\quad i\ne j,\quad j\ne k,\quad k\ne l,\quad l\ne i.
\end{aligned}
$$

and V is given by (3.94).

The first fourteen inequalities in (3.94) are fulfilled from the assumption (3.90) and because of $h\le h_0$. The last inequality in (3.94) also follows from (3.90), because for $h\le h_0$ we have

$$h < \frac{1}{\Lambda\Xi_1 + \Lambda^2\Xi_2 + \Lambda^3\Xi_3 + \Lambda^4\Xi_4},$$

i.e.

$$1 - h\Lambda\Xi_1 - h\Lambda^2\Xi_2 - h\Lambda^3\Xi_3 - h\Lambda^4\Xi_4 > 0.$$

Since $h \le h_0 < 1$, then $h^s < h$ $(s = 2, 3, 4)$. Thus

$$1 - h\Lambda\Xi_1 - h^2\Lambda^2\Xi_2 - h^3\Lambda^3\Xi_3 - h^4\Lambda^4\Xi_4 > 0.$$

The last inequality in (3.94) is a consequence of the above inequality.

For $h \le h_0$ from (3.93) we have

$$\begin{aligned}
w(K_{1k}) \le \frac{1}{\overline{V}}\Big\{&1 + h_0\Lambda\big(|a_{12}| + |a_{13}| + |a_{14}|\big) + \\
&+ h_0^2\Lambda^2\Big[|a_{12}|\big(|a_{23}| + |a_{24}|\big) + |a_{13}|\big(|a_{32}| + |a_{34}|\big) + \\
&\qquad + |a_{14}|\big(|a_{42}| + |a_{43}|\big) + |a_{22}||a_{33}| + |a_{22}||a_{44}| + |a_{33}||a_{44}|\Big] + \\
&+ h_0^3\Lambda^3\Big[|a_{12}|\big(|a_{23}||a_{34}| + |a_{24}||a_{43}| + |a_{33}||a_{44}|\big) + \\
&\qquad + |a_{13}|\big(|a_{22}||a_{44}| + |a_{24}||a_{32}| + |a_{34}||a_{42}|\big) + \\
&\qquad + |a_{14}|\big(|a_{22}||a_{33}| + |a_{23}||a_{42}| + |a_{32}||a_{43}|\big) + \\
&\qquad + |a_{22}||a_{34}||a_{43}| + |a_{23}||a_{32}||a_{44}| + |a_{24}||a_{33}||a_{42}|\Big]\Big\} \times \\
&\times \Lambda[w(T_k) + w(Y_k)],
\end{aligned}$$

$$\begin{aligned}
w(K_{2k}) \le \frac{1}{\overline{V}}\Big\{&1 + h_0\Lambda\big(|a_{21}| + |a_{23}| + |a_{24}|\big) + \\
&+ h_0^2\Lambda^2\Big[|a_{11}|\big(|a_{33}| + |a_{44}|\big) + |a_{21}|\big(|a_{13}| + |a_{14}|\big) + \\
&\qquad + |a_{23}|\big(|a_{31}| + |a_{34}|\big) + |a_{24}||a_{41}| + |a_{24}||a_{43}| + |a_{33}||a_{44}|\Big] + \\
&+ h_0^3\Lambda^3\Big[|a_{11}|\big(|a_{23}||a_{44}| + |a_{24}||a_{33}| + |a_{34}||a_{43}|\big) + \\
&\qquad + |a_{13}|\big(|a_{21}||a_{34}| + |a_{24}||a_{41}| + |a_{31}||a_{44}|\big) + \\
&\qquad + |a_{14}|\big(|a_{21}||a_{43}| + |a_{23}||a_{31}| + |a_{33}||a_{41}|\big) + \\
&\qquad + |a_{21}||a_{33}||a_{44}| + |a_{23}||a_{34}||a_{41}| + |a_{24}||a_{31}||a_{43}|\Big]\Big\} \times \\
&\times \Lambda[w(T_k) + w(Y_k)],
\end{aligned}$$

$$
\begin{aligned}
w(K_{3k}) \le \frac{1}{\overline{V}}\Big\{ & 1 + h_0\Lambda\big(|a_{31}| + |a_{32}| + |a_{34}|\big) + \\
& + h_0^2\Lambda^2\Big[|a_{11}|\big(|a_{22}| + |a_{44}|\big) + |a_{31}|\big(|a_{12}| + |a_{14}|\big) + \\
& \qquad + |a_{32}|\big(|a_{21}| + |a_{24}|\big) + |a_{22}||a_{44}| + |a_{34}||a_{41}| + |a_{34}||a_{42}|\Big] + \\
& + h_0^3\Lambda^3\Big[|a_{11}|\big(|a_{22}||a_{34}| + |a_{24}||a_{42}| + |a_{32}||a_{44}|\big) + \\
& \qquad + |a_{12}|\big(|a_{21}||a_{44}| + |a_{24}||a_{31}| + |a_{34}||a_{41}|\big) + \\
& \qquad + |a_{14}|\big(|a_{21}||a_{34}| + |a_{22}||a_{41}| + |a_{31}||a_{42}|\big) + \\
& \qquad + |a_{21}||a_{34}||a_{42}| + |a_{22}||a_{31}||a_{44}| + |a_{24}||a_{32}||a_{41}|\Big] \times \\
& \times \Lambda[w(T_k) + w(Y_k)],
\end{aligned}
$$

$$
\begin{aligned}
w(K_{4k}) \le \frac{1}{\overline{V}}\Big\{ & 1 + h_0\Lambda\big(|a_{41}| + |a_{42}| + |a_{43}|\big) + \\
& + h_0^2\Lambda^2\Big[|a_{11}|\big(|a_{22}| + |a_{33}|\big) + |a_{41}|\big(|a_{12}| + |a_{13}|\big) + \\
& \qquad + |a_{42}|\big(|a_{21}| + |a_{23}|\big) + |a_{22}||a_{33}| + |a_{31}||a_{43}| + |a_{32}||a_{43}|\Big] + \\
& + h_0^3\Lambda^3\Big[|a_{11}|\big(|a_{22}||a_{43}| + |a_{23}||a_{32}| + |a_{33}||a_{42}|\big) + \\
& \qquad + |a_{12}|\big(|a_{21}||a_{33}| + |a_{23}||a_{41}| + |a_{31}||a_{43}|\big) + \\
& \qquad + |a_{13}|\big(|a_{21}||a_{42}| + |a_{22}||a_{31}| + |a_{32}||a_{41}|\big) + \\
& \qquad + |a_{21}||a_{32}||a_{43}| + |a_{22}||a_{33}||a_{41}| + |a_{23}||a_{31}||a_{42}|\Big] \times \\
& \times \Lambda[w(T_k) + w(Y_k)],
\end{aligned}
$$

where

$$
\begin{aligned}
\overline{V} = & \big(1 - h_0\Lambda|a_{11}|\big)\big(1 - h_0\Lambda|a_{22}|\big)\big(1 - h_0\Lambda|a_{33}|\big)\big(1 - h_0\Lambda|a_{44}|\big) - \\
& - h_0^2\Lambda^2\big(|a_{12}||a_{21}| + |a_{13}||a_{31}| + |a_{14}||a_{41}| + |a_{23}||a_{32}| + \\
& \qquad + |a_{24}||a_{42}| + |a_{34}||a_{43}|\big) - \\
& - h_0^3\Lambda^3\big(|a_{12}||a_{23}||a_{31}| + |a_{12}||a_{24}||a_{41}| + |a_{13}||a_{21}||a_{32}| + \\
& \qquad + |a_{13}||a_{34}||a_{41}| + |a_{14}||a_{21}||a_{42}| + |a_{14}||a_{31}||a_{43}| + \\
& \qquad + |a_{23}||a_{34}||a_{42}| + |a_{24}||a_{32}||a_{43}|\big) -
\end{aligned}
$$

$$
\begin{aligned}
- h_0^4 \Lambda^4 \big(& |a_{12}||a_{23}||a_{34}||a_{41}| + |a_{12}||a_{24}||a_{31}||a_{43}| + \\
& + |a_{13}||a_{21}||a_{34}||a_{42}| + |a_{13}||a_{24}||a_{32}||a_{41}| + \\
& + |a_{14}||a_{21}||a_{32}||a_{43}| + |a_{14}||a_{23}||a_{31}||a_{42}| \big).
\end{aligned}
$$

Using these estimates, from the inequality (3.92) we obtain

$$
w(Y_{k+1}) \le w(Y_k)(1 + \nu_4 h \Lambda) + \nu_4 h \Lambda w(T_k) + [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha] h^{p+1},
$$
$$
k = 0, 1, \ldots, n-1,
$$

where

$$
\nu_4 = \frac{1}{\overline{V}} \sum_{i=1}^{4} |w_i| \left(1 + \sum_{j=1}^{3} A_{ij} h_0^j \Lambda_j \right),
$$

and where

$$
\begin{aligned}
A_{11} = {} & |a_{12}| + |a_{13}| + |a_{14}|, \\
A_{12} = {} & |a_{12}|(|a_{23}| + |a_{24}|) + |a_{13}|(|a_{32}| + |a_{34}|) + |a_{14}|(|a_{42}| + |a_{43}|) + \\
& + |a_{22}||a_{33}| + |a_{22}||a_{44}| + |a_{33}||a_{44}|, \\
A_{13} = {} & |a_{12}|(|a_{23}||a_{34}| + |a_{24}||a_{43}| + |a_{33}||a_{44}|) + \\
& + |a_{13}|(|a_{22}||a_{44}| + |a_{24}||a_{32}| + |a_{34}||a_{42}|) + \\
& + |a_{14}|(|a_{22}||a_{33}| + |a_{23}||a_{42}| + |a_{32}||a_{43}|) + \\
& + |a_{22}||a_{34}||a_{43}| + |a_{23}||a_{32}||a_{44}| + |a_{24}||a_{33}||a_{42}|, \\
A_{21} = {} & |a_{21}| + |a_{23}| + |a_{24}|, \\
A_{22} = {} & |a_{21}|(|a_{13}| + |a_{14}|) + |a_{23}|(|a_{31}| + |a_{34}|) + |a_{24}|(|a_{41}| + |a_{43}|) + \\
& + |a_{11}||a_{33}| + |a_{11}||a_{44}| + |a_{33}||a_{44}|, \\
A_{23} = {} & |a_{21}|(|a_{13}||a_{34}| + |a_{14}||a_{43}| + |a_{33}||a_{44}|) + \\
& + |a_{23}|(|a_{11}||a_{44}| + |a_{14}||a_{31}| + |a_{34}||a_{41}|) + \\
& + |a_{24}|(|a_{11}||a_{33}| + |a_{13}||a_{41}| + |a_{31}||a_{43}|) + \\
& + |a_{11}||a_{34}||a_{43}| + |a_{13}||a_{31}||a_{44}| + |a_{14}||a_{33}||a_{41}|, \\
A_{31} = {} & |a_{31}| + |a_{33}| + |a_{34}|,
\end{aligned}
$$

$$\begin{aligned}
A_{32} &= |a_{31}|(|a_{12}|+|a_{14}|)+|a_{32}|(|a_{21}|+|a_{24}|)+|a_{34}|(|a_{41}|+|a_{42}|)+\\
&\quad+|a_{11}||a_{22}|+|a_{11}||a_{44}|+|a_{22}||a_{44}|,\\
A_{33} &= |a_{31}|(|a_{12}||a_{24}|+|a_{14}||a_{42}|+|a_{22}||a_{44}|)+\\
&\quad+|a_{32}|(|a_{11}||a_{44}|+|a_{14}||a_{21}|+|a_{24}||a_{41}|)+\\
&\quad+|a_{34}|(|a_{11}||a_{22}|+|a_{12}||a_{41}|+|a_{21}||a_{42}|)+\\
&\quad+|a_{11}||a_{24}||a_{42}|+|a_{12}||a_{21}||a_{44}|+|a_{14}||a_{22}||a_{41}|,\\
A_{41} &= |a_{41}|+|a_{42}|+|a_{43}|,\\
A_{42} &= |a_{41}|(|a_{12}|+|a_{13}|)+|a_{42}|(|a_{21}|+|a_{23}|)+|a_{43}|(|a_{31}|+|a_{32}|)+\\
&\quad+|a_{11}||a_{22}|+|a_{11}||a_{33}|+|a_{22}||a_{33}|,\\
A_{43} &= |a_{41}|(|a_{12}||a_{23}|+|a_{13}||a_{32}|+|a_{22}||a_{33}|)+\\
&\quad+|a_{42}|(|a_{11}||a_{33}|+|a_{13}||a_{21}|+|a_{23}||a_{31}|)+\\
&\quad+|a_{43}|(|a_{11}||a_{22}|+|a_{12}||a_{31}|+|a_{21}||a_{32}|)+\\
&\quad+|a_{11}||a_{23}||a_{32}|+|a_{12}||a_{21}||a_{33}|+|a_{13}||a_{22}||a_{31}|.
\end{aligned}$$

Proceeding further, as in the proof of the theorem 3.4, we get

$$w(Y_k) \le Rw(Y_0) + S \max_{l=0,1,\ldots,k} w(T_l) + Qh^p, \tag{3.96}$$

where

$$R = \exp(\nu_4 \overset{*}{\eta_4} \Lambda), \quad S = R - 1, \quad Q = \frac{\exp(\nu_4 \overset{*}{\eta_4} \Lambda) - 1}{\nu_4 \Lambda}\left[w(\Psi(\Delta_t, \Delta_y)) + 2\alpha\right].$$

Taking into account that $T_0 = [0, 0]$, from (3.96) we get (3.92). ■

3.4. Finding the Integration Interval in Floating-Point Interval Arithmetic

If we denote the numbers ξ_i ($i = 0, 2, \ldots, m$) occurring in explicit methods (see (3.35) – (3.38)) by η_i, i.e. as in implicit methods (see (3.51) – (3.53)), then we can write that in both kinds of methods the step size h, where $0 < h \le h_0$, should be calculated from the formula [123]

$$h = \frac{\eta_m^*}{n},$$

where

$$\eta_m^* = \min\{\eta_0, \eta_2, \dots, \eta_m\},$$

for explicit methods, and

$$\eta_m^* = \min\{\eta_0, \eta_1, \dots, \eta_m\},$$

for implicit ones. According to (3.37), (3.38), (3.52) and (3.53), the numbers $\eta_1 > 0, \eta_2 > 0, \dots, \eta_m > 0$ are evaluated in such a way that

$$Y_0 + \eta_i c_i F(\Delta_t, \Delta_y) \subset \Delta_y, \tag{3.97}$$

and the number $\eta_0 > 0$ – from the relation

$$Y_0 + \eta_0 \sum_{i=1}^{m} w_i F(\Delta_t, \Delta_y) + \Big(\Psi(\Delta_t, \Delta_y) + [-\alpha, \alpha]\Big) h_0^p \subset \Delta_y, \tag{3.98}$$

for $Y_0 \subset \Delta_y$ and $y_0 \in Y_0$. (In each case the symbol ⊂ denotes proper inclusion.)

Since the number $\eta_m^* > 0$ should be found in floating-point interval arithmetic, let us consider first the relation (3.97). This relation can be written in the form

$$a + \gamma c \subset b, \tag{3.99}$$

where γ is the unknown number, and where – from one of the assumptions – $a \subset b$ (proper inclusion). If

$$a = \left[\underline{a}, \overline{a}\right], \quad b = \left[\underline{b}, \overline{b}\right], \quad c = \left[\underline{c}, \overline{c}\right],$$

then the inclusion (3.99) may be written as

$$\left[\underline{a} + \gamma \underline{c}, \overline{a} + \gamma \overline{c}\right] \subset \left[\underline{b}, \overline{b}\right].$$

Thus, the number γ should be evaluated in such a way that

$$\underline{a} + \gamma \underline{c} > \underline{b} \quad \text{and} \quad \overline{a} + \gamma \overline{c} < \overline{b}.$$

Four cases are possible:

1° $\underline{c} = 0$ and $\overline{c} = 0 \Rightarrow \gamma$ can be an arbitrary number,

2° $\underline{c} \ge 0$ and $\overline{c} > 0 \Rightarrow \gamma < \dfrac{\overline{b} - \overline{a}}{\overline{c}}$,

3° $\underline{c} < 0$ and $\overline{c} \le 0 \Rightarrow \gamma < \dfrac{\underline{b} - \underline{a}}{\underline{c}}$, (3.100)

4° $\underline{c} < 0$ and $\overline{c} > 0 \Rightarrow \gamma < \min\left\{\frac{\underline{b} - \underline{a}}{\underline{c}}, \frac{\overline{b} - \overline{a}}{\overline{c}}\right\}$.

From (3.100) it follows that to find γ in floating-point interval arithmetic we should complete the following steps:

- calculate the right-hand side of the appropriate inequality treating all values as intervals with the width equal to 0,
- from the result-interval obtained take the left endpoint (mark it by $\underline{\gamma}$),
- accept as γ the previous machine number with respect to $\underline{\gamma}$.

In an N-dimensional case we have more calculations, but the idea is the same. In this case the relations (3.97) are of the form

$$Y_{s0} + \eta_i c_i F_s(\varDelta_t, \varDelta_{y_1}, \varDelta_{y_2}, \dots, \varDelta_{y_N}) \subset \varDelta_{y_s},$$
$$s = 1, 2, \dots, N,$$

i.e.

$$\left[\underline{Y}_{s0}, \overline{Y}_{s0}\right] + \eta_i\left[\underline{c}_i, \overline{c}_i\right]\left[\underline{F}_s, \overline{F}_s\right] \subset \left[\underline{\varDelta}_{y_s}, \overline{\varDelta}_{y_s}\right].$$

Performing all operations in interval arithmetic, we have

$$\left[\underline{Y}_{s0}, \overline{Y}_{s0}\right] + \eta_i\left[\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}, \max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}\right] \subset$$
$$\subset \left[\underline{\varDelta}_{y_s}, \overline{\varDelta}_{y_s}\right],$$

i.e.

$$\left[\underline{Y}_{s0} + \eta_i \min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}, \overline{Y}_{s0} + \eta_i \max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}\right] \subset$$
$$\subset \left[\underline{\varDelta}_{y_s}, \overline{\varDelta}_{y_s}\right],$$

from which we get two inequalities

$$\eta_i \min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} > \underline{\varDelta}_{y_s} - \underline{Y}_{s0},$$
$$\eta_i \max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} < \overline{\varDelta}_{y_s} - \overline{Y}_{s0},$$

whose right-hand sides are non-negative. From the previous considerations (see (3.100)) it follows that we must take into account four possibilities:

1° $\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} = \max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} = 0 \Rightarrow \eta_i$ can be an arbitrary number,

2° $\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} \ge 0$ and $\max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} > 0$

$$\Rightarrow \eta_i < \frac{\overline{\varDelta}_{y_s} - \overline{Y}_{s0}}{\max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}},$$

3° $\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} < 0$ and $\max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} \le 0$

$$\Rightarrow \eta_i < \frac{\underline{\varDelta}_{y_s} - \underline{Y}_{s0}}{\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}},$$

4° $\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} < 0$ and $\max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\} > 0$

$$\Rightarrow \eta_i < \min\left\{\frac{\underline{\varDelta}_{y_s} - \underline{Y}_{s0}}{\min\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}}, \frac{\overline{\varDelta}_{y_s} - \overline{Y}_{s0}}{\max\left\{\underline{c}_i\underline{F}_s, \underline{c}_i\overline{F}_s, \overline{c}_i\underline{F}_s, \overline{c}_i\overline{F}_s\right\}}\right\}.$$

From the above relations one should find the numbers η_i ($i = 2, 3, \ldots, m$ for explicit methods and $i = 1, 2, \ldots, m$ for implicit ones) for each $s = 1, 2, \ldots, N$, and then take the smallest one.

Now, let us consider the inclusion (3.98). In a one-dimensional case this inclusion is of the form

$$a + \gamma \sum_i c_i + d \subset b, \tag{3.101}$$

where $a \subset b$ (proper inclusion). Note that if the endpoints of d are not sufficiently small numbers (with respect to their absolute values), then the number γ may not exist.

For $a = \left[\underline{a}, \overline{a}\right]$, $b = \left[\underline{b}, \overline{b}\right]$, $c_i = \left[\underline{c}_i, \overline{c}_i\right]$ and $d = \left[\underline{d}, \overline{d}\right]$ the relation (3.101) may be rewritten in the form

$$\left[\underline{a} + \gamma \sum_i \underline{c}_i + \underline{d}, \overline{a} + \gamma \sum_i \overline{c}_i + \overline{d}\right] \subset \left[\underline{b}, \overline{b}\right],$$

from which it follows that

$$\gamma \sum_i \underline{c}_i > \underline{b} - \underline{a} - \underline{d} \text{ and } \gamma \sum_i \overline{c}_i < \overline{b} - \overline{a} - \overline{d}.$$

Let us note that we are able to evaluate $\gamma > 0$ only if

$$\overline{b} - \overline{a} - \overline{d} > 0 \text{ and } \underline{b} - \underline{a} - \underline{d} > 0.$$

Proceeding as previously, we get four possibilities:

$$1^\circ \ \sum_i \underline{c}_i = 0 \text{ and } \sum_i \overline{c}_i = 0 \Rightarrow \gamma \text{ can be an arbitrary number},$$

$$2^\circ \ \sum_i \underline{c}_i \geq 0 \text{ and } \sum_i \overline{c}_i > 0 \Rightarrow \gamma < \frac{\overline{b} - \overline{a} - \overline{d}}{\sum_i \overline{c}_i},$$

$$3^\circ \ \sum_i \underline{c}_i < 0 \text{ and } \sum_i \overline{c}_i \leq 0 \Rightarrow \gamma < \frac{\underline{b} - \underline{a} - \underline{d}}{\sum_i \underline{c}_i}, \tag{3.102}$$

$$4^\circ \ \sum_i \underline{c}_i < 0 \text{ and } \sum_i \overline{c}_i > 0 \Rightarrow \gamma < \min\left\{ \frac{\underline{b} - \underline{a} - \underline{d}}{\sum_i \underline{c}_i}, \frac{\overline{b} - \overline{a} - \overline{d}}{\sum_i \overline{c}_i} \right\},$$

from which we can determine the number γ.

In an N-dimensional case the relations (3.101) are of the form

$$\left[\underline{Y}_{s0}, \overline{Y}_{s0}\right] + \eta_0 \sum_{i=1}^{m} \left[\underline{w}_i, \overline{w}_i\right]\left[\underline{F}_s, \overline{F}_s\right] + \left(\left[\underline{\Psi}_s, \overline{\Psi}_s\right] + \left[-\alpha_s, \alpha_s\right]\right)\left[\underline{h}_0, \overline{h}_0\right]^p \subset$$
$$\subset \left[\underline{\Delta}_s, \overline{\Delta}_s\right].$$

Hence,

$$\left[\underline{Y}_{s0}, \overline{Y}_{s0}\right] +$$
$$+ \eta_0 \sum_{i=1}^{m} \left[\min\left\{\underline{w}_i \underline{F}_s, \underline{w}_i \overline{F}_s, \overline{w}_i \underline{F}_s, \overline{w}_i \overline{F}_s\right\}, \max\left\{\underline{w}_i \underline{F}_s, \underline{w}_i \overline{F}_s, \overline{w}_i \underline{F}_s, \overline{w}_i \overline{F}_s\right\}\right] + \tag{3.103}$$
$$+ \left[\underline{\Psi}_s - \alpha_s, \overline{\Psi}_s + \alpha_s\right]\left[\underline{h}_0, \overline{h}_0\right]^p \subset \left[\underline{\Delta}_{y_s}, \overline{\Delta}_{y_s}\right].$$

Since $0 < \underline{h}_0 \leq \overline{h}_0$, then

$$\left[\underline{h}_0, \overline{h}_0\right]^p = \left[\underline{h}_0^p, \overline{h}_0^p\right].$$

Thus,

$$\left[\underline{\Psi}_s - \alpha_s, \overline{\Psi}_s + \alpha_s\right]\left[\underline{h}_0, \overline{h}_0\right]^p = \left[\underline{\Psi}_s - \alpha_s, \overline{\Psi}_s + \alpha_s\right]\left[\underline{h}_0^p, \overline{h}_0^p\right] =$$

$$= \left[\left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p, \left(\overline{\Psi}_s + \alpha_s\underline{h}_0^p\right)\right].$$

Taking into account this equality, we can write the relation (3.103) in the form

$$\left[\underline{Y}_{s0} + \eta_0 \sum_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} + \left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p,\right.$$

$$\left.\overline{Y}_{s0} + \eta_0 \sum_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} + \left(\overline{\Psi}_s + \alpha_s\right)\overline{h}_0^p\right] \subset \left[\underline{\Delta}_{y_s}, \overline{\Delta}_{y_s}\right].$$

From the above inclusion it follows that two inequalities should be fulfilled:

$$\eta_0 \sum_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} > \underline{\Delta}_{y_s} - \underline{Y}_{s0} - \left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p,$$

$$\eta_0 \sum_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} < \overline{\Delta}_{y_s} - \overline{Y}_{s0} - \left(\overline{\Psi}_s + \alpha_s\right)\overline{h}_0^p.$$

Thus, from the previous considerations (see (3.102)) we have

0° $\underline{\Delta}_{y_s} - \underline{Y}_{s0} - \left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p \le 0$ or $\overline{\Delta}_{y_s} - \overline{Y}_{s0} - \left(\overline{\Psi}_s + \alpha_s\right)\overline{h}_0^p \le 0 \Rightarrow$

$\Rightarrow$ the evaluation of η_0 is not possible (the initial step size h_0 should be decreased),

1° $\displaystyle\sum_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} =$

$= \displaystyle\sum_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} \Rightarrow \eta_0$ can be an arbitrary number,

2° $\displaystyle\sum_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} \ge 0$

and $\displaystyle\sum_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} > 0 \Rightarrow$

$$\Rightarrow \ \eta_0 < \frac{\overline{\Delta}_{y_s} - \overline{Y}_{s0} - \left(\overline{\Psi}_s + \alpha_s\right)\overline{h}_0^p}{\sum\limits_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\}},$$

3° $\sum\limits_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} < 0$

and $\sum\limits_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} \leq 0 \Rightarrow$

$$\Rightarrow \ \eta_0 < \frac{\underline{\Delta}_{y_s} - \underline{Y}_{s0} - \left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p}{\sum\limits_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\}},$$

4° $\sum\limits_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} < 0$

and $\sum\limits_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\} > 0 \Rightarrow$

$$\Rightarrow \ \eta_0 < \min\left\{\begin{array}{l} \dfrac{\overline{\Delta}_{y_s} - \overline{Y}_{s0} - \left(\overline{\Psi}_s + \alpha_s\right)\overline{h}_0^p}{\sum\limits_{i=1}^{m} \max\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\}}, \\[2em] \dfrac{\underline{\Delta}_{y_s} - \underline{Y}_{s0} - \left(\underline{\Psi}_s - \alpha_s\right)\underline{h}_0^p}{\sum\limits_{i=1}^{m} \min\left\{\underline{w}_i\underline{F}_s, \underline{w}_i\overline{F}_s, \overline{w}_i\underline{F}_s, \overline{w}_i\overline{F}_s\right\}} \end{array}\right\}.$$

From the above relations we should determine the number η_0 for each s = 1, 2, ... , N, and then take the smallest of them.

3.5. Possible Minimizations of the Widths of Interval Solutions

The constants Q, R and S, occurring in estimations of the widths of interval solutions in the theorems 3.2 and 3.4 – 3.7, depend on the coefficients w_i, c_i and a_{ij}, where $i = 1, 2, \dots, m$, $j = 1, 2, \dots, i-1$ for explicit methods and $j = 1, 2, \dots, m$ for implicit ones. Taking into account (3.4) and (3.9) we can consider these constants as depending only on w_i and a_{ij}, i.e.

$$Q = Q(w., a..), \quad R = R(w., a..), \quad S = S(w., a..).$$

It seems to be interesting to study the existence of such a system of coefficients w_i and a_{ij} that the constants Q, R and S will be minimal. As a consequence of the existence of such a system we would get minimal estimations of the widths of interval solutions.

According to the proof of Theorem 3.2 (see Sect. 3.2.3) for explicit methods we have

$$\begin{gathered} Q = [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]\frac{\exp(\gamma_m \xi_m^* \Lambda) - 1}{\gamma_m \Lambda} = [w(\Psi(\Delta_t, \Delta_y)) + 2\alpha]\overline{Q}, \\ R = \exp(\gamma_m \xi_m^* \Lambda), \quad S = R - 1, \end{gathered} \tag{3.104}$$

where

$$\gamma_m = \gamma_m(w., a..) = \sum_{i=1}^{m} |w_i| \sum_{j=0}^{i-1} \mu_{ij}(h_0 \Lambda)^j,$$

μ_{ij} are some constants depending on a_{ij}, and ξ_m^* is given by (3.36).

The problem of minimizing the function $\psi(t_k, y(t_k))$ (with respect to coefficients w_i and a_{ij}), occurring in (3.10), and hence also $w(\Psi(\Delta_t, \Delta_y))$, is solved in classical theory of the Runga-Kutta method, and therefore we will not consider this problem. Moreover, from the last equation in (3.104) it follows that if R will be minimized, then the same will happen to S. Thus, we will restrict our considerations to the terms

$$\overline{Q} = \overline{Q}(w., a..) \text{ and } R = R(w., a..).$$

Theorem 3.8. *There do not exist coefficients w_i and a_{ij}* ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, i-1$) *that minimize the constants $\overline{Q}$ and R given by* (3.104).

Proof. The necessary conditions for existing an extremum of $R = R(w., a..)$ are as follows:

$$\xi_m^* \frac{\partial \gamma_m}{\partial w_i} + \gamma_0 \frac{\partial \xi_m^*}{\partial w_i} = 0,$$

$$\xi_m^* \frac{\partial \gamma_m}{\partial a_{ij}} + \gamma_0 \frac{\partial \xi_m^*}{\partial a_{ij}} = 0,$$

$$i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, i-1.$$

For $\overline{Q} = \overline{Q}(w., a..)$ the necessary conditions are the following:

$$\left(\gamma_m \xi_m^* \Lambda - 1 + \frac{1}{\exp(\gamma_m \xi_m^* \Lambda)} \right) \frac{\partial \gamma_m}{\partial w_i} + \gamma_m^2 \Lambda \frac{\partial \xi_m^*}{\partial w_i} = 0,$$

$$\left(\gamma_m \xi_m^* \Lambda - 1 + \frac{1}{\exp(\gamma_m \xi_m^* \Lambda)} \right) \frac{\partial \gamma_m}{\partial a_{ij}} + \gamma_m^2 \Lambda \frac{\partial \xi_m^*}{\partial a_{ij}} = 0,$$

$$i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, i-1.$$

We will prove the theorem for two-stage methods, i.e. for $m = 2$. In the case $m > 2$ the idea of the proof is the same, but calculations are more complicated. We have

$$\gamma_2 = |w_1| + |w_2| \left(1 + |a_{21}| h_0 \Lambda\right), \quad \xi_2^* = \min\{\xi_0, \xi_2\}, \tag{3.105}$$

where ξ_0 and ξ_2 fulfill the conditions

$$Y_0 + \xi_0 \left(w_1 F(\Delta_t, \Delta_y) + w_2 F(\Delta_t, \Delta_y) \right) + \left(\Psi(\Delta_t, \Delta_y) + [-\alpha, \alpha] \right) h_0^2 \subset \Delta_y,$$

$$Y_0 + \xi_2 a_{21} F(\Delta_t, \Delta_y) \subset \Delta_y.$$

The function R has an extremum if

$$\begin{gathered} \xi_2^* \frac{\partial \gamma_2}{\partial w_i} + \gamma_2 \frac{\partial \xi_m^*}{\partial w_i} = 0, \quad i = 1, 2, \\ \xi_2^* \frac{\partial \gamma_2}{\partial a_{21}} + \gamma_2 \frac{\partial \xi_m^*}{\partial a_{21}} = 0. \end{gathered} \tag{3.106}$$

From (3.105) it follows that

$$\frac{\partial \gamma_2}{\partial w_1} = \pm 1, \quad \frac{\partial \gamma_2}{\partial w_2} = \pm\left(1 + |a_{21}| h_0 \Lambda\right), \quad \frac{\partial \gamma_2}{\partial a_{21}} = \pm |w_2| h_0 \Lambda.$$

If $\xi_0 \le \xi_2$, then $\xi_2^* = \xi_0$ and $\partial \xi_2^* / \partial a_{21} = 0$. Then, from the second equation (3.106) we obtain $\xi_0 |w_2| h_0 \Lambda = 0$. Since $\xi_0 > 0$, then $w_2 = 0$. This fact is in contradiction to $w_2c_2 = 1/2$, which follows from the equations determining the coefficients of two-stage methods. If $\xi_2 < \xi_0$, then $\xi_2^* = \xi_2$ and $\partial\xi_2^* / \partial w_i$ $(i = 1, 2)$. In this case, from the first equation (3.106) we have

$$\xi_2 = 0 \quad (i = 1),$$
$$\xi_2\left(1 + |a_{21}|h_0\Lambda\right) = 0 \quad (i = 2),$$

which contradicts the condition $\xi_2 > 0$.

Next, the function $\overline{Q}$ has an extremum if

$$\begin{aligned}
&\left(\gamma_2\xi_2^*\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_2^*\Lambda)}\right)\frac{\partial\gamma_2}{\partial w_i} + \gamma_2^2\Lambda\frac{\partial\xi_2^*}{\partial w_i} = 0, \quad i = 1, 2,\\
&\left(\gamma_2\xi_2^*\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_2^*\Lambda)}\right)\frac{\partial\gamma_2}{\partial a_{12}} + \gamma_2^2\Lambda\frac{\partial\xi_2^*}{\partial a_{12}} = 0.
\end{aligned} \tag{3.107}$$

If $\xi_0 \le \xi_2$, then $\xi_2^* = \xi_0$ and $\partial\xi_2^* / \partial a_{21} = 0$. But the second equation (3.107) yields

$$\left(\gamma_2\xi_0\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_0\Lambda)}\right)|w_2|h_0\Lambda = 0.$$

Since $w_2 \neq 0$, we have

$$\gamma_2\xi_0\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_0\Lambda) = 0}.$$

This equation may by fulfilled only if $\gamma_2\xi_0\Lambda = 0$, which is impossible taking into account that $\xi_0 > 0$, $\gamma_2 > 0$ and $\Lambda > 0$.

If $\xi_2 < \xi_0$, then $\xi_2^* = \xi_2$ and $\partial\xi_2^* / \partial w_i = 0$ $(i = 1, 2)$. In this case, from the first equation (3.107) it follows that

$$\gamma_2\xi_2\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_2\Lambda)} = 0 \quad (i = 1),$$
$$\left(\gamma_2\xi_2\Lambda - 1 + \frac{1}{\exp(\gamma_2\xi_2\Lambda)}\right)\left(1 + |a_{21}|h_0\Lambda\right) = 0 \quad (i = 2).$$

These equations can be fulfilled only if $\gamma_2\xi_2\Lambda = 0$, which is impossible. ■

In the case of implicit methods the constants *Q*, *R* and *S* are of the following forms:

$$Q = [w(\varPsi(\varDelta_t, \varDelta_y) + 2\alpha]\frac{\exp(v_m\overset{*}{\eta}_m\varLambda) - 1}{v_m\varLambda} = [w(\varPsi(\varDelta_t, \varDelta_y) + 2\alpha]\overline{Q},$$
$$R = \exp(v_m\overset{*}{\eta}_m\varLambda), \quad S = R - 1, \tag{3.108}$$

where $v_m = v_m(w., a..)$ $(m = 1, 2, 3, 4)$ are constants defined in the proofs of theorems 3.4 – 3.7, and $\overset{*}{\eta}_m$ is given by (3.51).

Theorem 3.9. *There do not exist coefficients w_i and a_{ij}* ($i = 1, 2, \dots, m; j = 1, 2, \dots, m$) *that minimize the constants $\overline{Q}$ and R given by* (3.108).

Proof. The proof resembles the previous one. We limit our considerations to the case $m = 2$ (for $m > 2$ only the calculations are more complicated, but the idea is the same).

First, let us consider the function $R = R(w., a..)$. The necessary conditions for an extremum of R to exist are as follows:

$$\begin{aligned} &\overset{*}{\eta}_2\frac{\partial v_2}{\partial w_i} + v_2\frac{\partial \overset{*}{\eta}_2}{\partial w_i} = 0, \quad i = 1, 2, \\ &\overset{*}{\eta}_2\frac{\partial v_2}{\partial a_{ij}} + v_2\frac{\partial \overset{*}{\eta}_2}{\partial a_{ij}} = 0, \quad i, j = 1, 2. \end{aligned} \tag{3.109}$$

Let us denote

$$\alpha = |w_1|(1 + h_0\varLambda|a_{12}|) + |w_2|(1 + h_0\varLambda|a_{21}|),$$
$$\beta = (1 - h_0\varLambda|a_{11}|)(1 - h_0\varLambda|a_{22}|) - h_0^2\varLambda^2|a_{12}||a_{21}|.$$

Then (see the proof of the theorem 3.5) $v_2 = \alpha / \beta$, and

$$\begin{aligned} &\frac{\partial v_2}{\partial w_1} = \pm\frac{1 + h_0\varLambda|a_{12}|}{\beta}, \quad \frac{\partial v_2}{\partial w_2} = \pm\frac{1 + h_0\varLambda|a_{21}|}{\beta}, \\ &\frac{\partial v_2}{\partial a_{11}} = \pm\frac{\alpha h_0\varLambda}{\beta}(1 - h_0\varLambda|a_{22}|), \quad \frac{\partial v_2}{\partial a_{22}} = \pm\frac{\alpha h_0\varLambda}{\beta}(1 - h_0\varLambda|a_{11}|), \\ &\frac{\partial v_2}{\partial a_{12}} = \pm\frac{h_0\varLambda}{\beta^2}(|w_1|\beta - \alpha h_0\varLambda|a_{21}|), \quad \frac{\partial v_2}{\partial a_{21}} = \pm\frac{h_0\varLambda}{\beta^2}(|w_2|\beta - \alpha h_0\varLambda|a_{12}|). \end{aligned} \tag{3.110}$$

From (3.52) and (3.53) it follows that if $\overset{*}{\eta}_2 = \eta_1$ or $\overset{*}{\eta}_2 = \eta_2$, then $\partial\overset{*}{\eta}_2 / \partial w_i = 0$ $(i = 1,2)$, and if $\overset{*}{\eta}_2 = \eta_0$, then $\partial\overset{*}{\eta}_2 / \partial a_{ij} = 0$ $(i, j = 1, 2)$. Thus, for $\overset{*}{\eta}_2 = \eta_1$ or $\overset{*}{\eta}_2 = \eta_2$ the necessary conditions have the following forms:

$$\eta_2^* \frac{\partial v_2}{\partial w_i} = 0, \quad i = 1, 2,$$
$$\eta_2^* \frac{\partial v_2}{\partial a_{ij}} + v_2 \frac{\partial \eta_2^*}{\partial a_{ij}} = 0, \quad i, j = 1, 2.$$

Since $\eta_2^* > 0,$ then from the first of the above equations it follows that for $i = 1, 2$ we have $\partial v_2 / \partial w_i = 0,$ which is impossible according to the formulas (3.110).

If $\eta_2^* = \eta_0,$ the conditions are as follows:

$$\eta_2^* \frac{\partial v_2}{\partial w_i} + v_2 \frac{\partial \eta_2^*}{\partial w_i} = 0, \quad i = 1, 2,$$
$$\eta_2^* \frac{\partial v_2}{\partial a_{ij}} = 0, \quad i, j = 1, 2.$$

Since $\eta_2^* = \eta_0 > 0,$ then from the second equation it follows that $\partial v_2 / \partial a_{ij} = 0$ for $i, j = 1, 2$. On the basis of the formulas (3.110) we have

$$\begin{aligned} \frac{\partial v_2}{\partial a_{11}} = 0 \;\Leftrightarrow\; 1 - h_0 \varLambda |a_{22}| = 0 \;\Leftrightarrow\; |a_{22}| = \frac{1}{h_0 \varLambda}, \\ \frac{\partial v_2}{\partial a_{22}} = 0 \;\Leftrightarrow\; 1 - h_0 \varLambda |a_{11}| = 0 \;\Leftrightarrow\; |a_{11}| = \frac{1}{h_0 \varLambda} \end{aligned} \tag{3.111}$$

and

$$\frac{\partial v_2}{\partial a_{12}} = 0 \;\Leftrightarrow\; |w_1| \beta + \alpha h_0 \varLambda |a_{21}| = 0.$$

Taking into account what β means, from the last equation we obtain

$$|w_1| \Big[\big(1 - h_0 \varLambda |a_{11}|\big)\big(1 - h_0 \varLambda |a_{22}|\big) - h_0^2 \varLambda^2 |a_{12}| |a_{21}| \Big] + \alpha h_0 \varLambda |a_{21}| = 0.$$

Hence, according to (3.111), we have

$$- h_0 \varLambda |w_1| |a_{12}| |a_{21}| + \alpha |a_{21}| = 0,$$

and taking into account what α means, we get

$$- h_0 \varLambda |w_1| |a_{12}| |a_{21}| + \Big[|w_1| \big(1 + h_0 \varLambda |a_{12}|\big) + |w_2| \big(1 + h_0 \varLambda |a_{21}|\big) \Big] |a_{21}| = 0.$$

From this equation it follows that either $|a_{21}| = 0$ or

$$|w_1| + |w_2| + |w_2| h_0 \varLambda |a_{21}| = 0.$$

The last equation contradicts $w_1 + w_2 = 1.$ Therefore,

$$\left| a_{21} \right| = 0. \tag{3.112}$$

If we consider the equation

$$\frac{\partial v}{\partial a_{21}} = 0 \quad \Leftrightarrow \quad \left| w_2 \left| \beta + \alpha h_0 \Lambda \right| a_{12} \right| = 0,$$

then, proceeding in a similar fashion, we get

$$\left| a_{12} \right| = 0. \tag{3.113}$$

But if conditions (3.111) – (3.113) are fulfill, then $\beta = 0$, i.e. the denominator in the formula determining v_2 is equal to zero, what is, of course, impossible. Thus, the function $\underline{R}$ (and therefore also the function $\underline{S}$) does not have an extremum.

For $\overline{Q} = \overline{Q}(w., a..)$ the necessary conditions for an extremum to exist are as follows:

$$\left(v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} \right) \frac{\partial v_2}{\partial w_i} + v_2^2 \Lambda \frac{\partial \overset{*}{\eta_2}}{\partial w_i} = 0, \quad i = 1, 2,$$

$$\left(v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} \right) \frac{\partial v_2}{\partial a_{ij}} + v_2^2 \Lambda \frac{\partial \overset{*}{\eta_2}}{\partial a_{ij}} = 0, \quad i, j = 1, 2.$$

As previously, we consider two cases: $\overset{*}{\eta_2} = \eta_1$ or $\overset{*}{\eta_2} = \eta_2$ and $\overset{*}{\eta_2} = \eta_0$. If $\overset{*}{\eta_2} = \eta_1$ or $\overset{*}{\eta_2} = \eta_2$, then $\partial \overset{*}{\eta_2} / \partial w_i = 0$ $(i = 1, 2)$ and the conditions are of the form

$$\left(v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} \right) \frac{\partial v_2}{\partial w_i} = 0, \quad i = 1, 2,$$

$$\left(v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} \right) \frac{\partial v_2}{\partial a_{ij}} + v_2^2 \Lambda \frac{\partial \overset{*}{\eta_2}}{\partial a_{ij}} = 0, \quad i = 1, 2.$$

Since

$$v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} > 0,$$

then from the first equation it follows that $\partial v_2 / \partial w_i = 0$ for $i = 1, 2$, which is impossible according to (3.112).

If $\overset{*}{\eta_2} = \eta_0$, then $\partial \overset{*}{\eta_2} / \partial a_{ij} = 0$ $(i, j = 1, 2)$ and the necessary conditions are as follows:

$$\left(v_2 \overset{*}{\eta_2} \Lambda - 1 + \frac{1}{\exp(v_2 \overset{*}{\eta_2} \Lambda)} \right) \frac{\partial v_2}{\partial w_i} + v_2^2 \Lambda \frac{\partial \overset{*}{\eta_2}}{\partial w_i} = 0, \quad i = 1, 2,$$

$$\left(\nu_2\eta_2^*\Lambda - 1 + \frac{1}{\exp(\nu_2\eta_2^*\Lambda)}\right)\frac{\partial\nu_2}{\partial a_{ij}} = 0, \quad i, j = 1, 2.$$

In this case from the second equation we have $\partial\nu_2 / \partial a_{ij} = 0,$ from which follow contradictory values of a_{ij}. Thus, the function $\overline{Q} = \overline{Q}(w.., a..)$ does not have an extremum. ■

3.6. Computational Complexity of Interval Methods of Runge-Kutta Type

In Section 3.4 we have presented an algorithm for finding (in floating-point interval arithmetic) the step size h in explicit and implicit algorithms. From an analysis of this algorithm it follows that in order to find each of the numbers $\eta_2 = \xi_2 > 0, \ldots, \eta_m = \xi_m > 0$ we need to perform at most $8N(m - 1)$ multiplications, $2N(m - 1)$ divisions, $2N$ subtractions and calculate the function $F(T, Y)$, where N denotes the number of equations in the initial value problem. Next, to find $\eta_0 = \xi_0 > 0$ we have to perform at most $9N + p - 1$ multiplications, $2N$ divisions, $(2m - 1)N$ additions, $5N$ subtractions and calculate $F(T, Y)$ and $\Psi(T, Y)$, where p denotes the order of the method used, and where $2N$ subtractions and the evaluation of $F(T, Y)$ are the same operations as previously. To recapitulate, finding all the numbers $\eta_0 = \xi_0 > 0,\ \eta_2 = \xi_2 > 0, \ldots, \eta_m = \xi_m > 0$ needs at most $12Nm + 7N + p - 1$ operations, one evaluation of $F(T, Y)$ and one evaluation of $\Psi(T, Y)$.

Since the step size h is the quotient (in floating-point interval arithmetic) of the smallest number of $\xi_0, \xi_2, \ldots, \xi_m$ and the number n, then finally we conclude that in order to find h we have to perform at most

$$l_p(h) = N(12m + 7) + p + 7 + 8\big(l(f) + l_p(\psi)\big)$$

operations, where p denotes the order of the method, N – the number of equations in the initial value problem, m – the number of stages, $l(f)$ – the number of operations needed to evaluate $f(t, y)$, and $l_p(\psi)$ denotes the number of operations needed to evaluate $\psi(t, y)$. It should be added that in any interval method the step size h is calculated only once.

In order to find the number of operations in the $(k + 1)$-th step of the method $(k = 0, 1, \ldots, n - 1)$, let us first consider the K_{ik} terms given by (3.33). The steps for determining the number of operations to find these terms is presented in Figure 3.1.

From Figure 3.1 it follows that to determine all the K_{ik} terms we should perform at most

$$l(K) = 8m \cdot l(f) + 5(m - 1)[5N(m + 2) + 2]$$

operations. Further operations are presented in Figure 3.2.

From the previous considerations it is obvious that the total number needed to find Y_{k+1} in n steps of the explicit interval method of Runge-Kutta type is equal to

$$\begin{aligned} l(Y) &= nN(10m+21) + 8nl_p(\psi) + 2p + nl(K) + l_p(h) = \\ &= n[N(25m^2+35m-29)+10(m-1)] + 8n[m\cdot l(f) + l_p(\psi)] + l_p(h). \end{aligned}$$

In n steps of conventional m-stage explicit Runge-Kutta method the number of operation is equal to

$$l(y) = n[N(m^2+2)+2(m-1)] + nm\cdot l(f).$$

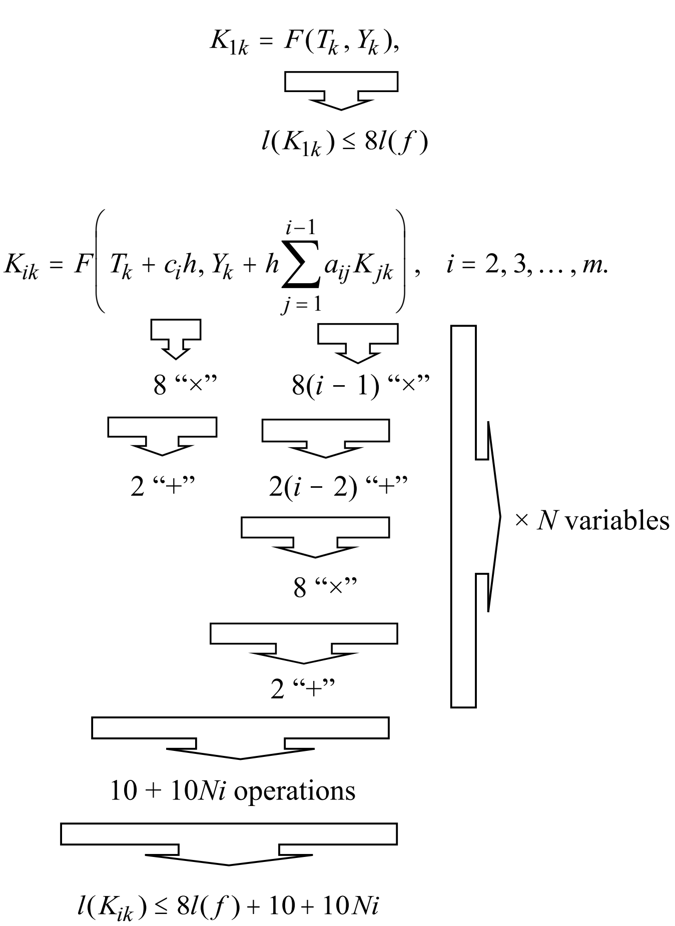

$K_{1k} = F(T_k, Y_k),$
$l(K_{1k}) \le 8l(f)$
$K_{ik} = F\left(T_k + c_i h, Y_k + h\sum_{j=1}^{i-1} a_{ij}K_{jk}\right), \quad i = 2, 3, \ldots, m.$
8 “×”
8(i – 1) “×”
2 “+”
2(i – 2) “+”
8 “×”
2 “+”
× N variables
10 + 10Ni operations
$l(K_{ik}) \le 8l(f) + 10 + 10Ni$

Figure 3.1. Determining the number of operations for calculating the K_{ik} terms

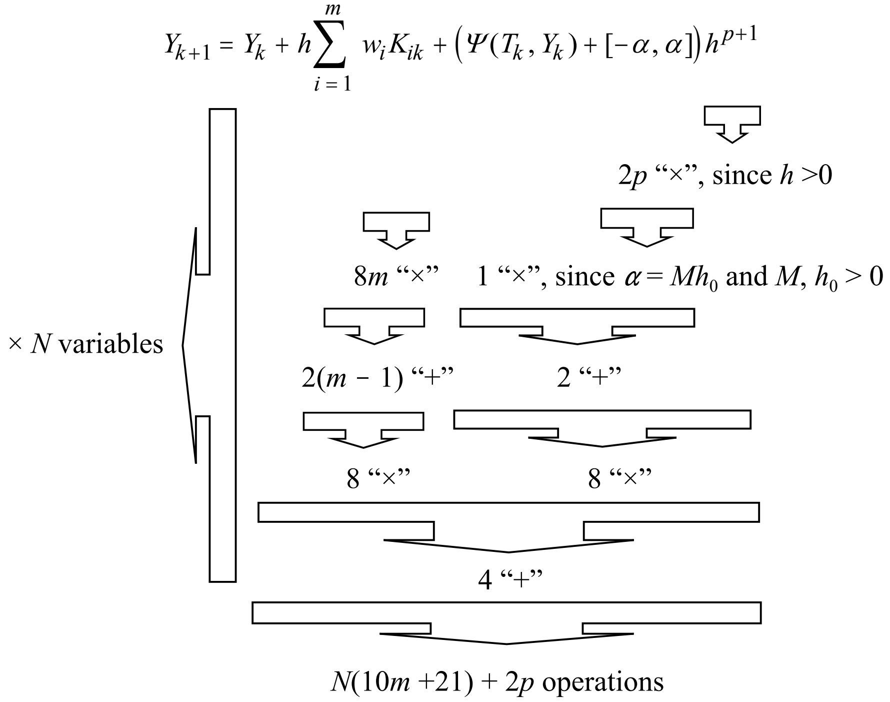

$Y_{k+1} = Y_k + h\sum_{i=1}^{m} w_i K_{ik} + \left(\Psi(T_k, Y_k) + [-\alpha, \alpha]\right)h^{p+1}$
2p "×", since h >0
8m "×"
1 "×", since α = Mh0 and M, h0 > 0
× N variables
2(m − 1) "+"
2 "+"
8 "×"
8 "×"
4 "+"
N(10m +21) + 2p operations

Figure 3.2. Finding the number of operations for calculating Y_{k+1} in the interval explicit method of Runge-Kutta type

Example 3.1

Let us consider a simple initial value problem

$$y' = \lambda y, \quad y(0) = 1.$$

We have: $N = 1$, $l(f)=1$, $l_1(\psi) = 3$, $l_2(\psi) = 22$, $l_3(\psi) = 82$, $l_4(\psi) = 291$, $p = m$. In Figure 3.3 we present the ratios of the maximum numbers of operations in explicit interval methods of Runge-Kutta type to the number of operations in conventional Runge-Kutta methods for $p = m = 1$, 2, 3 and 4. We see that this ratio increases together with the order of method. The values of $l_p(\psi)$ have the decisive influence on it [126]. ■

In implicit interval methods of Runge-Kutta type the step size h is calculated in a similar way and also only once. To find this step size we have to perform at most

$$l_p^*(h) = N(12m + 17) + p + 7 + 8\left(l(f) + l_p^*(\psi)\right)$$

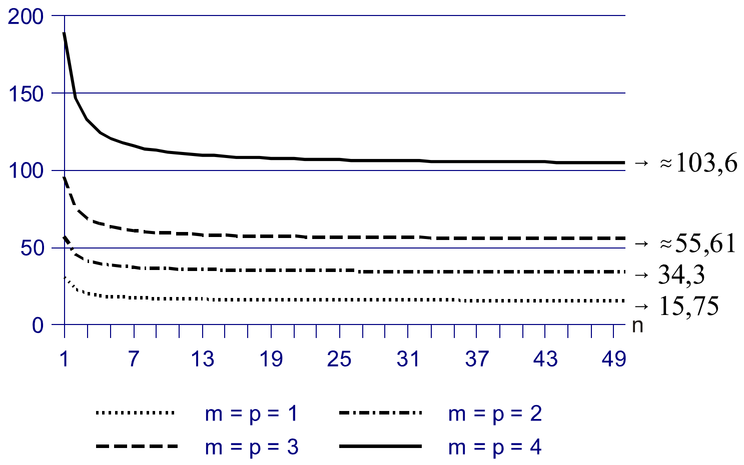


Figure 3.3. The ratios *l*(*Y*)/*l*(*y*) in explicit Runge-Kutta methods

operations ($10N$ more than in explicit methods). Moreover, in implicit methods in each step we have to solve a nonlinear system of equations with respect to K_{ik} by an iteration (see (3.55) and (3.56)). The first iteration requires at most $8N(m+1)+$ $+\,8$ multiplications, $2Nm+2$ additions and two evaluations of $F(T, Y)$. In each subsequent iteration we have to do at most $8N(m+1)$ multiplications, $2Nm$ additions and evaluate once the value of $F(T, Y)$. Thus, in s iterations for all $i = 1, 2, \ldots, m$ we have at most

$$l_s(K) = m[sN(10m+8) + 8(s+1)l(f) + 10]$$

operations. If the values of K_{ik} are known, the number of operations to calculate Y_{k+1} is the same as in explicit methods. Thus, to find all Y_{k+1} ($k = 0, 1, \ldots, n-1$) on the basis of (3.49) – (3.50) we have to perform at most [126]

$$\begin{aligned} l^*(Y) &= nN(10m+21) + 8nl_p^*(\psi) + 2p + nl_S(K) + l_p^*(h) = \\ &= n\{N[Sm(10m+8) + 10m + 21] + 10m\} + \\ &\quad + 8n[m(S+1)l(f) + l_p^*(\psi)] + 2p + l_p^*(h) \end{aligned}$$

operations, where S denotes the maximum number of iterations in all steps of the method. In conventional implicit Runge-Kutta methods the total number of operations is not greater than

$$l^*(y) = n\{N[Sm(2m+1) + 2m + 1] + 2m\} + nm(S+1)l(f).$$

3.7. Numerical Examples and a Comparison of Interval Methods of Runge-Kutta Type

Below we present some numerical experiments that confirm the theoretical justifications given in the previous sections.

Example 3.2

Let us solve, by a number of interval Runge-Kutta type methods, the initial value problem (2.2) with $\lambda = 0.5$ and consider the interval explicit methods (3.39), (3.41), (3.42), and implicit ones (3.57), (3.58), (3.59) and (3.61). Let us assume that in each of these methods the input data are as follows:

$$\Delta_t = [0, 10], \quad \Delta_y = \left[\underline{0.9}, 149\right], \quad h_0 = 0.001, \quad T_0 = [0, 0], \quad Y_0 = [1, 1],$$

where $\underline{x}$ denotes the largest machine number less or equal to x (similarly, further $\overline{x}$ will denote the smallest machine number greater or equal to x). Taking the appropriate values of M (see (3.34)), on the basis of the theory presented in Section 3.4 we can find the maximum integration intervals as shown in Table 3.2. From this table it follows that (regardless of the methods considered) the maximum integration interval is approximately equal to 1.985 (and is a little bit larger for the higher order methods). Thus, we have applied all of the methods mentioned for $t_{\max} = 1$, and partitioned the interval [0, 1] into $n = 2000$ parts. This yields $h = 0.0005$.

Table 3.2. The maximum integration intervals for selected interval methods of Runge-Kutta type and the problem (2.2)

Kind of method	*Method*	*Method order*	*M*	$t_{\max}$
Explicit	(3.39)	1	0.3	1.9863271771812081
	(3.41)	2	0.07	1.9865769313329530
	(3.42)	4	0.003	1.9865771812080529
Implicit	(3.57)	2	0.07	1.9865771709792058
	(3.58)	4	0.003	1.9865771812080535
	(3.59)	3	0.014	1.9865771811953255
	(3.61)	4	0.003	1.9865771812080536

The obtained interval results are presented in Tables 3.3 – 3.9. In implicit interval methods considered we assumed the accuracy 10^{-18} in both iterations (3.55) and (3.56), and we have obtained the same results (in each method the number of iterations was equal to 5 or 6).

Table 3.3. The interval solution of the problem (2.2) obtained by the method (3.39)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710804491833E+0000, 1.0512711112079961E+0000]	$\approx 3.08 \cdot 10^{-8}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051708853773120E+0000, 1.1051709484719755E+0000]	$\approx 6.31 \cdot 10^{-8}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618341923690531E+0001, 1.1618342894574618E+0001]	$\approx 9.71 \cdot 10^{-8}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214026892035326E+0000, 1.2214028220285831E+0000]	$\approx 1.33 \cdot 10^{-7}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840253281475776E+0000, 1.2840255985415268E+0000]	$\approx 1.70 \cdot 10^{-7}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498586984139620E+0000, 1.3498589083030083E+0000]	$\approx 2.10 \cdot 10^{-7}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190674177159933E+0000, 1.4190676691250936E+0000]	$\approx 2.51 \cdot 10^{-7}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918245438975329E+0000, 1.4918248389554660E+0000]	$\approx 2.95 \cdot 10^{-7}$
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683120076677390E+0000, 1.5683123486124284E+0000]	$\approx 3.41 \cdot 10^{-7}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487210675225932E+0000, 1.6487214567067031E+0000]	$\approx 3.89 \cdot 10^{-7}$

Table 3.4. The interval solution of the problem (2.2) obtained by the method (3.41)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963741955E+0000, 1.0512710963777842E+0000]	$\approx 3.59 \cdot 10^{-12}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180718951E+0000, 1.1051709180792563E+0000]	$\approx 7.36 \cdot 10^{-12}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427225061E+0001, 1.1618342427338333E+0001]	$\approx 1.13 \cdot 10^{-11}$

Table 3.4. (cont.)

T	*Y*	*Width of Y*
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581522626E+0000, 1.2214027581677591E+0000]	$\approx 1.55\cdot 10^{-11}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166775928E+0000, 1.2840254166974723E+0000]	$\approx 1.99\cdot 10^{-11}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075634959E+0000, 1.3498588075879832E+0000]	$\approx 2.45\cdot 10^{-11}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485782683E+0000, 1.4190675486075996E+0000]	$\approx 2.93\cdot 10^{-11}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976236700E+0000, 1.4918246976580938E+0000]	$\approx 3.44\cdot 10^{-11}$
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854698208E+0000, 1.5683121855095981E+0000]	$\approx 3.98\cdot 10^{-11}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212706768890E+0000, 1.6487212707222942E+0000]	$\approx 4.54\cdot 10^{-11}$

Table 3.5. The interval solution of the problem (2.2) obtained by the method (3.42)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963760240E+0000, 1.0512710963760241E+0000]	$\approx 2.18\cdot 10^{-17}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180756476E+0000, 1.1051709180756477E+0000]	$\approx 4.53\cdot 10^{-17}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427282830E+0001, 1.1618342427282832E+0001]	$\approx 7.04\cdot 10^{-17}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581601697E+0000, 1.2214027581601699E+0000]	$\approx 9.65\cdot 10^{-17}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166877414E+0000, 1.2840254166877416E+0000]	$\approx 1.24\cdot 10^{-16}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075760030E+0000, 1.3498588075760032E+0000]	$\approx 1.51\cdot 10^{-16}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485932571E+0000, 1.4190675485932574E+0000]	$\approx 1.79\cdot 10^{-16}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976412702E+0000, 1.4918246976412705E+0000]	$\approx 2.11\cdot 10^{-16}$

Table 3.5. (cont.)

T	*Y*	*Width of Y*
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854901686E+0000, 1.5683121854901690E+0000]	$\approx 2.43 \cdot 10^{-16}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212707001280E+0000, 1.6487212707001283E+0000]	$\approx 2.78 \cdot 10^{-16}$

Table 3.6. The interval solution of the problem (2.2) obtained by the method (3.57)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963742981E+0000, 1.0512710963778868E+0000]	$\approx 3.59 \cdot 10^{-12}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180721109E+0000, 1.1051709180794721E+0000]	$\approx 7.36 \cdot 10^{-12}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427228464E+0001, 1.1618342427341736E+0001]	$\approx 1.13 \cdot 10^{-11}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581527396E+0000, 1.2214027581682362E+0000]	$\approx 1.55 \cdot 10^{-11}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166782196E+0000, 1.2840254166980992E+0000]	$\approx 1.99 \cdot 10^{-11}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075642866E+0000, 1.3498588075887741E+0000]	$\approx 2.45 \cdot 10^{-11}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485792381E+0000, 1.4190675486085697E+0000]	$\approx 2.93 \cdot 10^{-11}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976248353E+0000, 1.4918246976592592E+0000]	$\approx 3.44 \cdot 10^{-11}$
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854711989E+0000, 1.5683121855109764E+0000]	$\approx 3.98 \cdot 10^{-11}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212706784987E+0000, 1.6487212707239042E+0000]	$\approx 4.54 \cdot 10^{-11}$

Table 3.7. The interval solution of the problem (2.2) obtained by the method (3.58)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963760240E+0000, 1.0512710963760241E+0000]	$\approx 4.41 \cdot 10^{-17}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180756476E+0000, 1.1051709180756477E+0000]	$\approx 9.03 \cdot 10^{-17}$

Table 3.7. (cont.)

T	*Y*	*Width of Y*
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427282830E+0001, 1.1618342427282833E+0001]	$\approx 1.41\cdot 10^{-16}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581601697E+0000, 1.2214027581601700E+0000]	$\approx 1.92\cdot 10^{-16}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166877414E+0000, 1.2840254166877417E+0000]	$\approx 2.46\cdot 10^{-16}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075760030E+0000, 1.3498588075760034E+0000]	$\approx 3.02\cdot 10^{-16}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485932571E+0000, 1.4190675485932576E+0000]	$\approx 3.63\cdot 10^{-16}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976412702E+0000, 1.4918246976412707E+0000]	$\approx 4.25\cdot 10^{-16}$
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854901686E+0000, 1.5683121854901692E+0000]	$\approx 4.92\cdot 10^{-16}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212707001280E+0000, 1.6487212707001286E+0000]	$\approx 5.61\cdot 10^{-16}$

Table 3.8. The interval solution of the problem (2.2) obtained by the method (3.59)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963760237E+0000, 1.0512710963760242E+0000]	$\approx 4.05\cdot 10^{-16}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180756471E+0000, 1.1051709180756480E+0000]	$\approx 8.29\cdot 10^{-16}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427282823E+0001, 1.1618342427282837E+0001]	$\approx 1.27\cdot 10^{-15}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581601687E+0000, 1.2214027581601706E+0000]	$\approx 1.74\cdot 10^{-15}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166877401E+0000, 1.2840254166877424E+0000]	$\approx 2.23\cdot 10^{-15}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075760013E+0000, 1.3498588075760042E+0000]	$\approx 2.75\cdot 10^{-15}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485932551E+0000, 1.4190675485932585E+0000]	$\approx 3.30\cdot 10^{-15}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976412678E+0000, 1.4918246976412718E+0000]	$\approx 3.87\cdot 10^{-15}$

Table 3.8. (cont.)

T	*Y*	*Width of Y*
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854901659E+0000, 1.5683121854901705E+0000]	$\approx 4.47\cdot 10^{-15}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212707001249E+0000, 1.6487212707001301E+0000]	$\approx 5.10\cdot 10^{-15}$

Table 3.9. The interval solution of the problem (2.2) obtained by the method (3.61)

T	*Y*	*Width of Y*
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	[1.0512710963760240E+0000, 1.0512710963760241E+0000]	$\approx 4.41\cdot 10^{-17}$
[1.9999999999999999E−0001, 2.0000000000000001E−0001]	[1.1051709180756475E+0000, 1.1051709180756477E+0000]	$\approx 9.09\cdot 10^{-17}$
[2.9999999999999999E−0001, 3.0000000000000001E−0001]	[1.1618342427282830E+0001, 1.1618342427282832E+0001]	$\approx 1.41\cdot 10^{-16}$
[3.9999999999999999E−0001, 4.0000000000000001E−0001]	[1.2214027581601696E+0000, 1.2214027581601699E+0000]	$\approx 1.93\cdot 10^{-16}$
[4.9999999999999999E−0001, 5.0000000000000001E−0001]	[1.2840254166877413E+0000, 1.2840254416687416E+0000]	$\approx 2.47\cdot 10^{-16}$
[5.9999999999999999E−0001, 6.0000000000000001E−0001]	[1.3498588075600286E+0000, 1.3498588075600321E+0000]	$\approx 3.03\cdot 10^{-16}$
[6.9999999999999999E−0001, 7.0000000000000001E−0001]	[1.4190675485932569E+0000, 1.4190675485932574E+0000]	$\approx 3.62\cdot 10^{-16}$
[7.9999999999999999E−0001, 8.0000000000000001E−0001]	[1.4918246976241270E+0000, 1.4918246976241275E+0000]	$\approx 4.25\cdot 10^{-16}$
[8.9999999999999999E−0001, 9.0000000000000001E−0001]	[1.5683121854901684E+0000, 1.5683121854901690E+0000]	$\approx 4.91\cdot 10^{-16}$
[9.9999999999999999E−0001, 1.0000000000000001E+0000]	[1.6487212707001277E+0000, 1.6487212707001283E+0000]	$\approx 5.60\cdot 10^{-16}$

According to (3.32) and (3.49), solving the problem (2.2) by the interval methods of Runge-Kutta type we must find interval extensions of $\psi(t, y)$. But this function is expressed by $f(t, y)$ and its partial derivatives (see (3.25) – (3.30)). For the problem considered and the methods of order $p = 1, 2, 3, 4$ we have

$$\frac{\partial f}{\partial t} = 0, \quad \frac{\partial f}{\partial y} = 0.5,$$

$$\frac{\partial^2 f}{\partial t^2} = \frac{\partial^2 f}{\partial t\,\partial y} = \frac{\partial^2 f}{\partial y^2} = 0,$$

$$\frac{\partial^3 f}{\partial t^3} = \frac{\partial^3 f}{\partial t^2\,\partial y} = \frac{\partial^3 f}{\partial t\,\partial y^2} = \frac{\partial^3 f}{\partial y^3} = 0,$$

$$\frac{\partial^4 f}{\partial t^4} = \frac{\partial^4 f}{\partial t^3\,\partial y} = \frac{\partial^4 f}{\partial t^2\,\partial y^2} = \frac{\partial^4 f}{\partial t\,\partial y^3} = \frac{\partial^4 f}{\partial y^4} = 0.$$

From the results presented it follows that in each case the exact solution belongs to the interval solutions obtained (compare Table 2.2). Moreover, the widths of intervals increase when the orders of methods decrease. These widths are similar for explicit and implicit methods of the same order.

In Table 3.10 we present the solutions obtained by conventional methods corresponding to the interval ones. In implicit methods we assumed the accuracy 10^{-18}. One can observe that only the fourth order methods of Runge-Kutta (3.15), Hammer-Hollingsworth (3.17) and Butcher (3.22) give solutions that belongs to the interval solutions obtained. ■

Table 3.10. The solutions of the problem (2.2) at t = 1.0 obtained by conventional methods

Method	*Order*	*y*	*Error*
Euler's (3.11)	1	1.6486182460106884E+0000	$\approx 1.03\cdot 10^{-4}$
Euler-Cauchy (3.14)	2	1.6487212621146481E+0000	$\approx 8.59\cdot 10^{-9}$
Runge-Kutta (3.15)	4	1.6487212707001281E+0000	$\approx 3.69\cdot 10^{-17}$
Implicit midpoint rule (3.16)	2	1.6487212749936732E+0000	$\approx 4.29\cdot 10^{-9}$
Hammer-Hollingsworth (3.17)	4	1.6487212707001281E+0000	$\approx 1.53\cdot 10^{-17}$
Semi-implicit (3.19)	3	1.6487212706988755E+0000	$\approx 1.25\cdot 10^{-12}$
Butcher's semi-implicit (3.22)	4	1.6487212707001282E+0000	$\approx 4.34\cdot 10^{-18}$

Example 3.3

Let us take into account a multidimensional problem and, as an example, let us consider the problem (2.5) with $M = 0$, $\kappa = 1$, $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 0$, $y_4(0) = 1$, the exact solution of which is given by (2.6). Let us try to solve this problem by the following explicit interval methods:

- (3.41), i.e. by the interval version of the Euler-Cauchy method which is a two-stage method ($m = 2$) and of the second order ($p = 2$),

- (3.42), i.e. by the interval version of the Runge-Kutta method ($m = p = 4$),

and by the following two-stage ($m = 2$) implicit interval methods:

- the first diagonally implicit method (3.60), i.e. by the method with "+" signs in K_{1k} term ($p = 3$),
- (3.58), i.e. by the interval version of the Hammer-Hollingsworth method ($p = 4$).

For these methods we need interval extensions of $\psi_s(t, y)$ ($s = 1, 2, 3, 4$), that are expressed by $f_s(t, y)$ and their partial derivatives (see (3.25) – (3.30)). For our problem we have

$$f_1(t, y) = y_3, \quad f_2(t, y) = y_4,$$

$$f_3(t, y) = -\frac{y_1}{r^3}, \quad f_4(t, y) = -\frac{y_2}{r^3}, \quad r = \sqrt{y_1^2 + y_2^2},$$

$$\frac{\partial f_1}{\partial y_3} = \frac{\partial f_2}{\partial y_4} = 1, \quad \frac{\partial f_3}{\partial y_1} = \frac{1}{r^3}\left(\frac{3y_1^2}{r^2} - 1\right),$$

$$\frac{\partial f_3}{\partial y_2} = \frac{\partial f_4}{\partial y_1} = \frac{3y_1y_2}{r^5}, \quad \frac{\partial f_4}{\partial y_2} = \frac{1}{r^3}\left(\frac{3y_2^2}{r^2} - 1\right),$$

$$\frac{\partial^2 f_3}{\partial y_1^2} = \frac{3y_1}{r^5}\left(3 - \frac{5y_1^2}{r^2}\right), \quad \frac{\partial^2 f_3}{\partial y_1 \partial y_2} = \frac{\partial^2 f_4}{\partial y_1^2} = \frac{3y_2}{r^5}\left(1 - \frac{5y_1^2}{r^2}\right),$$

$$\frac{\partial^2 f_3}{\partial y_2^2} = \frac{\partial^2 f_4}{\partial y_1 \partial y_2} = \frac{3y_1}{r^5}\left(1 - \frac{5y_2^2}{r^2}\right), \quad \frac{\partial^2 f_4}{\partial y_2^2} = \frac{3y_2}{r^5}\left(3 - \frac{5y_2^2}{r^2}\right),$$

$$\frac{\partial^3 f_3}{\partial y_1^3} = \frac{3}{r^5}\left(3 - \frac{30y_1^2}{r^2} + \frac{35y_1^4}{r^4}\right), \quad \frac{\partial^3 f_3}{\partial y_1^2 \partial y_2} = \frac{\partial^3 f_4}{\partial y_1^3} = \frac{15y_1y_2}{r^7}\left(\frac{7y_1^2}{r^2} - 3\right),$$

$$\frac{\partial^3 f_3}{\partial y_1 \partial y_2^2} = \frac{\partial^3 f_4}{\partial y_1^2 \partial y_2} = \frac{3}{r^5}\left(1 - \frac{5y_1^2}{r^2} - \frac{5y_2^2}{r^2} + \frac{35y_1^2 y_2^2}{r^4}\right),$$

$$\frac{\partial^3 f_3}{\partial y_2^3} = \frac{\partial^3 f_4}{\partial y_1 \partial y_2^2} = \frac{15y_1y_2}{r^7}\left(\frac{7y_2^2}{r^2} - 3\right), \quad \frac{\partial^3 f_4}{\partial y_2^3} = \frac{3}{r^5}\left(3 - \frac{30y_2^2}{r^2} + \frac{35y_2^4}{r^4}\right),$$

$$\frac{\partial^4 f_3}{\partial y_1^4} = -\frac{15y_1}{r^7}\left(15 - \frac{70y_1^2}{r^2} + \frac{63y_1^4}{r^4}\right),$$

$$\frac{\partial^4 f_3}{\partial y_1^3 \partial y_2} = \frac{\partial^4 f_4}{\partial y_1^4} = -\frac{15 y_2}{r^7}\left(3 - \frac{42 y_1^2}{r^2} + \frac{63 y_1^4}{r^4}\right),$$

$$\frac{\partial^4 f_3}{\partial y_1^2 \partial y_2^2} = \frac{\partial^4 f_4}{\partial y_1^3 \partial y_2} = -\frac{15 y_1}{r^7}\left(3 - \frac{7 y_1^2}{r^2} - \frac{21 y_2^2}{r^2} + \frac{63 y_1^2 y_2^2}{r^4}\right),$$

$$\frac{\partial^4 f_3}{\partial y_1 \partial y_2^3} = \frac{\partial^4 f_4}{\partial y_1^2 \partial y_2^2} = -\frac{15 y_2}{r^7}\left(3 - \frac{21 y_1^2}{r^2} - \frac{7 y_2^2}{r^2} + \frac{63 y_1^2 y_2^2}{r^4}\right),$$

$$\frac{\partial^4 f_3}{\partial y_2^4} = \frac{\partial^4 f_4}{\partial y_1 \partial y_2^3} = -\frac{15 y_1}{r^7}\left(3 - \frac{42 y_2^2}{r^2} + \frac{63 y_2^4}{r^4}\right),$$

$$\frac{\partial^4 f_4}{\partial y_2^4} = -\frac{15 y_2}{r^7}\left(15 - \frac{70 y_2^2}{r^2} + \frac{63 y_2^4}{r^4}\right),$$

and all other partial derivatives are equal to 0.

In all of the interval methods we should determine Δ_t and Δ_{y_s} $(s = 1, 2, 3, 4)$. We need these values in order to find the maximum time intervals in which we can apply our interval methods (see Section 3.4 for details). In the problem considered we cannot take the whole domains of definitions of y_s $(s = 1, 2)$, since in such a case (according to the formulas for f_s and their partial derivatives) we would get divisions by intervals containing zero in the relevant interval extensions. Thus, let us assume that

$$\Delta_t = [0,\ 1],\quad \Delta_{y_1} = \Delta_{y_4} = \left[\underline{0.8},\ \overline{1.2}\right],\quad \Delta_{y_2} = \Delta_{y_3} = \left[\underline{-\ 0.2},\ \overline{0.2}\right],$$
$$T_0 = [0,\ 0],\quad Y_1(0) = Y_4(0) = [1,\ 1],\quad Y_2(0) = Y_3(0) = [0,\ 0].$$

For these intervals, $h_0 = 0.01$ and the appropriate values of M_s for $s = 1, 2, 3, 4$ (see (3.34)) we obtain the maximum integration intervals presented in Table 3.11. We see that these intervals are approximately equal to 0.085. Taking $t_{\max} = 0.05$ and splitting the interval [0, 0.05] into 10 parts, we get at

$$T = [4.9999999999999999\text{E}-0002,\ 5.0000000000000000\text{E}-0002]$$

the interval solutions presented in Table 3.12. For the implicit methods (3.58) and (3.60) in the iteration (3.55) we assumed the accuracy 10^{-18} and we obtained the results after 9-10 iterations in the method (3.60) and after 8-9 iterations in the method (3.58). It should be noted that in each case the exact solution is included in the interval solution obtained (compare Table 2.2).

Table 3.11. The maximum integration intervals for selected interval methods of Runge-Kutta type and the problem (2.5)

Kind of method	*Method*	*Method order*	M_s (s = 1, 2, 3, 4)	$t_{\max}$
Explicit	(3.41)	2	1	0.084939410899132
	(3.42)	4	0.01	0.085333275175408
Implicit	(3.60)	3	0.1	0.085327095409820
	(3.58)	4	0.01	0.085333316275920

Table 3.12. The interval solutions of the problem (2.5) with $M = 0$ and $\kappa = 1$ at T = [4.9999999999999999E−0002, 5.0000000000000000E−0002]

Method	Y_s	*Width of* Y_s
(3.41)	Y_1 = [9.9875024733284893E−0001, 9.9875027293211718E−0001]	$\approx 2.56\cdot 10^{-8}$
	Y_2 = [4.9979156577987591E−0002, 4.9979182150246763E−0002]	$\approx 2.56\cdot 10^{-8}$
	Y_3 = [−4.9979183082789349E−0002,−4.9979155738949369E−0002]	$\approx 2.73\cdot 10^{-8}$
	Y_4 = [9.9875025200135706E−0001, 9.9875027763381159E−0001]	$\approx 2.56\cdot 10^{-8}$
(3.42)	Y_1 = [9.9875026039496204E−0001, 9.9875026039496845E−0001]	$\approx 6.40\cdot 10^{-15}$
	Y_2 = [4.9979169270675078E−0002, 4.9979169270681472E−0002]	$\approx 6.39\cdot 10^{-15}$
	Y_3 = [−4.9979169270681769E-0002,−4.9979169270674931E−0002]	$\approx 6.84\cdot 10^{-15}$
	Y_4 = [9.9875026039496203E−0001, 9.9875026039496845E−0001]	$\approx 6.41\cdot 10^{-15}$
(3.60)	Y_1 = [9.9875026038887552E−0001, 9.9875026040167861E−0001]	$\approx 1.28\cdot 10^{-11}$
	Y_2 = [4.9979169258420737E−0002, 4.9979169271207795E−0002]	$\approx 1.28\cdot 10^{-11}$
	Y_3 = [−4.9979169272626200E−0002,−4.9979169258950521E−0002]	$\approx 1.37\cdot 10^{-11}$
	Y_4 = [9.9875026038880590E−0001,−9.9875026040162306E−0001]	$\approx 1.28\cdot 10^{-12}$
(3.58)	Y_1 = [9.9875026039496244E−0001, 9.9875026039496886E−0001]	$\approx 6.40\cdot 10^{-15}$
	Y_2 = [4.9979169270675104E−0002, 4.9979169270681498E−0002]	$\approx 6.39\cdot 10^{-15}$
	Y_3 = [−4.9979169270681762E−0002,−4.9979169270674925E−0002]	$\approx 6.84\cdot 10^{-15}$
	Y_4 = [9.9875026039496244E−0001, 9.9875026039496886E−0001]	$\approx 6.41\cdot 10^{-15}$

In the problem considered the maximum integration intervals are very small and are approximately equal to merely 1.35% of the orbit period. Of course, we can take the intervals obtained at the last time interval and start any of the methods again. For such an approach and the method (3.60), at $t = 0.1$ we obtain the results presented in Table 3.13. But a better approach is to take the interval solutions obtained (by any of these methods) as starting intervals for multistep algorithms presented in the next chapter. ■

Table 3.13. The interval solutions of the problem (2.5) with $M = 0$ and $\kappa = 1$ at $t = 0.1$ obtained by the method (3.60)

Y_s	*Width of* Y_s
$Y_1 =$ [9.9500416526600052E−0001, 9.9500416529236769E−0001]	$\approx 2.64 \cdot 10^{-11}$
$Y_2 =$ [9.9833416622023612E−0002, 9.9833416648261636E−0002]	$\approx 2.62 \cdot 10^{-11}$
$Y_3 =$ [−9.9833416652173548E−0002,−9.9833416622005311E−0002]	$\approx 3.02 \cdot 10^{-11}$
$Y_4 =$ [9.9500416526580906E−0001, 9.9500416529230890E−0001]	$\approx 2.65 \cdot 10^{-11}$

Example 3.4

Let us consider the initial value problem (2.7) – (2.8) and try to solve it by the semi-implicit method (3.59) of order 3 and by the semi-implicit method (3.61) of order 4. To find interval extensions of $\psi_s(t, y)$ $(s = 1, 2)$, we need interval extensions of $f_s(t, y)$ and their partial derivatives. For the problem considered we have

$$f_1(t, y) = 3y_1 + 2y_2, \quad f_2(t, y) = 4y_1 + y_2,$$

$$\frac{\partial f_1}{\partial y_1} = 3, \quad \frac{\partial f_1}{\partial y_2} = 2, \quad \frac{\partial f_2}{\partial y_1} = 4, \quad \frac{\partial f_2}{\partial y_2} = 1,$$

and all other partial derivatives are equal to 0. For

$$\Delta_t = [0, 1], \quad \Delta_{y_1} = \left[\underline{-\ 0.1},\ 50\right], \quad \Delta_{y_2} = \left[\underline{0.9},\ 50\right], \quad h_0 = 0.01,$$
$$T_0 = [0, 0], \quad Y_1(0) = [0, 0], \quad Y_2(0) = [1, 1],$$

and the adequate values of M_s, where $s = 1, 2$ (see (3.34)), we get the maximum integration intervals given in Table 3.14.

Table 3.14. The maximum integration intervals for selected semi-implicit interval methods of Runge-Kutta type and the problem (2.7) – (2.8)

Method	*Method order*	M_s $(s = 1, 2)$	$t_{\max}$
(3.59)	3	3	0.19598727111148148
(3.61)	4	1	0.19599998926020417

Taking $t_{\max} = 0.15$ and $h = 0.0015$ at

$$T = [1.4999999999999999\text{E}-0001,\ 1.5000000000000000\text{E}-0001]$$

we obtain the interval solutions shown in Table 3.15.

Table 3.15. The interval solutions of the problem (2.7) – (2.8) at T = [1.4999999999999999E−0001, 1.5000000000000000E−0001]

Method	Y_s	*Width of* Y_s
(3.59)	Y_1 = [4.1876401301701367E−0001, 4.1876401306215377E−0001]	$\approx 4.51 \cdot 10^{-11}$
	Y_2 = [1.2794719894422099E+0000, 1.2794719894873501E+0000]	$\approx 4.51 \cdot 10^{-11}$
(3.61)	Y_1 = [4.1876401339587620E−0001, 4.1876401339589875E−0001]	$\approx 2.25 \cdot 10^{-14}$
	Y_2 = [1.2794719898209340E+0000, 1.2794719898209566E+0000]	$\approx 2.26 \cdot 10^{-14}$

It should be mentioned that in the iteration (3.55) we assumed the accuracy 10^{-18} and we obtained the results after 9 iterations for the method (3.59) and after 7 iterations for the method (3.61). Of course, the exact solution belongs to the interval solutions obtained (compare Table 2.3). ■

Example 3.5

Finally, let us consider the motion of a simple pendulum given by the initial value problem (2.12) – (2.13) and try to solve it by the interval version of Alexander's diagonally implicit method with $\zeta = -\cos 50°$ (see (3.62)). In this problem we have

$$f_1(t, y) = -u^2 y_2, \quad f_2(t, y) = y_1, \quad \frac{\partial f_1}{\partial y_2} = -u^2, \quad \frac{\partial f_2}{\partial y_1} = 1,$$

and all other partial derivatives are equal to 0. Assuming that $M_1 = M_2 = 10$,

$$\Delta_t = [0, 2], \quad \Delta_{y_1} = [-2.5, 2.5], \quad \Delta_{y_2} = [-1, 1], \quad h_0 = 0.01,$$

$$T_0 = [0, 0], \quad Y_1(0) = [0, 0],$$

$$Y_2(0) = \left[\underline{0.52359877559829887}, \overline{0.52359877559829888}\right],$$

and taking the intervals

$$\left[\underline{9.80665}, \overline{9.80665}\right] \text{ and } \left[\underline{-0.642787609686539327}, \overline{-0.642787609686539326}\right]$$

to represent the gravitational acceleration g at Earth's surface and the value of ζ, respectively, we obtained the maximum integration interval equal to

$$0.19056048860322129.$$

This means that the method considered could be applied to about 10% of the period. Taking the iteration (3.56) with accuracy 10^{-18} for finding K_{ik} ($i = 1, 2, 3$), $t_{\max} = 0.1$ and splitting the interval [0, 0.1] into 20 parts, we obtained the interval solution shown in Table 3.16. The number of iterations in the process (3.56) was equal to 8 and, of course, the exact solution belongs to the intervals obtained. ■

Table 3.16. The interval solution of the problem (2.12) – (2.13) obtained by the method (3.62)

T	Y_s	*Width of* Y_s
[4.9999999999999999E−0002, 5.0000000000000001E−0002]	Y_1 = [−2.5568972570054137E−0001, −2.5568972569288790E−0001]	$\approx 7.65 \cdot 10^{-12}$
	Y_2 = [5.1719344066940931E−0001, 5.1719344067582207E−0001]	$\approx 6.41 \cdot 10^{-12}$
[9.9999999999999999E−0002, 1.0000000000000001E−0001]	Y_1 = [−5.0512359899636122E−0001, −5.0512359897780303E−0001]	$\approx 1.86 \cdot 10^{-11}$
	Y_2 = [4.9813415251025603E−0001, 4.9813415252354455E−0001]	$\approx 1.33 \cdot 10^{-11}$

All the examples presented in this section, like the others performed by the author, confirm the theoretical studies developed in the previous sections. Explicit and implicit interval methods of Runge-Kutta type can be used to find interval solutions (in floating-point interval arithmetic) for one- and multidimensional problems. In each case the exact solution belongs to the interval solutions obtained.

It is obvious that higher order methods give interval solutions with smaller widths. One should consider implicit interval methods which with small number of stages give solutions with greater order, e.g. the interval version of the Hammer-Hollingworth method (3.58).

Sometimes the integration interval, calculated on the basis of formulas given in Section 3.4, is small, but then the intervals obtained can be used as starting points for mulistep interval methods presented in the next chapter.

Chapter 4

Multistep Interval Methods

4.1. Some Conventional Multistep Methods

The Runge-Kutta methods presented in Chapter 3 are examples of *single step* schemes, because each step is defined solely in terms of its initial point. At the end of the nineteenth century F. Bashforth and J. C. Adams proposed an approach in which the approximate solution at a point depends on the solution values and derivative values before the immediately previous point [15]. These explicit methods are known as the Adams-Bashforth method. In the nineteen twenties F. R. Moulton developed the implicit type of these methods which are known at present as the Adams-Moulton methods. Other types of linear multistep methods were proposed by E. J. Nyström [150] and W. E. Milne [129, 130], which also proposed the predictor-corrector methods. The modern theory of linear multistep methods was developed by D. Dahlquist [47] and P. Henrici [67, 68].

In order to construct the Adams-Bashforth and Adams-Moulton methods let us rewrite the initial value problem (2.1) in the equivalent integral form

$$y(t) = y(t_{k-1}) + \int_{t_{k-1}}^{t} f(\tau, y(\tau))d\tau, \quad t > t_{k-1},$$

from which we have

$$y(t_k) = y(t_{k-1}) + \int_{t_{k-1}}^{t_k} f(\tau, y(\tau))d\tau. \tag{4.1}$$

To obtain multistep methods we approximate the function $f(\tau, y(\tau))$ by an adequate interpolation polynomial and then we integrate this polynomial.

Let us denote by $P(\tau)$ a polynomial of the degree $n - 1$ such that

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

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:

$$\begin{aligned} P(t_{k-1} + th) &= f(t_{k-1}, y(t_{k-1})) + t\nabla f(t_{k-1}, y(t_{k-1})) + \\ &+ \ldots + \frac{t(t+1)\ldots(t+n-2)}{(n-1)!}\nabla^{n-1} f(t_{k-1}, y(t_{k-1})), \end{aligned} \tag{4.2}$$

where ∇ denotes the backward difference operator[1].

We approximate the integrand in (4.1) 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

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

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

$$y(t_k) = y(t_{k-1}) + h\sum_{j=0}^{n-1} \gamma_j \nabla^j f(t_{k-1}, y(t_{k-1})) + h^{n+1}\gamma_n \psi(\eta, y(\eta)), \tag{4.3}$$

where

$$\nabla^j f(t_{k-1}, y(t_{k-1})) = \sum_{m=0}^{j} (-1)^m \binom{j}{m} f(t_{k-1-m}, y(t_{k-1-m})), \tag{4.4}$$

$$\gamma_0 = 1, \quad \gamma_j = \frac{1}{j!}\int_0^1 s(s+1)\ldots(s+j-1)ds \quad \text{for } j = 1, 2, \ldots, n, \tag{4.5}$$

and $\psi(\eta, y(\eta)) \equiv f^{(n)}(\eta, y(\eta)) \equiv y^{(n+1)}(\eta)$, where η is an intermediate point in the interval $[t_{k-n}, t_k]$.

After replacing the unknown values $y(t_{k-n}), y(t_{k-n+1}), \ldots, y(t_{k-1})$ with approximations $y_{k-n}, y_{k-n+1}, \ldots, y_{k-1}$ obtained by applying another method (for example by a Runge-Kutta method) and excepting the error term $h^{k+1}\gamma_n\psi(\eta, y(\eta))$, for finding y_k (which approximate $y(t_k)$), we are given the following formula known as the *n-step explicit Adams-Bashforth method* [38, 41, 50, 62, 77, 97]:

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

$$y_k = y_{k-1} + h\sum_{j=0}^{n-1} \gamma_j \nabla^j f(t_{k-1}, y_{k-1}). \tag{4.6}$$

Using (4.4) we can write (4.6) in the form

$$y_k = y_{k-1} + h\sum_{j=1}^{n} \beta_{nj} f(t_{k-j}, y(t_{k-j})), \tag{4.7}$$

where

$$\beta_{nj} = (-1)^{j-1} \sum_{m=j-1}^{n-1} \binom{m}{j-1} \gamma_m, \quad j = 1, 2, \dots, n. \tag{4.8}$$

In particular, from (4.6) and (4.7) for a given n we get the following methods:

- $n = 1$ (*Euler's method*)

$$y_k = y_{k-1} + hf(t_{k-1}, y_{k-1}), \tag{4.9}$$

- $n = 2$

$$y_k = y_{k-1} + \frac{h}{2}[3f(t_{k-1}, y_{k-1}) - f(t_{k-2}, y_{k-2})], \tag{4.10}$$

- $n = 3$

$$y_k = y_{k-1} + \frac{h}{12}[23f(t_{k-1}, y_{k-1}) - 16f(t_{k-2}, y_{k-2}) + 5f(t_{k-3}, y_{k-3})], \tag{4.11}$$

- $n = 4$

$$\begin{aligned} y_k = y_{k-1} + \frac{h}{24}[&55f(t_{k-1}, y_{k-1}) - 59f(t_{k-2}, y_{k-2}) + 37f(t_{k-3}, y_{k-3}) - \\ &- 9f(t_{k-4}, y_{k-4})]. \end{aligned} \tag{4.12}$$

Let $\overline{P}(\tau)$ be a polynomial of the degree n such that

$$\overline{P}(t_{k-j}) = f(t_{k-j}, y(t_{k-j})), \quad j = 0, 1, \dots, n.$$

Changing the variable τ for t in such a way that $\tau = t_k + th$, we can write the polynomial $\overline{P}(\tau)$ in the form

$$\begin{aligned} \overline{P}(t_k + th) = f(t_k, y(t_k)) + t\nabla f(t_k, y(t_k)) + \\ + \dots + \frac{t(t+1)\dots(t+n-1)}{n!}\nabla^n f(t_k, y(t_k)). \end{aligned} \tag{4.13}$$

If we approximate the integrand in (4.1) by the polynomial $\overline{P}(\tau)$, i.e. we substitute

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

where $\bar{r}(\tau)$ is an interpolation error and

$$\bar{r}(t_k + th) = f^{(n+1)}(\bar{\zeta}(t))h^{n+1}\frac{t(t+1)\dots(t+n)}{(n+1)!},$$

where $\bar{\zeta}(t)$ is an intermediate point in the interval $[t_{k-n}, t_k]$, then after integration we get

$$y(t_k) = y(t_{k-1}) + h\sum_{j=0}^{n} \bar{\gamma}_j \nabla^j f(t_k, y(t_k)) + h^{n+2}\bar{\gamma}_{n+1}\bar{\psi}(\bar{\eta}, y(\bar{\eta})), \tag{4.14}$$

$$\nabla^j f(t_k, y(t_k)) = \sum_{m=0}^{j} (-1)^m \binom{j}{m} f(t_{k-m}, y(t_{k-m})), \tag{4.15}$$

$$\bar{\gamma}_0 = 1, \quad \bar{\gamma}_j = \frac{1}{j!}\int_{-1}^{0} t(t+1)\dots(t+j-1)dt \ \text{ for } j = 1, 2, \dots, n+1, \tag{4.16}$$

where $\bar{\psi}(\bar{\eta}, y(\bar{\eta})) \equiv f^{(n+1)}(\bar{\eta}, y(\bar{\eta})) \equiv y^{(n+2)}(\bar{\eta})$ and $\bar{\eta}$ is an intermediate point in the interval $[t_{k-n}, t_k]$.

If in (4.14) we ignore the term $h^{n+2}\bar{\gamma}_{n+1}\bar{\psi}(\bar{\eta}, y(\bar{\eta}))$ and replace the unknown values $y(t_{k-n}), y(t_{k-n+1}), \dots, y(t_{k-1})$ with approximations $y_{k-n}, y_{k-n+1}, \dots, y_{k-1}$, then in order to find $y_k \approx y(t_k)$ we obtain the *n-step implicit Adams-Moulton method* [38, 41, 50, 62, 77, 97]:

$$y_k = y_{k-1} + h\sum_{j=0}^{n} \bar{\gamma}_j \nabla^j f(t_k, y_k). \tag{4.17}$$

Taking into account (4.15) and the relation

$$\bar{\beta}_{nj} = (-1)^j \sum_{m=j}^{n} \binom{m}{j} \bar{\gamma}_m, \quad j = 0, 1, \dots, n, \tag{4.18}$$

we can write (4.17) in the form

$$y_k = y_{k-1} + h\sum_{j=0}^{n} \bar{\beta}_{nj} f(t_{k-j}, y_{k-j}). \tag{4.19}$$

From (4.17) and (4.19) we have

● $n = 1$ (the *trapezoidal rule*)

$$y_k = y_{k-1} + \frac{h}{2}[f(t_k, y_k) + f(t_{k-1}, y_{k-1})], \tag{4.20}$$

- $n = 2$

$$y_k = y_{k-1} + \frac{h}{12}[5f(t_k, y_k) + 8f(t_{k-1}, y_{k-1}) - f(t_{k-2}, y_{k-2})], \tag{4.21}$$

- $n = 3$

$$\begin{aligned} y_k = y_{k-1} + \frac{h}{24}[9f(t_k, y_k) + 19f(t_{k-1}, y_{k-1}) - 5f(t_{k-2}, y_{k-2}) + \\ + f(t_{k-3}, y_{k-3})]. \end{aligned} \tag{4.22}$$

The initial value problem (2.1) can be also written in another equivalent integral form, namely

$$y(t) = y(t_{k-2}) + \int_{t_{k-2}}^{t} f(\tau, y(\tau))d\tau, \quad t > t_{k-2},$$

from which we get

$$y(t_k) = y(t_{k-1}) + \int_{t_{k-2}}^{t_k} f(\tau, y(\tau))d\tau. \tag{4.23}$$

If we approximate the integrand in (4.23) by the polynomial (4.2), then we obtain

$$\begin{aligned} y(t_k) = y(t_{k-2}) + h\sum_{j=0}^{n-1} \nu_j \nabla^j f(t_{k-1}, y(t_{k-1})) + \\ + h^{n+1}\left[\nu_n^* \psi\left(\eta^*, y\left(\eta^*\right)\right) + \nu_n^{**} \psi\left(\eta^{**}, y\left(\eta^{**}\right)\right)\right], \end{aligned} \tag{4.24}$$

where $\psi(\eta, y(\eta)) \equiv f^{(n)}(\eta, y(\eta)) \equiv y^{(n+1)}(\eta)$, η^* and η^{**} are intermediate points in the interval $[t_{k-n}, t_k]$, and

$$\begin{aligned} \nu_0 = 2, \quad \nu_j = \frac{1}{j!}\int_{-1}^{1} t(t+1)\ldots(t+j-1)dt, \quad j = 1, 2, \ldots, n-1, \\ \nu_n^* = \frac{1}{n!}\int_{-1}^{0} t(t+1)\ldots(t+n-1)dt, \quad \nu_n^{**} = \frac{1}{n!}\int_{0}^{1} t(t+1)\ldots(t+n-1)dt, \end{aligned} \tag{4.25}$$

The coefficients ν_n^* and ν_n^{**} are very important in the interval methods considered (see Section 4.3).

From (4.24) the conventional *n-step method of Nyström* follows immediately. We have

$$y_k = y_{k-2} + h\sum_{j=0}^{n-1} \nu_j \nabla^j f(t_{k-1}, y_{k-1}). \tag{4.26}$$

The formula (4.26) can be also written in the form

$$y_k = y_{k-2} + h\sum_{j=1}^{n} \delta_{nj} f(t_{k-j}, y_{k-j}), \tag{4.27}$$

where

$$\delta_{nj} = (-1)^{j-1} \sum_{m=j-1}^{n-1} \binom{m}{j-1} \nu_m. \tag{4.28}$$

In particular, from (4.26) (or (4.27)) for a given n we get:

- $n = 1, 2$ (the *midpoint rule*)

$$y_k = y_{k-2} + 2hf(t_{k-1}, y_{k-1}), \tag{4.29}$$

- $n = 3$

$$y_k = y_{k-2} + \frac{h}{3}[7f(t_{k-1}, y_{k-1}) - 2f(t_{k-2}, y_{k-2}) + f(t_{k-3}, y_{k-3})], \tag{4.30}$$

- $n = 4$

$$\begin{aligned} y_k = y_{k-2} + \frac{h}{3}[8f(t_{k-1}, y_{k-1}) - 5f(t_{k-2}, y_{k-2}) + 4f(t_{k-3}, y_{k-3}) - \\ - f(t_{k-4}, y_{k-4})]. \end{aligned} \tag{4.31}$$

If in (4.23) we use the polynomial $\overline{P}(\tau)$ given by (4.13), then we obtain the exact relation containing either backward differences, i.e.

$$\begin{aligned} y(t_k) = y(t_{k-2}) + h\sum_{j=0}^{n} \overline{\nu}_j \nabla^j f(t_k, y(t_k)) + \\ + h^{n+2}\left[\overline{\nu}^*_{n+1}\overline{\psi}\left(\overline{\eta}^*, y\left(\overline{\eta}^*\right)\right) + \overline{\nu}^{**}_{n+1}\overline{\psi}\left(\overline{\eta}^{**}, y\left(\overline{\eta}^{**}\right)\right)\right], \end{aligned} \tag{4.32}$$

or only the values of the function, i.e.

$$\begin{aligned} y(t_k) = y(t_{k-2}) + h\sum_{j=0}^{n} \overline{\delta}_{nj} f(t_{k-j}, y(t_{k-j})) + \\ + h^{n+2}\left[\overline{\nu}^*_{n+1}\overline{\psi}\left(\overline{\eta}^*, y\left(\overline{\eta}^*\right)\right) + \overline{\nu}^{**}_{n+1}\overline{\psi}\left(\overline{\eta}^{**}, y\left(\overline{\eta}^{**}\right)\right)\right], \end{aligned} \tag{4.33}$$

where $\psi\left(\overline{\eta}, y\left(\overline{\eta}\right)\right) \equiv f^{(n+1)}\left(\overline{\eta}, y\left(\overline{\eta}\right)\right) \equiv y^{(n+2)}\left(\overline{\eta}\right)$, $\overline{\eta}^*$ and $\overline{\eta}^{**}$ are some points in $[t_{k-n}, t_k]$,

$$\overline{v}_0 = 2, \quad \overline{v}_j = \frac{1}{j!}\int\limits_{-2}^{0} t(t+1)\ldots(t+j-1)dt, \quad j = 1, 2, \ldots, n,$$

$$\overline{v}^*_{n+1} = \frac{1}{(n+1)!}\int\limits_{-1}^{0} t(t+1)\ldots(t+n)dt, \quad \overline{v}^{**}_{n+1} = \frac{1}{(n+1)!}\int\limits_{-2}^{-1} t(t+1)\ldots(t+n)dt, \tag{4.34}$$

$$\overline{\delta}_{nj} = (-1)^j \sum_{m=j}^{n} \binom{m}{j} \overline{v}_m, \quad j = 0, 1, \ldots, n.$$

From both of these formulas, i.e. from (4.32) or (4.33), we can get the conventional *n-step implicit method of Milne-Simpson*:

$$y_k = y_{k-2} + h\sum_{j=0}^{n} \overline{v}_j \nabla^j f(t_k, y_k). \tag{4.35}$$

In the case of intervals the formulas (4.32) and (4.33) give quite different multistep methods (see Section 4.5 for details).

From (4.35) for a given n we get the following methods:

- $n = 1$ (the *midpoint rule*)

$$y_k = y_{k-2} + 2hf(t_{k-1}, y_{k-1}), \tag{4.36}$$

- $n = 2, 3$ (the *Milne method*)

$$y_k = y_{k-2} + \frac{h}{3}[f(t_k, y_k) + 4f(t_{k-1}, y_{k-1}) + f(t_{k-2}, y_{k-2})]. \tag{4.37}$$

4.2. Interval Methods of Adams-Bashforth Type

4.2.1. Basic Formulas

Let Δ_t and Δ_y be the sets in which the function $f(t, y)$ occurring in the initial value problem (2.1) is defined (see Section 3.2 for details). Let $F(T, Y)$ denote an interval extension of $f(t, y)$, and let $\Psi(T, Y)$ denote an interval extension of $\psi(t, y(t)) \equiv f^{(n)}(t, y(t)) \equiv y^{(n+1)}(t)$. Let the assumptions about $F(T, Y)$ and $\Psi(T, Y)$ be the same as in Section 3.2. Moreover, let us assume 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. We can obtain such Y_k by applying an interval one-step method, for example an interval method of Runge-

-Kutta type presented in Chapter 3. In [167] Yu. I. Shokin proposed the explicit method of Adams-Bashforth type given by the following formula:

$$\begin{aligned} Y_k = {} & Y_{k-1} + h(\gamma_0 F_{k-1} + \gamma_1 \nabla F_{k-1} + \gamma_2 \nabla^2 F_{k-1} + \ldots + \gamma_{n-1} \nabla^{n-1} F_{n-1}) + \\ & + h^{n+1} \gamma_n \Psi(T_{k-1} + [-(n-1)h, 0], Y_{k-1} + [-(n-1)h, 0] F(\mathit{\Delta}_t, \mathit{\Delta}_y)), \end{aligned} \tag{4.38}$$

where

$$F_{k-1} = F(T_{k-1}, Y_{k-1}), \quad h = \frac{a}{m}, \quad t_k = kh \in T_k, \quad k = n, n+1, \ldots, m,$$

$a < \infty$ is a constant occurring in the definition of $\mathit{\Delta}_t$, and the coefficients γ_j ($j = 0, 1, \ldots, n$) are given by (4.5). Unfortunately, it can be shown that the formula (4.38) fails in the simplest case, i.e. when $n = 1$ [80].

Example 4.1

Let us consider the initial value problem of the form

$$y'(t) = 0.5y, \quad y(0) = 1, \tag{4.39}$$

where $t \in [0, 1]$. The exact solution of this problem is given by

$$y(t) = \exp(0.5t). \tag{4.40}$$

Applying (4.38) for $m = 2000$, $h = 0.0005$ and $Y_0 = [0, 0]$, we get (in floating-point interval arithmetic) the interval solution of the initial value problem (4.39) at $t = 1$ as follows

$$Y_{2000} = [1.64872126211491595, 1.64872126211491651].$$

On the other hand, the formula (4.40) yields the exact solution

$$y_{\text{exact}}(1) \approx 1.64872127070012815.$$

This simple example shows that $y_{\text{exact}}(1) \notin Y_{2000}$, and hence Y_{2000} is not the correct interval solution of the initial value problem (4.39). ■

After careful consideration of Shokin's formula (4.38), we have found that the reason for such behavior of the method is a defective error term of the form

$$h^{n+1} \gamma_n \Psi(T_{k-1} + [-(n-1)h, 0], Y_{k-1} + [-n(n-1)h, 0] F(\mathit{\Delta}_t, \mathit{\Delta}_y)). \tag{4.41}$$

As we checked, the term (4.41) does not improve the interval solution, and for this reason getting the correct result becomes impossible. A detailed study of conventional Adams-Bashforth methods led us to make some essential modifications in the original formula. The correct error term should be written as follows [80]:

$$h^{n+1} \gamma_n \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h] F(\mathit{\Delta}_t, \mathit{\Delta}_y)). \tag{4.42}$$

Substituting (4.42) into (4.38) we obtain the correct formula for *interval methods of Adams-Bashforth type* of the following form:

$$\begin{gathered}Y_k = Y_{k-1} + h\sum_{j=0}^{n-1} \gamma_j \nabla^j F_{k-1} + \\ + h^{n+1}\gamma_n \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y)), \\ k = n, n+1, \dots, m.\end{gathered} \tag{4.43}$$

In particular, for a given n we get the following methods:

- ● $n = 1$ (the interval version of Euler's method (4.9))

$$Y_k = Y_{k-1} + hF(T_{k-1}, Y_{k-1}) + \frac{h^2}{2}\Psi(T_{k-1} + [0, h], Y_{k-1} + [0, h]F(\Delta_t, \Delta_y)), \tag{4.44}$$

- ● $n = 2$ (the interval version of the method (4.10))

$$\begin{gathered}Y_k = Y_{k-1} + \frac{h}{2}(3F(T_{k-1}, Y_{k-1}) - F(T_{k-2}, Y_{k-2})) + \\ + \frac{5h^3}{12}\Psi(T_{k-1} + [-h, h], Y_{k-1} + [-h, h]F(\Delta_t, \Delta_y)),\end{gathered} \tag{4.45}$$

- ● $n = 3$ (the interval version of the method (4.11))

$$\begin{gathered}Y_k = Y_{k-1} + \frac{h}{12}(23F(T_{k-1}, Y_{k-1}) - 16F(T_{k-2}, Y_{k-2}) + 5F(T_{k-3}, Y_{k-3})) + \\ + \frac{3h^4}{8}\Psi(T_{k-1} + [-2h, h], Y_{k-1} + [-2h, h]F(\Delta_t, \Delta_y)),\end{gathered} \tag{4.46}$$

- ● $n = 4$ (the interval version of the method (4.12))

$$\begin{gathered}Y_k = Y_{k-1} + \frac{h}{24}(55F(T_{k-1}, Y_{k-1}) - 59F(T_{k-2}, Y_{k-2}) + 37F(T_{k-3}, Y_{k-3}) - \\ - 9F(T_{k-4}, Y_{k-4})) + \\ + \frac{251h^5}{720}\Psi(T_{k-1} + [-3h, h], Y_{k-1} + [-3h, h]F(\Delta_t, \Delta_y)).\end{gathered} \tag{4.47}$$

Since

$$\nabla^j F_{k-1} = \sum_{m=0}^{j} (-1)^m \binom{j}{m} F_{k-1-m},$$

the formula (4.43) can be written in the equivalent form

$$Y_k = Y_{k-1} + h\sum_{j=1}^{n} \beta_{nj} F_{k-j} + \tag{4.48}$$
$$+ h^{n+1}\gamma_n \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y)),$$

where the coefficients β_{nj} $(j = 1, 2, \dots, n)$ are given by (4.8).

Example 4.2

For comparison, let us apply modified formula (4.43) (or (4.48)) with $n = 1$, i.e. the formula (4.44), to the initial value problem (4.39). For $m = 2000$, $h = 0.0005$ and $Y_0 = [0, 0]$ we get the interval solution at $t = 1$ as follows

$$Y_{2000} = [1.64872126211491595, 1.64872128787216209],$$

and we have $y_{\text{exact}}(1) \in Y_{2000}$. Thus, in this case we have obtained the correct interval solution of the initial value problem (4.39). ■

Other numerical results of application of the interval methods of Adams-Bashforth type are presented in Section 4.7.

4.2.2. An Inclusion of the Exact Solution by Interval Solutions

For the methods (4.43) we can prove that the exact solution of the initial value problem (2.1) belongs to the intervals obtained by these methods [80]. Before that, it is convenient to present the following

Lemma 4.1. *If* $(t_i, y(t_i)) \in (T_i, Y_i)$ *for* $i = k-n, k-n+1, \dots, k-1$, *where* $Y_i = Y(t_i)$, *then for any* $j = 0, 1, \dots, n-1$ *we have*

$$\nabla^j f(t_{k-1}, y(t_{k-1})) \in \nabla^j F(T_{k-1}, Y_{k-1}). \tag{4.49}$$

Proof. Since $F(T, Y)$ is an interval extension of $f(t, y)$, and $(t_i, y(t_i)) \in (T_i, Y_i)$ for $i = k-n, k-n+1, \dots, k-1$, we can write

$$f(t_{k-1-m}, y(t_{k-1-m})) \in F(T_{k-1-m}, Y_{k-1-m}), \quad m = 0, 1, \dots, j.$$

This implies that

$$\sum_{m=0}^{j} (-1)^m \binom{j}{m} f(t_{k-1-m}, y(t_{k-1-m})) \in \sum_{m=0}^{j} (-1)^m \binom{j}{m} F(T_{k-1-m}, Y_{k-1-m}). \tag{4.50}$$

But

$$\sum_{m=0}^{j}(-1)^m\binom{j}{m}F(T_{k-1-m}, Y_{k-1-m}) = \nabla^j F(T_{k-1}, Y_{k-1}). \tag{4.51}$$

From (4.4), (4.50) and (4.51) the relation (4.49) follows immediately. ■

Theorem 4.1. *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* (2.1) *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 methods* (4.43).

Proof. Let us consider the formula (4.3) for $k = n$. We get

$$y(t_n) = y(t_{n-1}) + h\sum_{j=0}^{n-1}\gamma_j\nabla^j f(t_{n-1}, y(t_{n-1})) + h^{n+1}\gamma_n\psi(\eta, y(\eta)), \tag{4.52}$$

where $\eta \in [t_0, t_n]$. From the assumption we have $y(t_{n-1}) \in Y_{n-1}$, and from the Lemma 4.1 it follows that

$$h\sum_{j=0}^{n-1}\gamma_j\nabla^j f(t_{n-1}, y(t_{n-1})) \in h\sum_{j=0}^{n-1}\gamma_j\nabla^j F(T_{n-1}, Y_{n-1}).$$

Applying Taylor's formula we have

$$y(\eta) = y(t_{n-1}) + (\eta - t_{n-1})y'(t_{n-1} + \vartheta(\eta - t_{n-1})), \tag{4.53}$$

where $\vartheta \in [0, 1]$. Because $\eta \in [t_0, t_n]$ and $t_i = ih$ for $i = 0, 1, \ldots, m$, we get

$$\eta - t_{n-1} \in [-(n-1)h, h]. \tag{4.54}$$

Moreover, $y'(t) = f(t, y(t))$. Since

$$f[t_{n-1} + \vartheta(\eta - t_{n-1}), y(t_{n-1} + \vartheta(\eta - t_{n-1}))] \in F(\Delta_t, \Delta_y),$$

then

$$y'(t_{n-1} + \vartheta(\eta - t_{n-1})) \in F(\Delta_t, \Delta_y).$$

Taking into account the above considerations, from the formula (4.53) we get

$$y(\eta) \in Y_{n-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y). \tag{4.55}$$

As we assumed, Ψ is an interval extension of ψ. Thus, applying (4.54) and (4.55), we have

$$h^{n+1}\gamma_n\psi(\eta, y(\eta)) \in$$
$$\in h^{n+1}\gamma_n\Psi(T_{n-1} + [-(n-1)h, h], Y_{n-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y)).$$

Thus, we have shown that $y(t_k)$ belongs to the interval

$$Y_{n-1} + h\sum_{j=0}^{n-1} \gamma_j \nabla^j F(T_{n-1}, Y_{n-1}) +$$
$$+ h^{n+1}\gamma_n\Psi(T_{n-1} + [-(n-1)h, h], Y_{n-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y)).$$

but — according to the formula (4.43) — this is the interval Y_k. This conclusion ends the proof for $k = n$. In a similar way we can show the thesis of this theorem for $k = n + 1, n + 2, \dots, m$. ■

4.2.3. Widths of Interval Solution

An estimation of the widths of interval solutions obtained by the interval methods of Adams-Bashforth type is given in the following theorem [80, 167]:

Theorem 4.2. *If the intervals Y_k for $k = 0, 1, \dots, n-1$ are known, $t_i = ih \in T_i$, $i = 0, 1, \dots, m$, $h = \xi/m$, and Y_k for $k = n, n+1, \dots, m$ are obtained from* (4.45) *or* (4.50)*, then*

$$w(Y_k) \le A \max_{q=0,1,\dots,n-1} w(Y_q) + B \max_{j=1,2,\dots,m-1} w(T_j) + Ch^n, \tag{4.56}$$

where the constant A, B and C are independent of h.

Proof. From (4.48) we get

$$\begin{aligned} w(Y_k) \le w(Y_{k-1}) + h\sum_{j=1}^{n} \left|\beta_{nj}\right| w(F_{k-j}) + \\ + h^{n+1}\gamma_n w(\Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y))).\end{aligned} \tag{4.57}$$

We assumed that Ψ is monotonic with respect to inclusion. Moreover, if the step size h is such that satisfies the conditions

$$T_{k-1} + [-(n-1)h, h] \subset \Delta_t,$$
$$Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y) \subset \Delta_y,$$

then

$$\Psi(T_{k-1}+[-(n-1)h,h], Y_{k-1}+[-(n-1)h,h]F(\Delta_t,\Delta_y)) \subset \Psi(\Delta_t,\Delta_y). \tag{4.58}$$

From (4.58) we have

$$\begin{gathered} w(\Psi(T_{k-1}+[-(n-1)h,h], Y_{k-1}+[-(n-1)h,h]F(\Delta_t,\Delta_y))) \le \\ \le w(\Psi(\Delta_t,\Delta_y)). \end{gathered} \tag{4.59}$$

We also assumed that for the function *F* there exists a constant $\Lambda > 0$ such that

$$w(F_{k-j}) \le \Lambda(w(T_{k-j})+w(Y_{k-j})).$$

Therefore, from the inequality (4.57) we get

$$w(Y_k) \le w(Y_{k-1}) + h\Lambda\beta_n\sum_{j=1}^{n}(w(T_{k-j})+w(Y_{k-j})) + h^{n+1}\gamma_n w(\Psi(\Delta_t,\Delta_y)), \tag{4.60}$$

where

$$\beta_n = \max_{j=1,2,\ldots,n}\left|\beta_{nj}\right|.$$

Denoting

$$\beta = h\Lambda\beta_n, \quad \alpha = 1+\beta, \quad \gamma = h^{n+1}\gamma_n, \tag{4.61}$$

we can write (4.60) in the form

$$w(Y_k) \le \alpha\sum_{j=1}^{n} w(Y_{k-j}) + \beta\sum_{j=1}^{n} w(T_{k-j}) + \gamma w(\Psi(\Delta_t,\Delta_y)). \tag{4.62}$$

From (4.62) for $k = n$ we have

$$w(Y_n) \le \alpha\sum_{j=1}^{n} w(Y_{n-j}) + \beta\sum_{j=1}^{n} w(T_{n-j}) + \gamma w(\Psi(\Delta_t,\Delta_y)). \tag{4.63}$$

and for $k = n + 1$ we get

$$w(Y_{n+1}) \le \alpha w(Y_n) + \alpha\sum_{j=1}^{n-1} w(Y_{n-j}) + \sum_{j=1}^{n} w(T_{n+1-j}) + \gamma w(\Psi(\Delta_t,\Delta_y)).$$

Applying (4.63) to this inequality we obtain

$$w(Y_{n+1}) \le (\alpha^2 + \alpha)\sum_{j=1}^{n} w(Y_{n-j}) + \\ + \beta\left(\alpha\sum_{j=1}^{n} w(T_{n-j}) + \sum_{j=1}^{n} w(T_{n+1-j})\right) + \gamma(\alpha + 1)w(\Psi(\Delta_t, \Delta_y)). \tag{4.64}$$

From (4.62) for $k = n + 2$ we get

$$w(Y_{n+2}) \le \alpha w(Y_{n+1}) + \alpha w(Y_n) + \alpha\sum_{j=1}^{n-2} w(Y_{n-j}) + \\ + \beta\sum_{j=1}^{n} w(T_{n+2-j}) + \gamma w(\Psi(\Delta_t, \Delta_y)).$$

Insertion of (4.63) and (4.64) into this inequality yields

$$w(Y_{n+2}) \le (\alpha^3 + \alpha^2 + \alpha)\sum_{j=1}^{n} w(Y_{n-j}) + \\ + \beta\left((\alpha^2 + \alpha)\sum_{j=1}^{n} w(T_{n-j}) + \alpha\sum_{j=1}^{n} w(T_{n+1-j}) + \sum_{j=1}^{n} w(T_{n+2-j})\right) + \\ + \gamma(\alpha^2 + 2\alpha + 1)w(\Psi(\Delta_t, \Delta_y)). \tag{4.65}$$

Now, from (4.62) for $k = n + 3$ we get

$$w(Y_{n+3}) \le \alpha w(Y_{n+2}) + \alpha w(Y_{n+1}) + \alpha w(Y_n) + \alpha\sum_{j=1}^{n-3} w(Y_{n-j}) + \\ + \beta\sum_{j=1}^{n} w(T_{n+3-j}) + \gamma w(\Psi(\Delta_t, \Delta_y)).$$

Applying (4.63), (4.64) and (4.65) to the above formula we have

$$w(Y_{n+3}) \le (\alpha^4 + 3\alpha^3 + 3\alpha^2 + \alpha)\sum_{j=1}^{n} w(Y_{n-j}) +$$

$$+\beta\left((\alpha^3+\alpha^2+\alpha)\sum_{j=1}^{n} w(T_{n-j})+(\alpha^2+\alpha)\sum_{j=1}^{n} w(T_{n+1-j})+\right.$$
$$\left.+\alpha\sum_{j=1}^{n} w(T_{n+2-j})+\sum_{j=1}^{n} w(T_{n+3-j})\right)+$$
$$+\gamma(\alpha^3+3\alpha^2+3\alpha+1)w(\Psi(\Delta_t,\Delta_y)).$$

Thus, for each $i = 1, 2, \ldots, m - n$ we have

$$w(Y_{n+i}) \le \left(\sum_{l=0}^{i}\binom{i}{l}\alpha^{l+1}\right)\left(\sum_{j=1}^{n} w(Y_{n-j})\right)+$$
$$+\beta\sum_{p=0}^{i}\left(\sum_{l=0}^{p-1}\binom{p-1}{l}\alpha^{l+1}\right)\left(\sum_{j=1}^{n} w(T_{n+i-p-j})\right)+$$
$$+\gamma\left(\sum_{l=0}^{i}\binom{i}{l}\alpha^{l}\right)w(\Psi(\Delta_t,\Delta_y)).$$

Applying the notation (4.61) we obtain

$$w(Y_{n+i}) \le n\sum_{l=0}^{i}\binom{i}{l}(1+h\Lambda\beta_n)^{l+1}\max_{q=0,1,\ldots,n-1} w(Y_q)+$$
$$+h\Lambda\beta_n n\sum_{p=0}^{i}\sum_{l=0}^{p-1}\binom{p-1}{l}(1+h\Lambda\beta_n)^{l+1}\max_{j=0,1,\ldots,n+i-1} w(T_j)+ \tag{4.66}$$
$$+h^{n+1}\gamma_n\sum_{l=0}^{i}\binom{i}{l}(1+h\Lambda\beta_n)^{l} w(\Psi(\Delta_t,\Delta_y)).$$

Let us note that

$$\binom{i}{l} \le (m-n)! \quad \text{for } l = 0, 1, \ldots, i,$$
$$\binom{p-1}{l} \le (m-n)! \quad \text{for } l = 0, 1, \ldots, p-1,$$

$$(1 + h\Lambda\beta_n)^{l+1} \le \exp(\xi\Lambda\beta_n),$$

$$\sum_{l=0}^{p-1}(1 + h\Lambda\beta_n)^{l+1} \le \frac{\exp(\xi\Lambda\beta_n) - 1}{h\Lambda\beta_n}.$$

On the basis of the above we can make the following estimates:

$$n\sum_{l=0}^{i}\binom{i}{l}(1 + h\Lambda\beta_n)^{l+1} \le m(m - n + 1)!\exp(\xi\Lambda\beta_n),$$

$$n\sum_{p=0}^{i}\sum_{l=0}^{p-1}\binom{p-1}{l}(1 + h\Lambda\beta_n)^{l+1} \le m(m - n + 1)!\frac{\exp(\xi\Lambda\beta_n) - 1}{h\Lambda\beta_n},$$

$$\sum_{l=0}^{i}\binom{i}{l}(1 + h\Lambda\beta_n)^{l} \le (m - n + 1)!\frac{\exp(\xi\Lambda\beta_n) - 1}{h\Lambda\beta_n}.$$

Thus, from (4.66) we finally get

$$w(Y_{n+i}) \le A \max_{q=0,1,\ldots,n-1} w(Y_q) + B \max_{j=0,1,\ldots,m-1} w(T_j) + Ch^n \tag{4.67}$$

for each $i = 0, 1, \ldots, m - n$, where

$$A = m(m - n + 1)!\exp(\xi\Lambda\beta_n), \quad B = m(m - n + 1)!(\exp(\xi\Lambda\beta_n) - 1),$$

$$C = \frac{\gamma_n}{\Lambda\beta_n}(m - n + 1)!(\exp(\xi\Lambda\beta_n) - 1)w(\Psi(\Delta_t, \Delta_y)).$$

Since $T_0 = [0, 0]$, i.e. $w(T_0) = 0$, the inequality (4.58) follows immediately from (4.69). ■

4.3. Interval Methods of Nyström Type

Assuming that $F(T, Y)$ and $\Psi(T, Y)$ fulfill the same conditions as previously (see Sections 3.2 and 4.2), the explicit interval methods of Nyström type we define as follows [127]:

$$Y_k = Y_{k-2} + h\sum_{j=0}^{n-1} \nu_j\nabla^j F_{k-1} + h^{n+1}\left(\nu_n^{*}\Psi_n + \nu_n^{**}\Psi_n\right), \tag{4.68}$$

$$k = n, n+1, \ldots, m,$$

where $F_{k-1} = F(T_{k-1}, Y_{k-1})$, and

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

$\Psi(T, Y)$ is an interval extension of $\psi(t, y(t)) \equiv f^{(n)}(t, y(t)) \equiv y^{(n+1)}(t)$. In (4.68) it is assumed that for the integration interval $[0, \xi]$ the intervals Y_i such that $y(t_i) \in Y_i$ for $i = 0, 1, \dots, n-1$ are known, and that

$$h = \frac{\xi}{m}, \quad t_i = ih \in T_i, \quad i = 0, 1, \dots, m.$$

Let us note that in (4.68) we cannot write $\left(\nu_n^* + \nu_n^{**}\right)\Psi_n$ instead of

$$\nu_n^*\Psi_n + \nu_n^{**}\Psi_n,$$

because in general $\left|\nu_n^* + \nu_n^{**}\right|$ may be different from $\left|\nu_n^*\right| + \left|\nu_n^{**}\right|$. Moreover, the formula (4.68) can be written in more convenient form

$$Y_k = Y_{k-2} + h\sum_{j=1}^{n} \delta_{nj}F_{k-j} + h^{n+1}\left(\nu_n^*\Psi_n + \nu_n^{**}\Psi_n\right), \qquad (4.69)$$
$$k = n, n+1, \dots, m,$$

where the coefficients δ_{nj} are given by (4.28).

In particular, for a given n from (4.68) and (4.69) we have the following methods:

- $n = 1$ (the *interval midpoint rule*)

$$Y_k = Y_{k-2} + 2hF(T_{k-1}, Y_{k-1}) + \frac{h^2}{2}(\Psi_1 - \Psi_1), \qquad (4.70)$$

where

$$\Psi_1 = \Psi(T_{k-1} + [0, h], Y_{k-1} + [0, h]F(\Delta_t, \Delta_y)),$$

- $n = 2$ (in the conventional case we have the same method as for $n = 1$)

$$Y_k = Y_{k-2} + 2hF(T_{k-1}, Y_{k-1}) + \frac{h^3}{12}(5\Psi_2 - \Psi_2), \qquad (4.71)$$

where

$$\Psi_2 = \Psi(T_{k-1} + [-h, h], Y_{k-1} + [-h, h]F(\Delta_t, \Delta_y)),$$

- $n = 3$ (the interval version of the method (4.30))

$$Y_k = Y_{k-2} + \frac{h}{3}(7F(T_{k-1}, Y_{k-1}) - 2F(T_{k-2}, Y_{k-2}) + F(T_{k-3}, Y_{k-3})) + \qquad (4.72)$$

$$+\frac{h^4}{24}(9\Psi_3 - \Psi_3),$$

where

$$\Psi_3 = \Psi(T_{k-1} + [-2h, h], Y_{k-1} + [-2h, h]F(\Delta_t, \Delta_y)),$$

- $n = 4$

$$\begin{aligned} Y_k = Y_{k-2} + \frac{h}{3}(8F(T_{k-1}, Y_{k-1}) - 5F(T_{k-2}, Y_{k-2}) + 4F(T_{k-3}, Y_{k-3}) - \\ - F(T_{k-4}, Y_{k-4})) + \frac{h^5}{720}(251\Psi_4 - 19\Psi_4), \end{aligned} \tag{4.73}$$

where

$$\Psi_4 = \Psi(T_{k-1} + [-3h, h], Y_{k-1} + [-3h, h]F(\Delta_t, \Delta_y)).$$

For the explicit interval methods of Nyström type we can prove that the exact solution of the initial value problem belongs to the intervals obtained with these methods. We have

Theorem 4.3. *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* (2.1) *we have $y(t_k) \in Y_k$ for $k = n, n+1, \ldots, m$, where $Y_k = Y(t_k)$ are obtained from* (4.68).

We can also prove the following

Theorem 4.4. *If the intervals Y_k are known for $k = 0, 1, \ldots, n-1$, $t_i = ih \in T_i$ for $i = 0, 1, \ldots, m$, $h = \xi/m$, and the intervals Y_k for $k = n, n+1, \ldots, m$ are obtained from* (4.68) *or* (4.69)*, then*

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

where the nonnegative constants A, B and C are independent of h, and

$$A = m(m-n+1)!\exp(\xi\Lambda\delta_n),$$

$$B = m(m-n+1)!(\exp(\xi\Lambda\delta_n) - 1),$$

$$C = \frac{\left(\left|\nu_n^*\right| + \left|\nu_n^{**}\right|\right)}{\Lambda\delta_n}(m-n+1)!(\exp(\xi\Lambda\delta_n) - 1)w(\Psi(\Delta_t, \Delta_y)),$$

$$\delta_n = \max_{j = 1, 2, \ldots, m}\left|\delta_{nj}\right|.$$

The proofs of the theorems 4.3 and 4.4. are similar to the proofs of theorems 4.1 and 4.2, respectively.

4.4. Implicit Interval Methods of Adams-Moulton Type

4.4.1. Basic Formulas

As previously, let us denote by Δ_t and Δ_y the sets in which the function $f(t, y)$ is defined, and let $F(T, Y)$ and $\Psi(T, Y)$ denote interval extensions of $f(t, y)$ and $\overline{\psi}(t, y(t)) \equiv f^{(n+1)}(t, y(t) \equiv y^{(n+2)}(t)$, respectively (other assumptions for $F(T, Y)$ and $\overline{\Psi}(T, Y)$ are the same as for $F(T, Y)$ and $\Psi(T, Y)$ in Section 3.2). Let us assume that $y(0) \in Y_0$ and the intervals Y_i such that $y(t_i) \in Y_i$, $i = 1, 2, \ldots, n - 1$, are known. The implicit *interval n-step methods of Adams-Moulton type* can be defined as follows [79, 81, 83]:

$$\begin{aligned} Y_k = Y_{k-1} + h\sum_{j=0}^{n} \bar{\gamma}_j \nabla^j F_k +& \\ + h^{n+2}\bar{\gamma}_{n+1}\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)),& \\ k = n, n+1, \ldots, m,& \end{aligned} \tag{4.74}$$

where $h = \xi/m$, $t_i = ih \in T_i$, $i = 0, 1, \ldots, m$, $\bar{\gamma}_j$, $j = 0, 1, \ldots, n+1$, are given by (4.16), $F_k = F(T_k, Y_k)$, and

$$\nabla^j F_k = \sum_{m=0}^{j} (-1)^m \binom{j}{m} F_{k-m}. \tag{4.75}$$

Applying (4.75), the equation (4.74) can be written in the equivalent form as follows:

$$\begin{aligned} Y_k = Y_{k-1} + h\sum_{j=0}^{n} \bar{\gamma}_j F_k + h\sum_{j=1}^{n} \bar{\gamma}_j \sum_{m=1}^{j} (-1)^m \binom{j}{m} F_{k-m} +& \\ + h^{n+2}\bar{\gamma}_{n+1}\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)).& \end{aligned} \tag{4.76}$$

In particular, for a given n from (4.74) (or (4.76)) we get the following methods (see also [83]):

- $n = 1$

$$\begin{aligned} Y_k = {} & Y_{k-1} + \frac{h}{2}(2F(T_k, Y_k) - F(T_k, Y_k) + F(T_{k-1}, Y_{k-1})) - \\ & - \frac{h^3}{12}\overline{\Psi}(T_k + [-h, 0], Y_k + [-h, 0]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.77}$$

- $n = 2$

$$\begin{aligned} Y_k = {} & Y_{k-1} + \frac{h}{12}(12F(T_k, Y_k) - 7F(T_k, Y_k) + 8F(T_{k-1}, Y_{k-1}) - \\ & \qquad - F(T_{k-2}, Y_{k-2})) - \\ & - \frac{h^4}{24}\overline{\Psi}(T_k + [-2h, 0], Y_k + [-2h, 0]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.78}$$

- $n = 3$

$$\begin{aligned} Y_k = {} & Y_{k-1} + \frac{h}{24}(24F(T_k, Y_k) - 15F(T_k, Y_k) + 19F(T_{k-1}, Y_{k-1}) - \\ & \qquad - 5F(T_{k-2}, Y_{k-2}) + F(T_{k-3}, Y_{k-3})) - \\ & - \frac{19h^5}{720}\overline{\Psi}(T_k + [-3h, 0], Y_k + [-3h, 0]F(\Delta_t, \Delta_y)). \end{aligned} \tag{4.79}$$

Let us recall that in interval arithmetic the distribution law is not generally satisfied. However, since intervals are also the sets, the subdistributive law holds (see Section 1.1 for details). But this means that the values of the interval extensions of f in the above formulas with the same indices cannot be subtracted.

In real arithmetic we have

$$\sum_{j=0}^{n} \overline{\beta}_{nj} f_{k-j} = \sum_{j=0}^{n} \bar{\gamma}_j \nabla^j f_k,$$

where $f_{k-j} = f(t_{k-j}, y(t_{k-j}))$, $j = 0, 1, \ldots, n$. Hence, the formula (4.17) is equivalent to (4.19). But in interval arithmetic we have

$$\sum_{j=0}^{n} \overline{\beta}_{nj} F_{k-j} \subset \sum_{j=0}^{n} \bar{\gamma}_j \nabla^j F_k, \tag{4.80}$$

where the subset relation ($\subset$) is defined as not necessarily proper, and we get another kind of implicit interval methods corresponding to the conventional formula (4.19), namely

$$\begin{aligned} Y_k = Y_{k-1} + h\overline{\beta}_{n0}F_k + h\sum_{j=1}^{n} \overline{\beta}_{nj}F_{k-j} + \\ + h^{n+2}\overline{\gamma}_{n+1}\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)), \\ k = n, n+1, \ldots, m, \end{aligned} \tag{4.81}$$

where $h = \xi/m$, $t_i = h \in T_i$, $i = 0, 1, \ldots, m$, and $\overline{\beta}_{nj}$, $j = 0, 1, \ldots, n$, are given by (4.18).

For a given n from (4.81) we get the following methods (see also [83]):

- $n = 1$ (the *interval trapezoidal rule*)

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{2}(F(T_k, Y_k) + F(T_{k-1}, Y_{k-1})) - \\ - \frac{h^3}{12}\overline{\Psi}(T_k + [-h, 0], Y_k + [-h, 0]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.82}$$

- $n = 2$ (the interval version of the method (4.21))

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{12}(5F(T_k, Y_k) + 8F(T_{k-1}, Y_{k-1}) - F(T_{k-2}, Y_{k-2})) - \\ - \frac{h^4}{24}\overline{\Psi}(T_k + [-2h, 0], Y_k + [-2h, 0]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.83}$$

- $n = 3$ (the interval version of the method (4.22))

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{24}(9F(T_k, Y_k) + 19F(T_{k-1}, Y_{k-1}) - 5F(T_{k-2}, Y_{k-2}) + \\ + F(T_{k-3}, Y_{k-3})) - \\ - \frac{19h^5}{720}\overline{\Psi}(T_k + [-3h, 0], Y_k + [-3h, 0]F(\Delta_t, \Delta_y)). \end{aligned} \tag{4.84}$$

If we denote by Y_k^1 the interval solutions obtained from the formula (4.74) (or (4.76)), i.e. from the formula with backward interval differences, and by Y_k^2 the interval solutions obtained from (4.81), then we can prove

Theorem 4.5. $Y_k^2 \subset Y_k^1$,

which means that the second kind of implicit interval formula gives the interval solution with a smaller width, i.e. it is better. The proof of the Theorem 4.5 follows immediately from (4.80).

Let us note that (4.74) (or (4.76)) and (4.81) are nonlinear interval equations with respect to Y_k, $k = n, n + 1, \ldots, m$. This implies that in each step of implicit interval methods we have to solve an interval equation of the form

$$X = G(T, X),$$

where

$$T \in \mathbf{I}\varDelta_t \subset \mathbf{IR}, \quad X = (X_1, X_2, \ldots, X_N)^{\mathrm{T}} \in \mathbf{I}\varDelta_y \subset \mathbf{IR}^N,$$
$$G\colon \mathbf{I}\varDelta_t \times \mathbf{I}\varDelta_y \to \mathbf{IR}^N.$$

If G is a contracting mapping, then we can apply the iteration (3.54), which for the interval methods of Adams-Moulton type given by (4.76) is of the form

$$\begin{aligned} Y_k^{(l+1)} &= Y_k + \sum_{j=0}^{n} \bar{\gamma}_j F(T_k, Y_k^{(l)}) + \\ &+ h\sum_{j=1}^{n} \bar{\gamma}_j \sum_{m=1}^{j} (-1)^m \binom{j}{m} F(T_{k-m}, Y_{k-m}) + \\ &+ h^{n+2}\bar{\gamma}_{n+1}\overline{\varPsi}(T_k + [-nh, 0], Y_k^{(l)} + [-nh, 0]F(\varDelta_t, \varDelta_y)), \\ &\quad l = 0, 1, \ldots, \quad k = n, n+1, \ldots, m. \end{aligned} \tag{4.85}$$

For the interval methods given by (4.81) we have the following process:

$$\begin{aligned} Y_k^{(l+1)} &= Y_{k-1} + h\overline{\beta}_{n0} F(T_k, Y_k^{(l)}) + h\sum_{j=1}^{n} \overline{\beta}_{nj} F(T_{k-j}, Y_{k-j}) + \\ &+ h^{n+2}\bar{\gamma}_{n+1}\overline{\varPsi}(T_k + [-nh, 0], Y_k^{(l)} + [-nh, 0]F(\varDelta_t, \varDelta_y)), \\ &\quad l = 0, 1, \ldots, \quad k = n, n+1, \ldots, m. \end{aligned} \tag{4.86}$$

In (4.85) and (4.86) we usually choose $Y_k^{(0)} = Y_{k-1}$.

4.4.2. The Exact Solution vs. Interval Solutions

For the methods of the form (4.74) we can prove that the exact solution of the initial value problem (2.1) belongs to the intervals obtained by these methods [79] (very similar considerations can be carried out for the methods (4.81) and therefore we omit it). Before that it is convenient to present the following

Lemma 4.2. *If* $(t_i, y(t_i)) \in (T_i, Y_i)$ *for* $i = k - n, k - n + 1, \ldots, k - 1$, *where* $Y_i = Y(t_i)$, *then for any* $j = 0, 1, \ldots, n$ *we have*

$$\nabla^j f(t_k, y(t_k)) \in \nabla^j F(T_k, Y_k). \tag{4.87}$$

Proof. Since $F(T, Y)$ is an interval extension of $f(t, y)$, then $f(t, y) \in F(T, Y)$ for each $t \in \Delta_t$ and for each $y \in \Delta_y$. This fact implies that $(t_k, y(t_k)) \in (T_k, Y_k)$, where $Y_k \subset \Delta_y$. Moreover, $(t_i, y(t_i)) \in (T_i, Y_i)$ for $i = k - n, k - n + 1, \ldots, k - 1$, and hence we get the inclusion as follows:

$$f(t_{k-m}, y(t_{k-m})) \in F(T_{k-m}, Y_{k-m}), \quad m = 0, 1, \ldots, j.$$

This implies that

$$\sum_{m=0}^{j} (-1)^m \binom{j}{m} f(t_{k-m}, y(t_{k-m})) \in \sum_{m=0}^{j} (-1)^m \binom{j}{m} F(T_{k-m}, Y_{k-m}) =$$

$$= \nabla^j F(T_k, Y_k).$$

From (4.15) and the above relations, the inclusion (4.87) is self evident. ■

Theorem 4.6. *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* (2.1) *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* (4.74).

Proof. Let us consider the formula (4.74) for $k = n$. We get

$$y(t_n) = y(t_{n-1}) + h \sum_{j=0}^{n} \bar{\gamma}_j \nabla^j f(t_n, y(t_n)) + h^{n+2} \bar{\gamma}_{n+1} \bar{\psi}(\eta, y(\eta)), \tag{4.88}$$

where $\eta \in [t_0, t_n]$. From the assumption we have $y(t_{n-1}) \in Y_{n-1}$, and from the Lemma 4.2 it follows that

$$h \sum_{j=0}^{n} \bar{\gamma}_j \nabla^j f(t_n, y(t_n)) \in h \sum_{j=0}^{n} \bar{\gamma}_j \nabla^j F(T_n, Y_n).$$

From the Taylor formula we have

$$y(\eta) = y(t_n) + y'(t_n + \vartheta(\eta - t_n))(\eta - t_n), \tag{4.89}$$

where $\vartheta \in [0, 1]$. Because $\eta \in [t_0, t_n]$ and $t_i = ih$ for $i = 0, 1, \ldots, m$, we get

$$\eta - t_n \in [-nh, 0]. \tag{4.90}$$

Moreover, $y'(t) = f(t, y(t))$. Since

$$f(t_n + \vartheta(\eta - t_n), y(t_n + \vartheta(\eta - t_n))) \in F(\Delta_t, \Delta_y),$$

then

$$y'(t_n + \vartheta(\eta - t_n)) \in F(\varDelta_t, \varDelta_y).$$

In addition, $F(T, Y)$ is an interval extension of $f(t, y)$, and hence $f(t, y) \in F(T, Y)$ for each $t \in \varDelta_t$ and $y \in \varDelta_y$. For these reasons we can state that $y(t_n) \in Y_n$, where $Y_n \subset \varDelta_y$. Taking into account the above considerations, from the formula (4.89) we get

$$y(\eta) \in Y_n\_[-nh, 0]F(\varDelta_t, \varDelta_y). \tag{4.91}$$

Since $\overline{\Psi}$ is an interval extension of $\overline{\psi}$, then applying (4.90) and (4.91) we have

$$h^{n+2}\bar{\gamma}_{n+1}\overline{\psi}(\eta, y(\eta)) \in h^{n+2}\bar{\gamma}_{n+1}\overline{\Psi}(T_n + [-nh, 0], Y_n + [-nh, 0]F(\varDelta_t, \varDelta_y)).$$

Thus, we have shown that $y(t_k)$ belongs to the interval

$$Y_{n-1} + h\sum_{j=0}^{n} \bar{\gamma}_j \nabla^j F(T_n, Y_n) + h^{n+2}\bar{\gamma}_{n+1}\overline{\Psi}(T_n + [-nh, 0], Y_n + [-nh, 0]F(\varDelta_t, \varDelta_y)),$$

which, in fact, is the interval Y_n (see (4.74)). This conclusion ends the proof for $k = n$. In a similar way it is possible to show the thesis of this theorem for $k = n + 1$, $n + 2, \ldots, m$. ■

4.4.3. Estimations of the Widths of Interval Solutions

For the implicit interval methods of Adams-Moulton type in the form (4.81) we prove a theorem which estimates the widths of interval solutions [79] (for the methods of the form (4.74) the proof is similar).

Theorem 4.7. *If the intervals Y_k for $k = 0, 1, \ldots, n - 1$ are known, $t_i = ih \in T_i$, $i = 0, 1, \ldots, m$,*

$$h = \frac{\xi}{m}, \quad 0 < h \le h_0,$$

where

$$h_0 < \frac{1}{\varLambda\bar{\beta}_n}, \quad \bar{\beta}_n = \max_{j = 0, 1, \ldots, n} \left|\bar{\beta}_{nj}\right|,$$

and Y_k for $k = n, n + 1, \ldots, m$ are obtained from (4.81), *then*

$$w(Y_k) \le A \max_{q = 0, 1, \ldots, n-1} w(Y_q) + B \max_{j = 1, 2, \ldots, m} w(T_j) + Ch^{n+1}, \tag{4.92}$$

where the nonnegative constants A, B and C are independent of h.

Proof. From (4.81) we have

$$w(Y_k) \le w(Y_{k-1}) + h\sum_{j=0}^{n} \left|\overline{\beta}_{nj}\right| w(F(T_{k-j}, Y_{k-j})) + \\ + h^{n+2}\left|\overline{\gamma}_{n+1}\right| w\Big(\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)\Big). \tag{4.93}$$

The function $\overline{\Psi}$ is monotonic with respect to inclusion on the basis of an assumption, and if the step size h is such that

$$T_k + [-nh, 0] \subset \Delta_t, \\ Y_k + [-nh, 0]F(\Delta_t, \Delta_y) \subset \Delta_y,$$

then

$$\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)) \subset \overline{\Psi}(\Delta_t, \Delta_y).$$

From the above inclusion it follows that

$$w\Big(\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y))\Big) \le w\Big(\overline{\Psi}(\Delta_t, \Delta_y)\Big).$$

On the basis of an assumption given in Section 3.2, there exists a constant $\Lambda > 0$ such that

$$w(F(T_{k-j}, Y_{k-j})) \le \Lambda(w(T_{k-j}) + w(Y_{k-j})).$$

Thus, from (4.93) we obtain

$$w(Y_k) \le w(Y_{k-1}) + h\Lambda\overline{\beta}_n \sum_{j=0}^{n} (w(T_{k-j}) + w(Y_{k-j})) + \\ + h^{n+2}\left|\overline{\gamma}_{n+1}\right| w\Big(\overline{\Psi}(\Delta_t, \Delta_y)\Big), \tag{4.94}$$

where

$$\overline{\beta}_n = \max_{j=0,1,\ldots,n} \left|\overline{\beta}_{nj}\right|.$$

Let us denote

$$\overline{\beta} = h\Lambda\overline{\beta}_n, \quad \overline{\alpha} = 1 + \overline{\beta}, \quad \overline{\gamma} = h^{n+2}\left|\overline{\gamma}_{n+1}\right|. \tag{4.95}$$

Then we can write the inequality (4.94) as follows:

$$w(Y_k) \le w(Y_{k-1}) + \bar{\beta} w(Y_k) + \bar{\beta}\sum_{j=1}^{n} w(Y_{k-j}) + \bar{\beta}\sum_{j=1}^{n} w(T_{k-j}) + \bar{\gamma} w\Big(\overline{\Psi}(\varDelta_t, \varDelta_y)\Big),$$

that is equivalent to

$$\Big(1-\bar{\beta}\Big) w(Y_k) \le \bar{\alpha}\sum_{j=1}^{n} w(Y_{k-j}) + \bar{\beta}\sum_{j=1}^{n} w(T_{k-j}) + \bar{\gamma} w\Big(\overline{\Psi}(\varDelta_t, \varDelta_y)\Big). \tag{4.96}$$

Let us assume that

$$1 - \bar{\beta} = 1 - h\varLambda\bar{\beta}_n > 0. \tag{4.97}$$

The condition (4.97) is satisfied if $0 < h \le h_0$, where

$$h_0 < \frac{1}{\varLambda\bar{\beta}_n}.$$

On the basis of the above assumption the inequality (4.96) can be written in the form

$$w(Y_k) \le \bar{\nu}\bar{\alpha}\sum_{j=1}^{n} w(Y_{k-j}) + \bar{\nu}\bar{\beta}\sum_{j=1}^{n} w(T_{k-j}) + \bar{\nu}\bar{\gamma}\Big(\overline{\Psi}(\varDelta_t, \varDelta_y)\Big), \tag{4.98}$$

where

$$\bar{\nu} = \frac{1}{1 - h_0\varLambda\bar{\beta}_n}.$$

From (4.98) for $k = n$ we have

$$w(Y_n) \le \bar{\nu}\bar{\alpha}\sum_{j=1}^{n} w(Y_{n-j}) + \bar{\nu}\bar{\beta}\sum_{j=1}^{n} w(T_{n-j}) + \bar{\nu}\bar{\gamma} w\Big(\overline{\Psi}(\varDelta_t, \varDelta_y)\Big), \tag{4.99}$$

and for $k = n + 1$ we get

$$w(Y_{n+1}) \le \bar{\nu}\bar{\alpha} w(Y_n) + \bar{\nu}\bar{\alpha}\sum_{j=1}^{n} w(Y_{n-j}) + \bar{\nu}\bar{\beta}\sum_{j=1}^{n} w(T_{n-j}) + \bar{\nu}\bar{\gamma}\Big(\overline{\Psi}(\varDelta_t, \varDelta_y)\Big).$$

Applying (4.99) to the above inequality we obtain

$$\begin{aligned} w(Y_{n+1}) \le \Big(\big(\bar{\nu}\bar{\alpha}\big)^2 + \bar{\nu}\bar{\alpha}\Big)\sum_{j=1}^{n} w(Y_{n-j}) + \\ + \bar{\nu}\bar{\beta}\left(\bar{\nu}\bar{\alpha}\sum_{j=0}^{n} w(T_{n-j}) + \sum_{j=0}^{n} w(T_{n+1-j})\right) + \end{aligned} \tag{4.100}$$

$$+\,\overline{\nu}\overline{\gamma}\left(\overline{\nu}\overline{\alpha}+1\right)w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).$$

From (4.98) for $k = n + 2$ we get

$$w(Y_{n+2}) \le \overline{\nu}\overline{\alpha}w(Y_{n+1}) + \overline{\nu}\overline{\alpha}w(Y_n) + \overline{\nu}\overline{\alpha}\sum_{j=1}^{n-2} w(Y_{n-j}) +$$
$$+\,\overline{\nu}\overline{\beta}\sum_{j=0}^{n} w(T_{n+2-j}) + \overline{\nu}\overline{\gamma}w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).$$

Insertion of (4.99) and (4.100) into the above inequality yields

$$w(Y_{n+2}) \le \left(\left(\overline{\nu}\overline{\alpha}\right)^3 + 2\left(\overline{\nu}\overline{\alpha}\right)^2 + \overline{\nu}\overline{\alpha}\right)\sum_{j=1}^{n} w(Y_{n-j}) +$$
$$+\,\overline{\nu}\overline{\beta}\left(\left(\left(\overline{\nu}\overline{\alpha}\right)^2 + \overline{\nu}\overline{\alpha}\right)\sum_{j=0}^{n} w(T_{n-j}) + \overline{\nu}\overline{\alpha}\sum_{j=0}^{n} w(T_{n+1-j}) + \right. \tag{4.101}$$
$$\left. +\sum_{j=0}^{n} w(T_{n+2-j})\right) + \overline{\nu}\overline{\gamma}\left(\left(\overline{\nu}\overline{\alpha}\right)^2 + 2\overline{\nu}\overline{\alpha} + 1\right)w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).$$

Now, from (4.98) for $k = n + 3$ we get

$$w(Y_{n+3}) \le \overline{\nu}\overline{\alpha}w(Y_{n+2}) + \overline{\nu}\overline{\alpha}w(Y_{n+1}) + \overline{\nu}\overline{\alpha}w(Y_n) + \overline{\nu}\overline{\alpha}\sum_{j=1}^{n-3} w(Y_{n-j}) +$$
$$+\,\overline{\nu}\overline{\beta}\sum_{j=0}^{n} w(T_{n+3-j}) + \overline{\nu}\overline{\gamma}w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).$$

Applying (4.99), (4.100) and (4.101) to the above formula we obtain

$$w(Y_{n+3}) \le \left(\left(\overline{\nu}\overline{\alpha}\right)^4 + 3\left(\overline{\nu}\overline{\alpha}\right)^3 + 3\left(\overline{\nu}\overline{\alpha}\right)^2 + \overline{\nu}\overline{\alpha}\right)\sum_{j=1}^{n} w(Y_{n-j}) +$$
$$+\,\overline{\nu}\overline{\beta}\left(\left(\left(\overline{\nu}\overline{\alpha}\right)^3 + 2\left(\overline{\nu}\overline{\alpha}\right)^2 + \overline{\nu}\overline{\alpha}\right)\sum_{j=0}^{n} w(T_{n-j}) + \right.$$

$$\begin{aligned}
&+\left(\left(\overline{v\alpha}\right)^2+\overline{v\alpha}\right)\sum_{j=0}^{n} w(T_{n+1-j})+\\
&+\left.\overline{v\alpha}\sum_{j=0}^{n} w(T_{n+2-j})+\sum_{j=0}^{n} w(T_{n+3-j})\right)+\\
&+\overline{v\gamma}\left(\left(\overline{v\alpha}\right)^3+3\left(\overline{v\alpha}\right)^2+3\overline{v\alpha}+1\right)w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).
\end{aligned} \tag{4.102}$$

Thus, we see that for each $i = 0, 1, \ldots, m - n$ we have

$$\begin{aligned}
w(Y_{n+i}) \le &\left(\sum_{l=0}^{i}\binom{i}{l}\left(\overline{v\alpha}\right)^{l+1}\right)\left(\sum_{j=1}^{n} w(Y_{n-j})\right)+\\
&+\overline{v\beta}\sum_{p=0}^{l}\left(\sum_{l=0}^{p-1}\binom{p-1}{l}\left(\overline{v\alpha}\right)^{l+1}\right)\left(\sum_{j=0}^{n} w(T_{n+i-p-j})\right)+\\
&+\overline{v\gamma}\sum_{l=0}^{i}\binom{i}{l}\left(\overline{v\alpha}\right)^{l} w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).
\end{aligned}$$

Applying the notation (4.95) we get

$$\begin{aligned}
w(Y_{n+i}) \le\ & n\sum_{l=0}^{i}\binom{i}{l}\left[\bar{v}\left(1+h\Lambda\bar{\beta}_n\right)\right]^{l+1}\max_{q=0,1,\ldots,n-1} w(Y_q)+\\
&+\bar{v}h\Lambda\bar{\beta}_n(n+1)\sum_{p=0}^{i}\sum_{l=0}^{p-1}\binom{p-1}{l}\left[\bar{v}\left(1+h\Lambda\bar{\beta}_n\right)\right]^{l+1}\times\\
&\times\max_{j=0,1,\ldots,n+i} w(T_j)+\\
&+\bar{v}h^{n+2}\left|\bar{\gamma}_{n+1}\right|\sum_{l=0}^{i}\binom{i}{l}\left[\bar{v}\left(1+h\Lambda\bar{\beta}_n\right)\right]^{l} w\left(\overline{\Psi}(\Delta_t,\Delta_y)\right).
\end{aligned} \tag{4.103}$$

The following estimations are true:

$$\binom{i}{l} \le (m-n)! \quad \text{for} \quad l = 0, 1, \ldots, i,$$

$$\binom{p-1}{l} \le (m-n)! \quad \text{for } \; l = 0, 1, \dots, p-1,$$

and

$$\left(1 + h\Lambda\bar{\beta}_n\right) \le \exp\left(\xi\Lambda\bar{\beta}_n\right),$$

$$\sum_{l=0}^{p-1} \bar{\nu}^{l+1}\left(1 + h\Lambda\bar{\beta}_n\right)^{l+1} \le \frac{\bar{\nu}^m \exp\left(\xi\Lambda\bar{\beta}_n\right) - 1}{\bar{\nu}h\Lambda\bar{\beta}_n}.$$

From the above estimations it follows that (see [79] for details)

$$n\sum_{l=0}^{i} \binom{i}{l}\left[\bar{\nu}\left(1 + h\Lambda\bar{\beta}_n\right)\right]^{l+1} \le m(m-n)!\,\bar{\nu}\exp\left(\xi\Lambda\bar{\beta}_n\right)\frac{1-\bar{\nu}^m}{1-\bar{\nu}},$$

$$(n+1)\sum_{p=0}^{i}\sum_{l=0}^{p-1}\binom{p-1}{l}\left[\bar{\nu}\left(1 + h\Lambda\bar{\beta}_n\right)\right]^{l+1} \le (m+1)(m-n+1)!\frac{\bar{\nu}\exp\left(\xi\Lambda\bar{\beta}_n\right) - 1}{\bar{\nu}h\Lambda\bar{\beta}_n},$$

$$\sum_{l=0}^{i}\binom{i}{l}\left[\bar{\nu}\left(1 + h\Lambda\bar{\beta}_n\right)\right]^{l} \le (m-n)!\frac{\bar{\nu}\exp\left(\xi\Lambda\bar{\beta}_n\right) - 1}{\bar{\nu}h\Lambda\bar{\beta}_n}.$$

Thus, from (4.103) we finally get

$$w(Y_{n+i}) \le A \max_{q=0,1,\dots,n-1} w(Y_q) + B \max_{j=0,1,\dots,m} w(T_j) + Ch^{n+1} \tag{4.104}$$

for each $i = 0, 1, \dots, m - n$, where

$$A = m(m-n)!\,\bar{\nu}\exp\left(\xi\Lambda\bar{\beta}_n\right)\frac{1-\bar{\nu}^m}{1-\bar{\nu}},$$

$$B = (m+1)(m-n+1)!\left(\bar{\nu}^m\exp\left(\xi\Lambda\bar{\beta}_n\right) - 1\right),$$

$$C = \frac{\left|\bar{\gamma}_{n+1}\right|}{\Lambda\bar{\beta}_n}(m-n)!\left(\bar{\nu}^m\exp\left(\xi\Lambda\bar{\beta}_n\right) - 1\right)w\left(\Psi(\Delta_t, \Delta_y)\right).$$

Taking into account that $T_0 = [0, 0]$, the inequality (4.93) is self evident. ■

4.5. Implicit Interval Methods of Milne-Simpson Type

As previously, let $\bar{\Psi}(T, Y)$ denote the interval extension of

$$\bar{\psi}(t, y) \equiv f^{(n+1)}(t, y) \equiv y^{(n+2)}(t),$$

and let us assume that $\overline{\Psi}(T, Y)$ is monotonic with respect to inclusion and determined for all $T \subset \varDelta_t$ and $Y \subset \varDelta_y$. Using (4.32) we get the following implicit interval methods [127]:

$$Y_k = Y_{k-2} + h\sum_{j=0}^{n} \overline{v}_j \nabla^j F_k + h^{n+2}\left(\overline{v}^*_{n+1}\overline{\Psi}_n + \overline{v}^{**}_{n+1}\overline{\Psi}_n\right), \qquad k = n, n+1, \ldots, m, \tag{4.105}$$

where $F_k = F(T_k, Y_k)$, and where

$$\overline{\Psi}_n = \overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\varDelta_t, \varDelta_y)).$$

The second kind of *interval methods of Milne-Simpson type*, based on (4.33), are as follows [127]:

$$Y_k = Y_{k-2} + h\sum_{j=0}^{n} \overline{\delta}_{nj} F_{k-j} + h^{n+2}\left(\overline{v}^*_{n+1}\overline{\Psi}_n + \overline{v}^{**}_{n+1}\overline{\Psi}_n\right), \qquad k = n, n+1, \ldots, m. \tag{4.106}$$

In particular, for a given n from (4.105) we get the following methods of the first kind:

- $n = 1$

$$\begin{aligned} Y_k = Y_{k-2} &+ 2h(F(T_k, Y_k) - F(T_k, Y_k) + F(T_{k-1}, Y_{k-1})) + \\ &+ \frac{h^3}{12}\left(5\overline{\Psi}_1 - \overline{\Psi}_1\right), \end{aligned} \tag{4.107}$$

where

$$\overline{\Psi}_1 = \overline{\Psi}(T_k + [-h, 0], Y_k + [-h, 0]F(\varDelta_t, \varDelta_y)),$$

- $n = 2$

$$\begin{aligned} Y_k = Y_{k-2} &+ \frac{h}{3}(7F(T_k, Y_k) - 6F(T_k, Y_k) + 6F(T_{k-1}, Y_{k-1}) - \\ &\qquad - 2F(T_{k-1}, Y_{k-1}) + F(T_{k-2}, Y_{k-2})) + \\ &+ \frac{h^4}{24}\left(\overline{\Psi}_2 - \overline{\Psi}_2\right), \end{aligned} \tag{4.108}$$

where

$$\overline{\Psi}_2 = \overline{\Psi}(T_k + [-2h, 0], Y_k + [-2h, 0]F(\varDelta_t, \varDelta_y)),$$

- $n = 3$ (in the conventional case we have the same method as for $n = 2$ – see (4.37))

$$\begin{aligned} Y_k = Y_{k-2} + \frac{h}{3}(7F(T_k, Y_k) - 6F(T_k, Y_k) + 6F(T_{k-1}, Y_{k-1}) - \\ - 2F(T_{k-1}, Y_{k-1}) + F(T_{k-2}, Y_{k-2}) + \\ + \frac{h^5}{720}\left(11\overline{\Psi}_3 - 19\overline{\Psi}_3\right), \end{aligned} \tag{4.109}$$

where

$$\overline{\Psi}_3 = \overline{\Psi}(T_k + [-3h, 0], Y_k + [-3h, 0]F(\Delta_t, \Delta_y)).$$

Below are examples of the implicit methods of the second kind (obtained from (4.106)).

- $n = 1$

$$Y_k = Y_{k-2} + 2hF(T_{k-1}, Y_{k-1}) + \frac{h^3}{12}\left(5\overline{\Psi}_1 - \overline{\Psi}_1\right), \tag{4.110}$$

- $n = 2$ (the interval version of Milne’s method (4.37))

$$\begin{aligned} Y_k = Y_{k-2} + \frac{h}{3}(F(T_k, Y_k) + 4F(T_{k-1}, Y_{k-1}) + F(T_{k-2}, Y_{k-2})) + \\ + \frac{h^4}{24}\left(\overline{\Psi}_2 - \overline{\Psi}_2\right), \end{aligned} \tag{4.111}$$

- $n = 3$

$$\begin{aligned} Y_k = Y_{k-2} + \frac{h}{3}(F(T_k, Y_k) + 4F(T_{k-1}, Y_{k-1}) + F(T_{k-2}, Y_{k-2})) + \\ + \frac{h^5}{720}\left(11\overline{\Psi}_3 - 19\overline{\Psi}_3\right). \end{aligned} \tag{4.112}$$

If we denote:

Y_k^1 – the interval solution obtained from (4.105), i.e. from the formula with backward interval differences,

Y_k^2 – the interval solution obtained from (4.106), i.e. from the formula without backward interval differences,

then we can prove

Theorem 4.8. $Y_k^2 \subset Y_k^1$.

The proof of the above theorem follows immediately from the inclusion

$$\sum_{j=0}^{n} \overline{\delta}_{nj} F_{k-j} \subset \sum_{j=0}^{n} \overline{v}_j \nabla^j F_k.$$

Let us note that we can get only one kind of explicit interval methods of Nyström type (see Section 4.3). It follows from the fact that for these methods we have

$$\sum_{j=1}^{n} \delta_{nj} F_{k-j} = \sum_{j=0}^{n-1} \nu_j \nabla^j F_{k-1},$$

because in explicit methods all coefficients ν_j are nonnegative.

In each step of the interval methods of Milne-Simpson type (of both kinds) we have to solve a system of nonlinear interval equations. If the right-hand sides of (4.105) and (4.106) are contracting mapping, then the iteration follows immediately from the well-known fixed-point theorem. For the second kind of interval methods of Milne-Simpson type, i.e. for (4.106), the iteration is as follows:

$$\begin{aligned} Y_k^{(l+1)} &= Y_{k-2} + h\bar{\delta}_{n0} F(T_k, Y_k^{(l)}) + h\sum_{j=1}^{n} \bar{\delta}_{nj} F(T_{k-j}, Y_{k-j}) + \\ &+ h^{n+2}\left(\bar{\nu}_{n+1}^{*}\overline{\Psi}_n^{(l)} + \bar{\nu}_{n+1}^{**}\overline{\Psi}_n^{(l)}\right), \\ &l = 0, 1, \ldots, \quad k = n, n+1, \ldots, m, \end{aligned} \tag{4.113}$$

where

$$\overline{\Psi}_n^{(l)} = \overline{\Psi}(T_k + [-nh, 0], Y_k^{(l)} + [-nh, 0]F(\Delta_t, \Delta_y)),$$

and we can choose $Y_k^{(0)} = Y_{k-1}$.

As for other interval methods, for the implicit interval methods of Milne-Simpson type we can prove that the exact solution of the initial value problem (2.1) belongs to the interval solutions obtained. We have [127]

Theorem 4.9. *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* (2.1) *we have* $y(t_k) \in Y_k$ *for* $k = n, n+1, \ldots, m$, *where* $Y_k = Y(t_k)$ *are obtained from* (4.105) *or* (4.106).

We omit the proof as it is similar to the proof of Theorem 4.6.

We can also estimate the widths of the interval solution obtained [127].

Theorem 4.10. *If the intervals* Y_k *for* $k = 0, 1, \ldots, n-1$ *are known,* $t_i = ih \in T_i$, $i = 0, 1, \ldots, m$,

$$h = \frac{\xi}{m}, \quad 0 < h \le h_0,$$

where

$$h_0 < \frac{1}{\Lambda\bar{\delta}_n}, \quad \bar{\delta}_n = \max_{j=0,1,\ldots,n} \left|\bar{\delta}_{nj}\right|,$$

and Y_k *for* $k = n, n+1, \ldots, m$ *are obtained from* (4.105) *or* (4.106), *then*

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

where the nonnegative constants A, B and C are independent of h, and

$$A = m(m-n)!\,\bar{\mu}\exp\left(\xi\Lambda\bar{\delta}_n\right)\frac{1-\bar{\mu}^m}{1-\bar{\mu}},$$

$$B = (m+1)(m-n+1)!\left(\bar{\mu}^m \exp\left(\xi\Lambda\bar{\delta}_n\right) - 1\right),$$

$$C = \frac{\left|\bar{v}^{*}_{n+1}\right| + \left|\bar{v}^{**}_{n+1}\right|}{\Lambda\bar{\delta}_n}(m-n)!\left(\bar{\mu}^m \exp\left(\xi\Lambda\bar{\delta}_n\right) - 1\right) w\left(\Psi(\Delta_t, \Delta_y)\right),$$

$$\bar{\mu} = \frac{1}{1 - h_0\Lambda\bar{\delta}_n}.$$

The proof of this theorem is similar to that of Theorem 4.7 (see [127] for details).

Since the application of the implicit interval methods requires the nonlinear interval equation to be solved in each step, an initial approximation to the solution is needed. When calculating Y_k we can take Y_{k-1} as such an approximation, but a better approach is to take Y_k obtained from an explicit method. This leads to the construction of predictor-corrector methods. The first stage in each step of such methods is a prediction that computes the initial approximation by applying an explicit method, and the second stage is a correction that improves the obtained solution by employing an implicit methods. Interval predictor-corrector methods are considered in [83].

4.6. Computational Complexity of Multistep Interval Methods

In order to find the number of operations in interval methods of Adams-Bashforth it is convenient to use the relation

$$h\sum_{j=1}^{n} \beta_{nj}F_{k-j} = \frac{h}{\rho_n}\sum_{j=1}^{n} \hat{\beta}_{nj}F_{k-j},$$

where $\hat{\beta}_{nj}$ and $\rho_n > 0$ are integers (represented exactly in computer memory), and write the methods (4.48) in the form

$$Y_k = Y_{k-1} + \frac{h}{\rho_n}\sum_{j=0}^{n} \hat{\beta}_{nj} F_{k-j} +$$
$$+ h^{n+1}\gamma_n \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y)).$$

The way to determine the number of operations in one method step is presented in Figure 4.1.

Thus, in one step of any interval method of Adams-Bashforth type we need to perform $N(6n + 24) + 2$ operations and calculate the functions F_{k-j} ($j = 1, 2, \ldots, n$) and Ψ. Moreover, in the first step, i.e. for $n = k$, it is necessary to do $2n + 4$ operations (to calculate $h^{n+1}\gamma_n$), two divisions (to calculate $h = \xi/m$) and evaluate the function $F(\Delta_t, \Delta_y)$ and the expression $[-(n - 1)h, h]F(\Delta_t, \Delta_y)$. Let us note that in the first step we need to evaluate n functions F_{k-j}, but in each subsequent step, i.e. for $k = n + 1, n + 2, \ldots, m$, we need to evaluate only F_{k-1}.

All in all (for $k = n, n + 1, \ldots, m$) in any interval method of Adams-Bashforth type it is necessary to perform at most [126]

$$\begin{aligned} l(Y) = (m - n + 1)[N(6n + 24) + 2] + 8N + 2n + 7 + \\ + 8(m + 1)l(f) + 8(m - n + 1)l_n(\psi) \end{aligned} \tag{4.114}$$

operations, where $l(f)$ denotes the number of operations in evaluating $f(t, y)$, and $l_n(\psi)$ is the number of operations to find $\psi(t, y)$ in n-step method. In conventional Adams-Bashforth methods we have

$$l(y) = (m - n + 1)N(2n + 1) + 2 + m \cdot l(f). \tag{4.115}$$

Example 4.3

Let us consider the same initial value problem as in Example 3.1. In interval methods of Adams-Bashforth type we have: $N = 1$, $l(f) = 1$, $l_1(\psi) = 3$, $l_2(\psi) = 12$, $l_3(\psi) = 37$, $l_4(\psi) = 114$. In Figure 4.2 we present the ratios $l(Y)/l(y)$ as the functions of integration step numbers. As in Runge-Kutta methods these ratios increase together with the order of method, and the values of $l_n(\psi)$ have the decisive influence on it. ■

In the interval methods of Nyström type we have to evaluate the term

$$h^{n+1}\left(v_n^{*}\Psi_n + v_n^{**}\Psi_n\right), \tag{4.116}$$

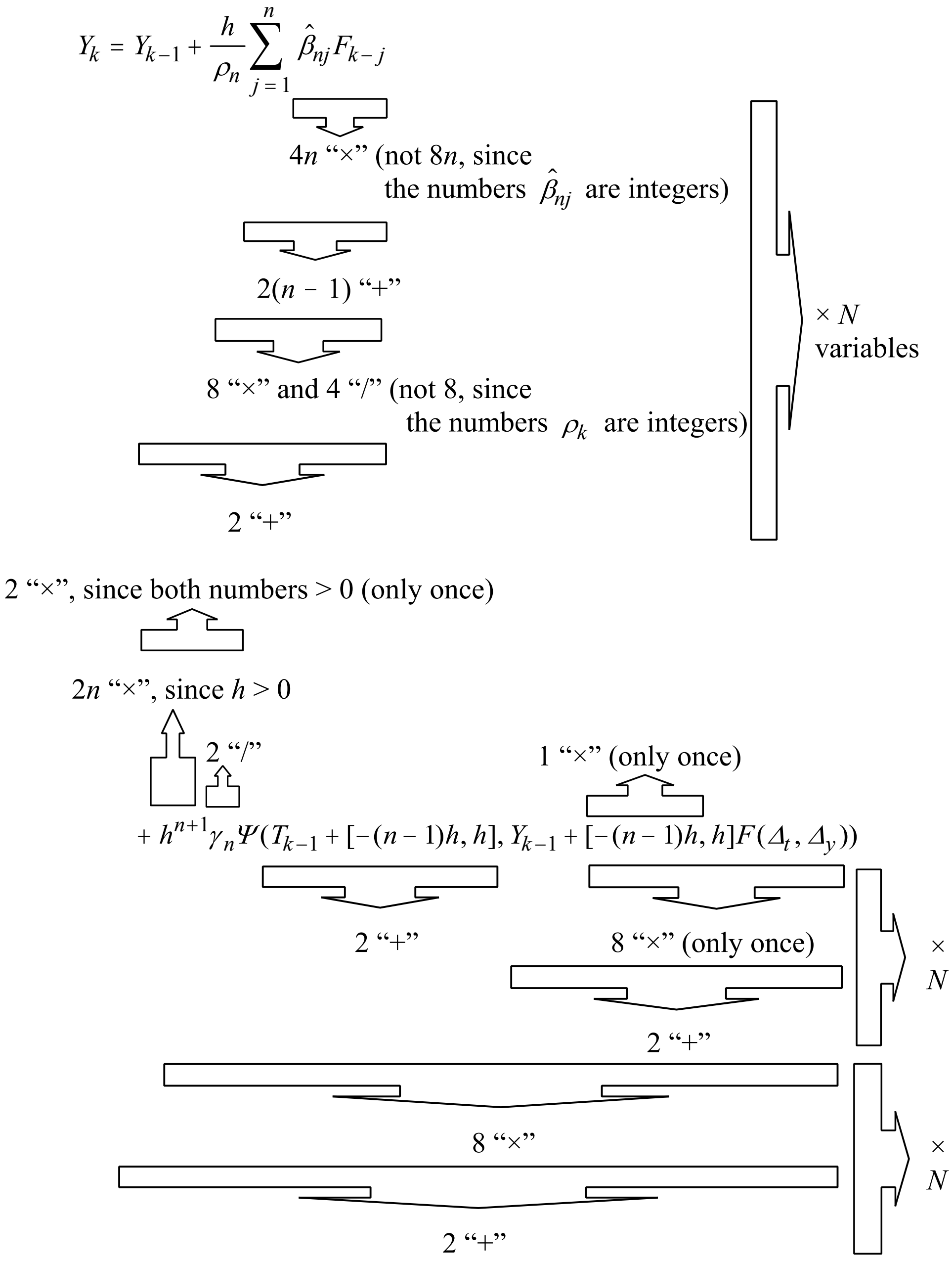

$Y_k = Y_{k-1} + \frac{h}{\rho_n}\sum_{j=1}^{n}\hat{\beta}_{nj}F_{k-j}$
$4n$ “×” (not $8n$, since the numbers $\hat{\beta}_{nj}$ are integers)
$2(n - 1)$ “+”
8 “×” and 4 “/” (not 8, since the numbers ρ_k are integers)
2 “+”
× N variables
2 “×”, since both numbers > 0 (only once)
$2n$ “×”, since $h > 0$
2 “/”
1 “×” (only once)
$+ h^{n+1}\gamma_n\Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y))$
2 “+”
8 “×” (only once)
2 “+”
× N
8 “×”
2 “+”
× N

Figure 4.1. Finding the number of operations for calculating Y_k in the interval mulistep methods of Adams-Bashforth type

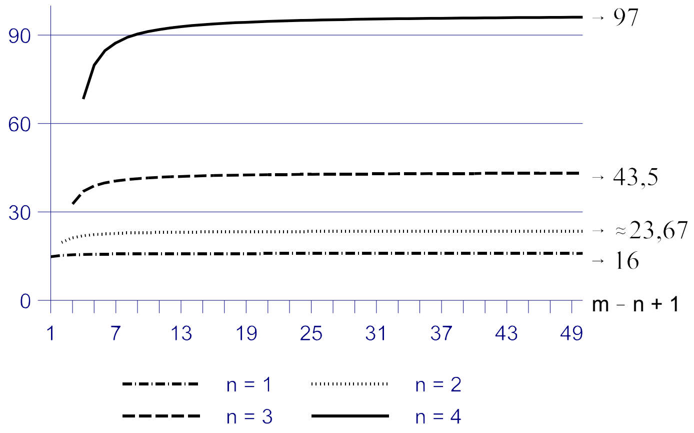


Figure 4.2. The ratios *l*(*Y*)/*l*(*y*) in Adams-Bashforth methods

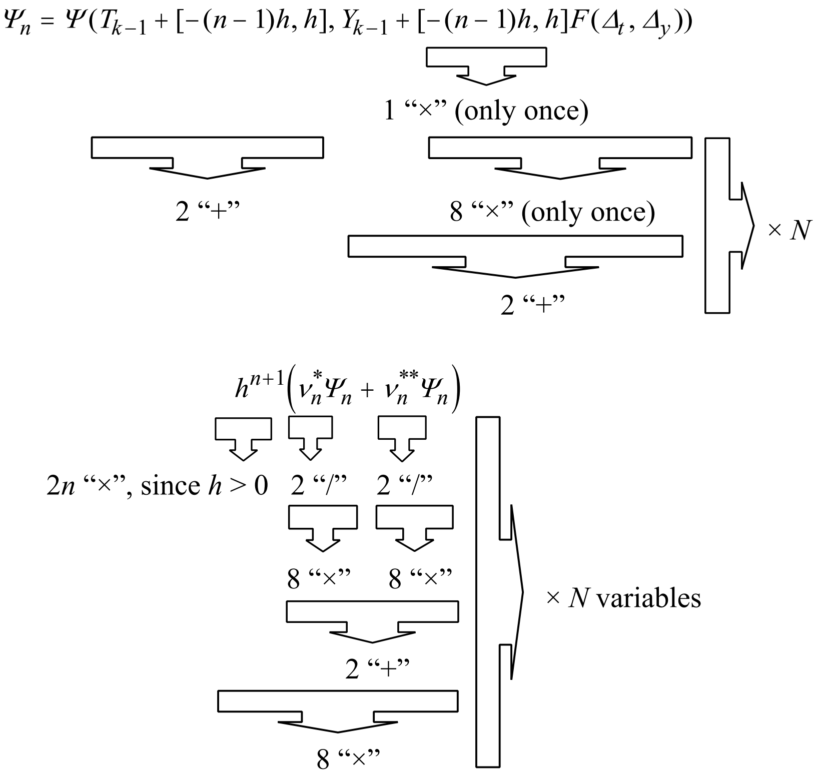

$\Psi_n = \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y))$
1 “×” (only once)
2 “+”
8 “×” (only once)
2 “+”
× N
$h^{n+1}\left(\nu_n^{*}\Psi_n + \nu_n^{**}\Psi_n\right)$
2n “×”, since $h > 0$
2 “/”
2 “/”
8 “×”
8 “×”
2 “+”
8 “×”
× N variables

Figure 4.3. Finding the number of operations for calculating (4.116)

where $\Psi_n = \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y))$, instead of the term

$$h^{n+1}\gamma_n \Psi(T_{k-1} + [-(n-1)h, h], Y_{k-1} + [-(n-1)h, h]F(\Delta_t, \Delta_y))$$

that occurs in the interval methods of Adams-Bashforth type. The number of other operations (in floating-point arithmetic) is the same in both kinds of these methods. The procedure for determining the number of operations in evaluating the term (4.116) in one step of any interval method of Nyström type is presented in Figure 4.3.

Drawing from the analysis of Figure 4.3 and taking into account other operations we conclude that the total number of operations in interval methods of Nyström type is equal to [126]

$$\begin{aligned}\hat{l}(Y) = (m-n+1)[N(6n+42)+2] + 8N + 2n + 7 + \\ + 8(m+1)l(f) + 8(m-n+1)l_n(\psi).\end{aligned} \tag{4.117}$$

In conventional Nyström methods we have the same number of operations as in conventional methods of Adams-Bashforth, i.e. this number is given by (4.115).

For interval methods of Adams-Moulton type let us consider the formula without backward differences, i.e. (4.81). In each step we have to solve a nonlinear interval equations using iteration of the form (4.86), which can be written as follows:

$$\begin{aligned} Y_k^{(l+1)} = Y_{k-1} + \frac{h}{\bar{\rho}_n}\left(\hat{\bar{\beta}}_{n0}F(T_k, Y_k^{(l)}) + \sum_{j=1}^{n}\hat{\bar{\beta}}_{nj}F(T_{k-j}, Y_{k-j})\right) + \\ + h^{n+2}\bar{\gamma}_{n+1}\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)), \\ l = 0, 1, \ldots, \end{aligned}$$

where $\hat{\bar{\beta}}_{nj}$ and $\bar{\rho}_n > 0$ are integers.

At the first integration step (for $k = n$) one should calculate constant values of the n-step method, i.e.

$$\frac{h}{\bar{\rho}_n}, \quad h^{n+2}\bar{\gamma}_{n+1}, \quad [-nh, 0]F(\Delta_t, \Delta_y),$$

which requires $4N + 2n + 7$ operations and evaluation of the function $F(\Delta_t, \Delta_y)$. At the first iteration (at each step) it is necessary to perform $6nN - 2N + 1$ operations concerning evaluations of the expressions

$$\sum_{j=1}^{n}\hat{\bar{\beta}}_{nj}F(T_{k-j}, Y_{k-j}) \text{ and } T_k + [-nh, 0].$$

Moreover, at the first iteration of the first step we need to evaluate the functions $F(T_{k-j}, Y_{k-j})$ for $j = 1, 2, \ldots, n$, and at the first iteration of each next step we have to evaluate the function $F(T_{k-1}, Y_{k-1})$. In addition, at each iteration (also at the first) it is necessary to perform $17N$ operations and evaluate the functions $F(T_k, Y_k^{(l)})$ and Ψ.

If we denote by S the maximum number of iterations, then at all steps, i.e. for $k = n, n+1, \ldots, m$, we have to perform at most

$$\begin{aligned} l^*(Y) = {} & (m-n+1)(17SN + 6nM - 2N + 1) + 4N + 2n + 7 + \\ & + 8[S(m-n+1) + m + 1]l(f) + 8(m-n+1)l_n^*(\psi) \end{aligned}$$

operations. In conventional methods of Adams-Moulton the total number of operation is equal at most to

$$l^*(y) = (m-n+1)(4SN + 2nN - 1) + 2 + [S(m-n+1) + m]l(f).$$

Example 4.4

For the initial value problem as in Example 3.1, the data as in Example 4.3 and with assumption that $S = 3$, for the methods of Adams-Moulton type we obtain

- $n = 1$

$$\lim_{m\to\infty} \frac{l^*(Y)}{l^*(y)} \approx 10.82,$$

- $n = 2$

$$\lim_{m\to\infty} \frac{l^*(Y)}{l^*(y)} \approx 15.58,$$

- $n = 3$

$$\lim_{m\to\infty} \frac{l^*(Y)}{l^*(y)} \approx 53.71.$$

We see that the limit ratios are similar to those obtained in Adams-Bashfoth methods, which for a given n give the interval solutions with one order less (compare Figure 4.2). ■

For the interval methods of Milne-Simpson type the only difference in computational complexity (in comparison with the interval methods of Adams-Moulton type) results from the fact that we should calculate the term

$$h^{n+2}\left(\overline{v}_{n+1}^{*}\overline{\Psi}_n + \overline{v}_{n+1}^{**}\overline{\Psi}_n\right),$$

where $\overline{\Psi}_n = \overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y))$, instead of the term

$$h^{n+2}\overline{\gamma}_{n+1}\overline{\Psi}(T_k + [-nh, 0], Y_k + [-nh, 0]F(\Delta_t, \Delta_y)),$$

which appears in the interval methods of Adams-Moulton type. We get the similar difference as that between the formulas (4.117) and (4.114).

4.7. Numerical Examples and a Comparison of Multistep Interval Methods

In each multistep interval method we need some starting intervals to begin calculations. Such intervals may be obtained by any one-step method. In the examples presented below the interval methods of Runge-Kutta type, presented in Chapter 3, was used to obtain such intervals.

Example 4.5

Let us consider the commonly used test problem (2.2) with λ =0.5. Let us assume that in multistep interval methods, considered in this example, the starting intervals have been obtained by the interval method (3.42) with the following data:

$$\Delta_t = [0, 10], \quad \Delta_y = \left[\underline{0.9}, 149\right], \quad h_0 = 0.001, \quad T_0 = [0, 0], \quad Y_0 = [1, 1], \quad M = 0.003.$$

Taking $h = 0.0005$ the interval method (3.42) yields the interval solutions presented in Table 4.1.

Table 4.1. Starting intervals for interval multistep methods and the problem (2.2) obtained by the interval method (3.42)

k	$t_k = kh \in T_k$	Y_k
1	0.0005	[1.0002500312526043E+0000, 1.0002500312526044E+0000]
2	0.0010	[1.0005001250208359E+0000, 1.0005001250208360E+0000]
3	0.0015	[1.0007502813203256E+0000, 1.0007502813203257E+0000]

In Table 4.2 we present the results obtained by explicit multistep interval methods of Adams-Bashforth type for different number of steps n. It can be seen that increasing n we obtain better solutions taking into account the widths of intervals. But that is true only for $n \leq 4$. If we use the interval methods of Adams-Bashforth type for $n > 4$, i.e.

Table 4.2.The solutions of the problem (2.2) obtained by interval methods of Adams-Bashforth type for *n* ≤ 4

Method	t_k	Y_k	*Width*
(4.44)	0.2	[1.1051709169246437E+0000, 1.1051709223468415E+0000]	$\approx 5.42 \cdot 10^{-9}$
$n = 1$	0.4	[1.2214027556160577E+0000, 1.2214027670307107E+0000]	$\approx 1.14 \cdot 10^{-8}$
	0.6	[1.3498588033584851E+0000, 1.3498588213958252E+0000]	$\approx 1.80 \cdot 10^{-8}$
	0.8	[1.4918246914264993E+0000, 1.4918247167830407E+0000]	$\approx 2.54 \cdot 10^{-8}$
	1.0	[1.6487212621146479E+0000, 1.6487212955601577E+0000]	$\approx 3.34 \cdot 10^{-8}$
(4.45)	0.2	[1.1051709180745339E+0000, 1.1051709180769049E+0000]	$\approx 2.37 \cdot 10^{-12}$
$n = 2$	0.4	[1.2214027581576920E+0000, 1.2214027581629653E+0000]	$\approx 5.27 \cdot 10^{-12}$
	0.6	[1.3498588075718577E+0000, 1.3498588075806753E+0000]	$\approx 8.82 \cdot 10^{-12}$
	0.8	[1.4918246976350853E+0000, 1.4918246976482317E+0000]	$\approx 1.31 \cdot 10^{-11}$
	1.0	[1.6487212706914478E+0000, 1.6487212707098811E+0000]	$\approx 1.84 \cdot 10^{-11}$
(4.46)	0.2	[1.1051709180756470E+0000, 1.1051709180756482E+0000]	$\approx 1.06 \cdot 10^{-15}$
$n = 3$	0.4	[1.2214027581601685E+0000, 1.2214027581601711E+0000]	$\approx 2.47 \cdot 10^{-15}$
	0.6	[1.3498588075760007E+0000, 1.3498588075760053E+0000]	$\approx 4.48 \cdot 10^{-15}$
	0.8	[1.4918246976412665E+0000, 1.4918246976412740E+0000]	$\approx 7.39 \cdot 10^{-15}$
	1.0	[1.6487212707001222E+0000, 1.6487212707001339E+0000]	$\approx 1.15 \cdot 10^{-14}$
(4.47)	0.2	[1.1051709180756474E+0000, 1.1051709180756478E+0000]	$\approx 2.55 \cdot 10^{-16}$
$n = 4$	0.4	[1.2214027581601695E+0000, 1.2214027581601701E+0000]	$\approx 5.57 \cdot 10^{-16}$
	0.6	[1.3498588075760025E+0000, 1.3498588075760037E+0000]	$\approx 1.14 \cdot 10^{-15}$
	0.8	[1.4918246976412691E+0000, 1.4918246976412715E+0000]	$\approx 2.29 \cdot 10^{-15}$
	1.0	[1.6487212707001259E+0000, 1.6487212707001305E+0000]	$\approx 4.51 \cdot 10^{-15}$

- $n = 5$

$$\begin{aligned} Y_k = Y_{k-1} &+ \frac{h}{720}(1901F(T_{k-1}, Y_{k-1}) - 2774F(T_{k-2}, Y_{k-2}) + \\ &\qquad + 2616F(T_{k-3}, Y_{k-3}) - 1274F(T_{k-4}, Y_{k-4}) + \\ &\qquad + 251F(T_{k-5}, Y_{k-5})) + \\ &+ \frac{95h^6}{288}\Psi(T_{k-1} + [-4h, h], Y_{k-1} + [-4h, h]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.118}$$

- $n = 6$

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{1440}&(4277F(T_{k-1}, Y_{k-1}) - 7923F(T_{k-2}, Y_{k-2}) + \\ &+ 9982F(T_{k-3}, Y_{k-3}) - 7298F(T_{k-4}, Y_{k-4}) + \\ &+ 2877F(T_{k-5}, Y_{k-5}) - 475F(T_{k-6}, Y_{k-6})) + \\ + \frac{19087h^7}{60480}&\Psi(T_{k-1} + [-5h, h], Y_{k-1} + [-5h, h]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.119}$$

- $n = 7$

$$\begin{aligned} Y_k = Y_{k-1} + \frac{h}{60480}&(198721F(T_{k-1}, Y_{k-1}) - 447288F(T_{k-2}, Y_{k-2}) + \\ &+ 705549F(T_{k-3}, Y_{k-3}) - 688256F(T_{k-4}, Y_{k-4}) + \\ &+ 407139F(T_{k-5}, Y_{k-5}) - 134472F(T_{k-6}, Y_{k-6}) + \\ &+ 19087F(T_{k-7}, Y_{k-7})) + \\ + \frac{5257h^8}{17280}&\Psi(T_{k-1} + [-6h, h], Y_{k-1} + [-6h, h]F(\Delta_t, \Delta_y)), \end{aligned} \tag{4.120}$$

then we obtain intervals with greater widths (see Table 4.3). This is caused by a great number of calculations in these methods and by a significant increase of rounding errors following from that, which is not compensated for the method orders.

Table 4.3.The solutions of the problem (2.2) obtained by interval methods of Adams-Bashforth type for *n* = 5, 6, 7

Method	t_k	Y_k	*Width*
(4.118)	0.2	[1.1051709180756474E+0000, 1.1051709180756479E+0000]	$\approx 4.19 \cdot 10^{-16}$
$n = 5$	0.4	[1.2214027581601690E+0000, 1.2214027581601706E+0000]	$\approx 1.49 \cdot 10^{-15}$
	0.6	[1.3498588075760005E+0000, 1.3498588075760057E+0000]	$\approx 5.15 \cdot 10^{-15}$
	0.8	[1.4918246976412615E+0000, 1.4918246976412791E+0000]	$\approx 1.75 \cdot 10^{-14}$
	1.0	[1.6487212707000986E+0000, 1.6487212707001578E+0000]	$\approx 5.91 \cdot 10^{-14}$
(4.119)	0.2	[1.1051709180756470E+0000, 1.1051709180756482E+0000]	$\approx 1.08 \cdot 10^{-15}$
$n = 6$	0.4	[1.2214027581601645E+0000, 1.2214027581601750E+0000]	$\approx 1.04 \cdot 10^{-14}$
	0.6	[1.3498588075759536E+0000, 1.3498588075760525E+0000]	$\approx 9.88 \cdot 10^{-14}$
	0.8	[1.4918246976408020E+0000, 1.4918246976417385E+0000]	$\approx 9.36 \cdot 10^{-13}$
	1.0	[1.6487212706956903E+0000, 1.6487212707045659E+0000]	$\approx 8.88 \cdot 10^{-12}$

Table 4.3. (cont.)

Method	t_k	Y_k	*Width*
(4.120)	0.2	[1.1051709180756443E+0000, 1.1051709180756511E+0000]	$\approx 6.68 \cdot 10^{-15}$
$n = 7$	0.4	[1.2214027581599533E+0000, 1.2214027581603865E+0000]	$\approx 4.33 \cdot 10^{-13}$
	0.6	[1.3498588075619817E+0000, 1.3498588075900247E+0000]	$\approx 2.80 \cdot 10^{-11}$
	0.8	[1.4918246967334331E+0000, 1.4918246985491077E+0000]	$\approx 1.82 \cdot 10^{-9}$
	1.0	[1.6487212119209625E+0000, 1.6487213294792939E+0000]	$\approx 1.17 \cdot 10^{-7}$

The results obtained by interval methods of Nyström type for $n = 1, 2, 3$ and 4 are presented in Table 4.4. As in the case of interval methods of Adams-Bashforth type, for $n > 4$ we get interval solutions with greater widths. It is interesting that for the problem considered and for the same n ($n > 1$), i.e. for the same number of method steps, interval methods of Nyström type give an interval solution with smaller widths.

Table 4.4.The solutions of the problem (2.2) obtained by interval methods of Nyström type

Method	t_k	Y_k	*Width*
(4.70)	0.2	[1.1051709142138796E+0000, 1.1051709196354981E+0000]	$\approx 5.42 \cdot 10^{-9}$
$n = 1$	0.4	[1.2214027499092897E+0000, 1.2214027613229209E+0000]	$\approx 1.14 \cdot 10^{-8}$
	0.6	[1.3498587943404760E+0000, 1.3498588123764994E+0000]	$\approx 1.80 \cdot 10^{-8}$
	0.8	[1.4918246787488588E+0000, 1.4918247041039496E+0000]	$\approx 2.54 \cdot 10^{-8}$
	1.0	[1.6487212453923441E+0000, 1.6487212788364476E+0000]	$\approx 3.34 \cdot 10^{-8}$
(4.71)	0.2	[1.1051709180749699E+0000, 1.1051709180763254E+0000]	$\approx 1.35 \cdot 10^{-12}$
$n = 2$	0.4	[1.2214027581587431E+0000, 1.2214027581615966E+0000]	$\approx 2.85 \cdot 10^{-12}$
	0.6	[1.3498588075737485E+0000, 1.3498588075782577E+0000]	$\approx 4.51 \cdot 10^{-12}$
	0.8	[1.4918246976381009E+0000, 1.4918246976444397E+0000]	$\approx 6.34 \cdot 10^{-12}$
	1.0	[1.6487212706959476E+0000, 1.6487212707043086E+0000]	$\approx 8.36 \cdot 10^{-12}$

Table 4.4. (cont.)

Method	t_k	Y_k	*Width*
(4.72)	0.2	[1.1051709180756473E+0000, 1.1051709180756480E+0000]	$\approx 5.78 \cdot 10^{-16}$
$n = 3$	0.4	[1.2214027581601692E+0000, 1.2214027581601704E+0000]	$\approx 1.14 \cdot 10^{-15}$
	0.6	[1.3498588075760021E+0000, 1.3498588075760040E+0000]	$\approx 1.81 \cdot 10^{-15}$
	0.8	[1.4918246976412689E+0000, 1.4918246976412716E+0000]	$\approx 2.60 \cdot 10^{-15}$
	1.0	[1.6487212707001263E+0000, 1.6487212707001299E+0000]	$\approx 3.54 \cdot 10^{-14}$
(4.73)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 1.60 \cdot 10^{-16}$
$n = 4$	0.4	[1.2214027581601697E+0000, 1.2214027581601700E+0000]	$\approx 2.42 \cdot 10^{-16}$
	0.6	[1.3498588075760029E+0000, 1.3498588075760033E+0000]	$\approx 3.51 \cdot 10^{-16}$
	0.8	[1.4918246976412700E+0000, 1.4918246976412706E+0000]	$\approx 5.00 \cdot 10^{-16}$
	1.0	[1.6487212707001277E+0000, 1.6487212707001285E+0000]	$\approx 7.01 \cdot 10^{-15}$

For implicit interval methods of Adams-Moulton and Milne-Simspon types we have tested both kinds of methods, i.e. based on interval backward difference formulas and based only on interval values of functions. The obtained results (see Tables 4.5 and 4.6) confirm Theorems 4.5 and 4.8, i.e. that the formulas without interval backward differences are better. It should be added that in all iterations, e.g. (4.85), (4.86) and (4.113), we have assumed the accuracy 10^{-18} and the number of iterations has not exceeded 5.

Table 4.5.The solutions of the problem (2.2) obtained by interval methods of Adams-Moulton type

Method	t_k	Y_k	*Width*
(4.77)	0.2	[1.1051709180755696E+0000, 1.1051709180758077E+0000]	$\approx 2.38 \cdot 10^{-13}$
$n = 1$	0.6	[1.3498588075756735E+0000, 1.3498588075765571E+0000]	$\approx 8.83 \cdot 10^{-13}$
	1.0	[1.6487212706993650E+0000, 1.6487212707012116E+0000]	$\approx 1.85 \cdot 10^{-12}$
(4.82)	0.2	[1.1051709180755756E+0000, 1.1051709180758017E+0000]	$\approx 2.26 \cdot 10^{-13}$
$n = 1$	0.4	[1.2214027581600107E+0000, 1.2214027581604866E+0000]	$\approx 4.76 \cdot 10^{-13}$
	0.6	[1.3498588075757393E+0000, 1.3498588075764912E+0000]	$\approx 7.52 \cdot 10^{-13}$
	0.8	[1.4918246976408816E+0000, 1.4918246976419386E+0000]	$\approx 1.05 \cdot 10^{-12}$
	1.0	[1.6487212706995912E+0000, 1.6487212707009854E+0000]	$\approx 1.39 \cdot 10^{-12}$

Table 4.5. (cont.)

Method	t_k	Y_k	*Width*
(4.83)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 2.14 \cdot 10^{-16}$
$n = 2$	0.4	[1.2214027581601697E+0000, 1.2214027581601701E+0000]	$\approx 3.44 \cdot 10^{-16}$
	0.6	[1.3498588075760029E+0000, 1.3498588075760035E+0000]	$\approx 4.91 \cdot 10^{-16}$
	0.8	[1.4918246976412700E+0000, 1.4918246976412708E+0000]	$\approx 6.53 \cdot 10^{-16}$
	1.0	[1.6487212707001278E+0000, 1.6487212707001287E+0000]	$\approx 8.37 \cdot 10^{-16}$
(4.79)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 1.80 \cdot 10^{-16}$
$n = 3$	0.6	[1.3498588075760029E+0000, 1.3498588075760034E+0000]	$\approx 4.18 \cdot 10^{-16}$
	1.0	[1.6487212707001277E+0000, 1.6487212707001286E+0000]	$\approx 8.26 \cdot 10^{-16}$
(4.84)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 1.60 \cdot 10^{-16}$
$n = 3$	0.4	[1.2214027581601697E+0000, 1.2214027581601700E+0000]	$\approx 2.33 \cdot 10^{-16}$
	0.6	[1.3498588075760029E+0000, 1.3498588075760033E+0000]	$\approx 3.15 \cdot 10^{-16}$
	0.8	[1.4918246976412701E+0000, 1.4918246976412706E+0000]	$\approx 4.10 \cdot 10^{-16}$
	1.0	[1.6487212707001279E+0000, 1.6487212707001285E+0000]	$\approx 5.20 \cdot 10^{-16}$

Table 4.6.The solutions of the problem (2.2) obtained by interval methods of Milne-Simspon type

Method	t_k	Y_k	*Width*
(4.107)	0.2	[1.1051709180753335E+0000, 1.1051709180760855E+0000]	$\approx 7.52 \cdot 10^{-13}$
$n = 1$	0.6	[1.3498588075747373E+0000, 1.3498588075778749E+0000]	$\approx 3.14 \cdot 10^{-12}$
	1.0	[1.6487212706971386E+0000, 1.6487212707046242E+0000]	$\approx 7.49 \cdot 10^{-12}$
(4.110)	0.2	[1.1051709180753706E+0000, 1.1051709180760484E+0000]	$\approx 6.78 \cdot 10^{-13}$
$n = 1$	0.4	[1.2214027581596169E+0000, 1.2214027581610438E+0000]	$\approx 1.42 \cdot 10^{-12}$
	0.6	[1.3498588075751788E+0000, 1.3498588075774335E+0000]	$\approx 2.25 \cdot 10^{-12}$
	0.8	[1.4918246976401830E+0000, 1.4918246976433526E+0000]	$\approx 3.17 \cdot 10^{-12}$
	1.0	[1.6487212706987910E+0000, 1.6487212707029717E+0000]	$\approx 4.18 \cdot 10^{-12}$
(4.108)	0.2	[1.1051709180756475E+0000, 1.1051709180756477E+0000]	$\approx 1.07 \cdot 10^{-16}$
$n = 2$	0.6	[1.3498588075760028E+0000, 1.3498588075760034E+0000]	$\approx 4.95 \cdot 10^{-16}$
	1.0	[1.6487212707001275E+0000, 1.6487212707001289E+0000]	$\approx 1.30 \cdot 10^{-15}$

Table 4.6. (cont.)

Method	t_k	Y_k	*Width*
(4.111)	0.2	[1.1051709180756475E+0000, 1.1051709180756477E+0000]	$\approx 8.59 \cdot 10^{-17}$
$n = 2$	0.4	[1.2214027581601697E+0000, 1.2214027581601700E+0000]	$\approx 1.83 \cdot 10^{-16}$
	0.6	[1.3498588075760029E+0000, 1.3498588075760033E+0000]	$\approx 2.88 \cdot 10^{-16}$
	0.8	[1.4918246976412701E+0000, 1.4918246976412706E+0000]	$\approx 4.04 \cdot 10^{-16}$
	1.0	[1.6487212707001279E+0000, 1.6487212707001285E+0000]	$\approx 5.32 \cdot 10^{-16}$
(4.109)	0.2	[1.1051709180756476E+0000, 1.1051709180756477E+0000]	$\approx 4.27 \cdot 10^{-17}$
$n = 3$	0.6	[1.3498588075760030E+0000, 1.3498588075760033E+0000]	$\approx 2.00 \cdot 10^{-16}$
	1.0	[1.6487212707001278E+0000, 1.6487212707001285E+0000]	$\approx 5.38 \cdot 10^{-16}$
(4.112)	0.2	[1.1051709180756476E+0000, 1.1051709180756477E+0000]	$\approx 3.07 \cdot 10^{-17}$
$n = 3$	0.4	[1.2214027581601698E+0000, 1.2214027581601699E+0000]	$\approx 6.26 \cdot 10^{-17}$
	0.6	[1.3498588075760030E+0000, 1.3498588075760032E+0000]	$\approx 1.00 \cdot 10^{-16}$
	0.8	[1.4918246976412702E+0000, 1.4918246976412704E+0000]	$\approx 1.41 \cdot 10^{-16}$
	1.0	[1.6487212707001280E+0000, 1.6487212707001283E+0000]	$\approx 1.85 \cdot 10^{-16}$

From the results presented in Tables 4.5 and 4.6 one more conclusion follows: for the problem considered implicit interval methods of Milne-Simpson type give, for the same number of method steps n ($n > 1$), better solutions (we obtain intervals with smaller widths).

As in the case of explicit interval mutistep methods, for implicit ones the increase of method steps n does not necessarily result in an improvement of interval solutions. In Figure 4.4 we present the widths of interval solutions at the final moment ($t = 1$) for all interval methods considered.

It can also be interesting how the step sizes affect the widths of interval solutions. If we take a step size that is too small then we have to do more calculations and therefore we cause more rounding errors. We present the appropriate relations for the interval implicit method in Figure 4.5.

In implicit interval methods, for which the obtained solutions are presented in Tables 4.5 and 4.6, we took the solutions Y_{k-1} as starting points $Y_k^{(0)}$ in the iteration to obtain Y_k. In order to reduce the number of iterations a better approach is to take Y_k obtained in explicit interval methods as initial approximations. In Table 4.7 we present the results obtained using such an approach. As a predictor in each case we took the n-step explicit interval method of Adams-Bashforth type and we used the implicit interval method of Adams-Moulton type with the same number n of method steps as a corrector. We assumed the accuracy 10^{-18} in iterations and

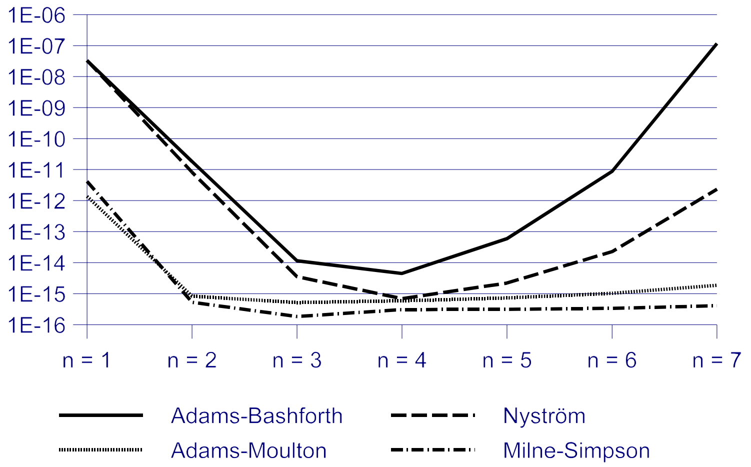


Figure 4.4. The widths of interval solutions obtained by the multistep methods versus the number *n* of method steps (at *t* = 1)

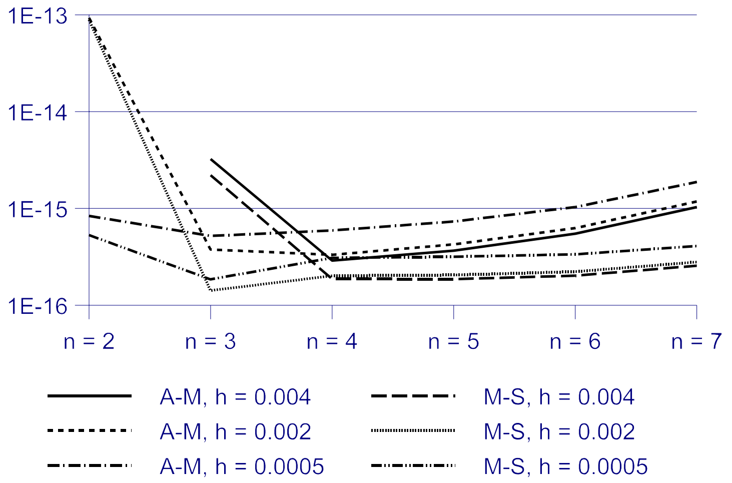


Figure 4.5. The widths of interval solutions obtained by the implicit multistep methods versus the number *n* of method steps for different step size (at *t* = 1)

h = 0.0005. For *n* = 1 only three iterations were needed, for *n* = 2 only two, and for *n* = 3 only one iteration turned out necessary.

Table 4.7.The solutions of the problem (2.2) obtained by interval predictor-corrector methods of Adams type

Method	t_k	Y_k	*Width*
(4.44)	0.2	[1.1051709180755756E+0000, 1.1051709180758017E+0000]	$\approx 2.26 \cdot 10^{-13}$
(4.82)	0.4	[1.2214027581600107E+0000, 1.2214027581604866E+0000]	$\approx 4.76 \cdot 10^{-13}$
$n = 1$	0.6	[1.3498588075757393E+0000, 1.3498588075764912E+0000]	$\approx 7.52 \cdot 10^{-13}$
	0.8	[1.4918246976408816E+0000, 1.4918246976419386E+0000]	$\approx 1.05 \cdot 10^{-12}$
	1.0	[1.6487212706995912E+0000, 1.6487212707009854E+0000]	$\approx 1.39 \cdot 10^{-12}$
(4.45)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 2.14 \cdot 10^{-16}$
(4.83)	0.4	[1.2214027581601697E+0000, 1.2214027581601701E+0000]	$\approx 3.44 \cdot 10^{-16}$
$n = 2$	0.6	[1.3498588075760029E+0000, 1.3498588075760035E+0000]	$\approx 4.91 \cdot 10^{-16}$
	0.8	[1.4918246976412700E+0000, 1.4918246976412708E+0000]	$\approx 6.54 \cdot 10^{-16}$
	1.0	[1.6487212707001278E+0000, 1.6487212707001287E+0000]	$\approx 8.38 \cdot 10^{-16}$
(4.46)	0.2	[1.1051709180756475E+0000, 1.1051709180756478E+0000]	$\approx 1.60 \cdot 10^{-16}$
(4.84)	0.4	[1.2214027581601697E+0000, 1.2214027581601700E+0000]	$\approx 2.34 \cdot 10^{-16}$
$n = 3$	0.6	[1.3498588075760029E+0000, 1.3498588075760033E+0000]	$\approx 3.16 \cdot 10^{-16}$
	0.8	[1.4918246976412701E+0000, 1.4918246976412706E+0000]	$\approx 4.10 \cdot 10^{-16}$
	1.0	[1.6487212707001279E+0000, 1.6487212707001285E+0000]	$\approx 5.21 \cdot 10^{-16}$

For the conventional methods corresponding to the interval method considered we obtain the solutions presented in Table 4.8. Let us note that only in the case of the fourth order methods, i.e. Adams-Bashforth (4.12), Nyström's (4.31), Adams-Moulton (4.22) and Milne's (4.37), the solutions belongs to the interval solutions obtained.

Example 4.6

In Section 3.7 we considered the simplified Hill equations describing the motion of the Moon and we used a number of interval methods of Runge-Kutta type to solve the initial value problem (2.5) (see Example 3.3). It appears that by such methods we able to obtain the interval solutions only for merely 1.35% of the orbit period. But these methods can be used to obtain starting intervals for multistep methods.

Assuming the same initial data as in Example 3.3 and $h = 0.0005$, by the implicit interval method (3.58) we find starting intervals presented in Table 4.9.

Table 4.8. The solutions of the problem (2.2) at $t = 1.0$ obtained by conventional multistep methods

Method	*Order*	y	*Error*
Adams-Bashforth (4.10)	2	1.6487212492463580E+0000	$\approx 2.15 \cdot 10^{-8}$
Adams-Bashforth (4.11)	3	1.6487212706953040E+0000	$\approx 4.82 \cdot 10^{-12}$
Adams-Bashforth (4.12)	4	1.6487212707001270E+0000	$\approx 1.13 \cdot 10^{-15}$
Midpoint rule (4.29)	2	1.6487212621143958E+0000	$\approx 8.59 \cdot 10^{-9}$
Nyström’s (4.30)	3	1.6487212706979834E+0000	$\approx 2.14 \cdot 10^{-12}$
Nyström’s (4.31)	4	1.6487212707001276E+0000	$\approx 5.30 \cdot 10^{-16}$
Trapezoidal rule (4.20)	2	1.6487212749936732E+0000	$\approx 4.29 \cdot 10^{-9}$
Adams-Moulton (4.21)	3	1.6487212707006645E+0000	$\approx 5.36 \cdot 10^{-13}$
Adams-Moulton (4.22)	4	1.6487212707001282E+0000	$\approx 7.47 \cdot 10^{-17}$
Milne’s (4.37)	4	1.6487212707001282E+0000	$\approx 5.10 \cdot 10^{-18}$

Table 4.9. Starting intervals for four-step interval methods and the problem (2.5) with $M = 0$ and $\kappa = 1$

k	$t_k = kh \in T_k$	Y_{sk}
1	0.0005	Y_{11} = [9.9999987500000260E−0001, 9.9999987500000261E−0001] Y_{21} = [4.9999997918888892E−0004, 4.9999997916666694E−0004] Y_{31} = [−4.9999997916666694E−0004,−4.9999997916666692E−0004] Y_{41} = [9.9999987500000260E−0001, 9.9999987500000261E−0001]
2	0.0010	Y_{12} = [9.9999950000004166E−0001, 9.9999950000004167E−0001] Y_{22} = [9.9999983333334166E−0004, 9.9999983333334168E−0004] Y_{32} = [−9.9999983333334168E−0004,−9.9999983333334165E−0004] Y_{42} = [9.9999950000004166E−0001, 9.9999950000004167E−0001]
3	0.0015	Y_{13} = [9.9999887500021093E−0001, 9.9999887500021094E−0001] Y_{23} = [1.4999994375000632E−0003, 1.4999994375000633E−0003] Y_{33} = [−1.4999994375000633E−0003,−1.4999994375000632E−0003] Y_{43} = [9.9999887500021093E−0001, 9.9999887500021094E−0001]

In order to solve our problem, let us take into account the four-step explicit interval method (4.73). For this method we need interval extensions of $f_s(t, y)$ and $\psi_s(t, y) \equiv f_s^{(4)}(t, y) \equiv y_s^{(5)}(t)$, where $s = 1, 2, 3, 4$. From (2.5) it follows that for $M = 0$ and $\kappa = 1$ we have

$$f_1(t, y) = y_3, \quad f_2(t, y) = y_4, \quad f_3(t, y) = -\frac{y_1}{r^3}, \quad f_4(t, y) = -\frac{y_2}{r^3}, \tag{4.121}$$

where $r = \sqrt{y_1^2 + y_2^2}$. From (4.121), after somewhat burdensome calculations, we get

$$\begin{aligned}
\psi_1(t, y) = {} & \frac{3}{r^5}\left(3 - \frac{30y_1^2}{r^2} + \frac{35y_1^4}{r^4}\right) + \frac{15y_1y_2}{r^7}\left(\frac{21y_1^2}{r^2} - 9\right) y_3^2 y_4 + \\
& + \frac{9}{r^5}\left(1 - \frac{5y_1^2}{r^2} - \frac{5y_2^2}{r^2} + \frac{35y_1^2y_2^2}{r^4}\right) y_3 y_4^2 + \frac{15y_1y_2}{r^7}\left(\frac{7y_2^2}{r^2} - 3\right) y_4^3 + \\
& + \frac{1}{r^6}\left(1 - \frac{33y_1^2}{r^2} - \frac{9y_2^2}{r^2} + \frac{54y_1^4}{r^4} + \frac{54y_1^2y_2^2}{r^4}\right) y_3 + \\
& + \frac{6y_1y_2}{r^8}\left(\frac{9y_1^2}{r^2} + \frac{9y_2^2}{r^2} - 4\right) y_4,
\end{aligned}$$

$$\begin{aligned}
\psi_2(t, y) = {} & \frac{15y_1y_2}{r^7}\left(\frac{7y_1^2}{r^2} - 3\right) y_3^3 + \frac{9}{r^5}\left(1 - \frac{5y_1^2}{r^2} - \frac{5y_2^2}{r^2} + \frac{35y_1^2y_2^2}{r^4}\right) y_3^2 y_4 + \\
& + \frac{15y_1y_2}{r^7}\left(\frac{21y_2^2}{r^2} - 9\right) y_3 y_4^2 + \frac{3}{r^5}\left(3 - \frac{30y_2^2}{r^2} + \frac{35y_2^4}{r^4}\right) y_4^3 + \\
& + \frac{6y_1y_2}{r^8}\left(\frac{9y_1^2}{r^2} + \frac{9y_2^2}{r^2} - 4\right) y_3 + \\
& + \frac{1}{r^6}\left(1 - \frac{9y_1^2}{r^2} - \frac{33y_2^2}{r^2} + \frac{54y_1^2y_2^2}{r^4} + \frac{54y_2^4}{r^4}\right) y_4,
\end{aligned}$$

$$\begin{aligned}
\psi_3(t, y) = {} & \frac{15y_1}{r^7}\left(-15 + \frac{70y_1^2}{r^2} - \frac{63y_1^4}{r^4}\right) y_3^4 + \\
& + \frac{180y_2}{r^7}\left(-1 + \frac{14y_1^2}{r^2} - \frac{21y_1^4}{r^4}\right) y_3^3 y_4 + \\
& + \frac{90y_1}{r^7}\left(-3 + \frac{7y_1^2}{r^2} + \frac{21y_2^2}{r^2} - \frac{63y_1^2y_2^2}{r^4}\right) y_3^2 y_4^2 + \\
& + \frac{60y_2}{r^7}\left(-3 + \frac{21y_1^2}{r^2} + \frac{7y_2^2}{r^2} - \frac{63y_1^2y_2^2}{r^4}\right) y_3 y_4^3 +
\end{aligned}$$

$$
\begin{aligned}
&+\frac{45y_1}{r^7}\left(-1+\frac{14y_2^2}{r^2}-\frac{21y_2^4}{r^4}\right)y_4^4+\\
&+\frac{3y_1}{r^8}\left(-33+\frac{250y_1^2}{r^2}+\frac{105y_2^2}{r^2}-\frac{285y_1^4}{r^4}-\frac{285y_1^2y_2^2}{r^4}\right)y_3^2+\\
&+\frac{6y_2}{r^8}\left(-11+\frac{181y_1^2}{r^2}+\frac{36y_2^2}{r^2}-\frac{285y_1^4}{r^4}-\frac{285y_1^2y_2^2}{r^4}\right)y_3y_4+\\
&+\frac{3y_1}{r^8}\left(-11+\frac{33y_1^2}{r^2}+\frac{178y_2^2}{r^2}-\frac{285y_1^2y_2^2}{r^4}-\frac{285y_2^4}{r^4}\right)y_4^2+\\
&+\frac{y_1}{r^9}\left(-1+\frac{33y_1^2}{r^2}+\frac{33y_2^2}{r^2}-\frac{54y_1^4}{r^4}-\frac{108y_1^2y_2^2}{r^4}-\frac{54y_2^4}{r^4}\right),
\end{aligned}
$$

$$
\begin{aligned}
\psi_4(t,y)=&\frac{45y_2}{r^7}\left(-1+\frac{14y_1^2}{r^2}-\frac{21y_1^4}{r^4}\right)y_3^4+\\
&+\frac{60y_1}{r^7}\left(-3+\frac{7y_1^2}{r^2}+\frac{21y_2^2}{r^2}-\frac{63y_1^2y_2^2}{r^4}\right)y_3^3y_4+\\
&+\frac{90y_2}{r^7}\left(-3+\frac{21y_1^2}{r^2}+\frac{7y_2^2}{r^2}-\frac{63y_1^2y_2^2}{r^4}\right)y_3^2y_4^2+\\
&+\frac{180y_1}{r^7}\left(-1+\frac{14y_1^2}{r^2}-\frac{21y_2^4}{r^4}\right)y_3y_4^3+\\
&+\frac{15y_2}{r^7}\left(-15+\frac{70y_2^2}{r^2}-\frac{63y_2^4}{r^4}\right)y_4^4+\\
&+\frac{3y_2}{r^8}\left(-11+\frac{178y_1^2}{r^2}+\frac{33y_2^2}{r^2}-\frac{285y_1^4}{r^4}-\frac{285y_1^2y_2^2}{r^4}\right)y_3^2+\\
&+\frac{6y_1}{r^8}\left(-11+\frac{36y_1^2}{r^2}+\frac{181y_2^2}{r^2}-\frac{285y_1^2y_1^2}{r^4}-\frac{285y_2^4}{r^4}\right)y_3y_4+\\
&+\frac{3y_2}{r^8}\left(-33+\frac{105y_1^2}{r^2}+\frac{250y_2^2}{r^2}-\frac{285y_1^2y_1^2}{r^4}-\frac{285y_2^4}{r^4}\right)y_4^2+\\
&+\frac{y_2}{r^9}\left(-1+\frac{33y_1^2}{r^2}+\frac{33y_2^2}{r^2}-\frac{54y_1^4}{r^4}-\frac{108y_1^2y_1^2}{r^4}-\frac{54y_2^4}{r^4}\right).
\end{aligned}
$$

From these formulas the interval extensions of $f_s(t, y)$ and $\psi_s(t, y)$ can now be easily find. As in Example 3.3, we cannot take the whole domains of definitions of y_s ($s = 1, 2$) to determine Δ_{y_s}, because we would get divisions by intervals containing zero. Assuming that $h_0 = 0.0005$ and

$$\Delta_t = [0, 1.0015], \quad \Delta_{y_1} = \Delta_{y_4} = \left[\underline{0.4}, \overline{1.1}\right], \quad \Delta_{y_2} = \left[\underline{-0.1}, 1\right], \quad \Delta_{y_3} = \left[-1, \overline{0.1}\right],$$

we can find the interval solutions at $t = 1$, and additionally at $t = 1.0005$, 1.0010, 1.0015. The intervals obtained for these values of t can be used as new starting points. Taking

$$\Delta_{y_1} = \Delta_{y_4} = \left[-0.5, \overline{0.6}\right], \quad \Delta_{y_2} = \left[\underline{0.8}, \overline{1.1}\right], \quad \Delta_{y_3} = \left[-1, \overline{-0.7}\right],$$

we obtain the interval solutions at $t = 2$, and additionally at $t = 2.0005$, 2.0010, 2.0015. Proceeding further in the same way and assuming that

$$\Delta_{y_1} = \Delta_{y_4} = \left[\underline{-1.1}, \overline{-0.3}\right], \quad \Delta_{y_2} = [0, 1], \quad \Delta_{y_4} = [-1, 0]$$

for $t \in [2, 3.0015]$,

$$\Delta_{y_1} = \Delta_{y_4} = \left[-1, \overline{-0.6}\right], \quad \Delta_{y_2} = \left[\underline{-0.9}, \overline{0.2}\right], \quad \Delta_{y_3} = \left[\underline{-0.2}, \overline{0.9}\right]$$

for $t \in [3, 4.0015]$ and

$$\Delta_{y_1} = \Delta_{y_4} = \left[\underline{-0.6}, \overline{0.4}\right], \quad \Delta_{y_2} = \left[-1, \overline{-0.7}\right], \quad \Delta_{y_3} = \left[\underline{0.7}, 1\right]$$

for $t \in [4, 5.0015]$, it is possible to get the interval solutions up to $t = 5.0015$. In Table 4.10 we present the solutions obtained at $t = 1, 2, 3, 4, 5$. Unfortunately, for t approximately greater than 5 the wrapping effect causes the interval solution to be worthless (the intervals are too large). This effect is shown in Figure 4.6. Of course, in each case the exact solution belongs to the interval solution obtained (compare Table 2.2) ■

Table 4.10. The interval solutions of the problem (2.5) with $M = 0$ and $\kappa = 1$ obtained by the four-step method (4.73)

$t_k \in T_k$	$Y_s = Y_{sk}$	*Width of* Y_s
1.0	$Y_1 = [\ 5.4030230586442026E-0001,\ 5.4030230587174773E-0001]$	$\approx 7.33 \cdot 10^{-12}$
	$Y_2 = [\ 8.4147098480488006E-0001,\ 8.4147098481072161E-0001]$	$\approx 5.84 \cdot 10^{-12}$
	$Y_3 = [-8.4147098481461534E-0001, -8.4147098480141829E-0001]$	$\approx 1.32 \cdot 10^{-11}$
	$Y_4 = [\ 5.4030230586128248E-0001,\ 5.4030230587430483E-0001]$	$\approx 1.30 \cdot 10^{-11}$

Table 4.10. (cont.)

$t_k \in T_k$	$Y_s = Y_{sk}$	*Width of* Y_s
2.0	Y_1 = [−4.1614683694363343E−0001,−4.1614683615103482E−0001]	$\approx 7.93 \cdot 10^{-10}$
	Y_2 = [9.0929742535117589E−0001, 9.0929742829964891E−0001]	$\approx 2.95 \cdot 10^{-9}$
	Y_3 = [−9.0929742770118559E−0001,−9.0929742595018418E−0001]	$\approx 1.75 \cdot 10^{-9}$
	Y_4 = [−4.1614683945624377E−0001,−4.1614683363825487E−0001]	$\approx 5.82 \cdot 10^{-9}$
3.0	Y_1 = [−9.8999289387333900E−0001,−9.8999209933056823E−0001]	$\approx 7.95 \cdot 10^{-7}$
	Y_2 = [1.4111973686998131E−0001, 1.4112027925037821E−0001]	$\approx 5.42 \cdot 10^{-7}$
	Y_3 = [−1.4112083868795111E−0001,−1.4111917742596258E−0001]	$\approx 1.66 \cdot 10^{-6}$
	Y_4 = [−9.8999288198336006E−0001,−9.8999211121904063E−0001]	$\approx 7.71 \cdot 10^{-7}$
4.0	Y_1 = [−6.5378657167462219E−0001,−6.5350066759754963E−0001]	$\approx 2.86 \cdot 10^{-4}$
	Y_2 = [−7.5687781480230884E−0001,−7.5672717419414288E−0001]	$\approx 1.51 \cdot 10^{-4}$
	Y_3 = [7.5654581887407921E−0001, 7.5705920035140095E−0001]	$\approx 5.13 \cdot 10^{-4}$
	Y_4 = [−6.5382363465163544E−0001,−6.5346358514279663E−0001]	$\approx 3.60 \cdot 10^{-4}$
5.0	Y_1 = [2.7059624260337538E−0001, 2.9668409962868753E−0001]	$\approx 2.61 \cdot 10^{-2}$
	Y_2 = [−1.0053307587715992E+0000,−9.1223612943957444E−0001]	$\approx 9.31 \cdot 10^{-2}$
	Y_3 = [9.3617509799322116E−0001, 9.8086374299139689E−0001]	$\approx 4.47 \cdot 10^{-2}$
	Y_4 = [1.9154124355505947E−0001, 3.7920966729796944E−0001]	$\approx 1.88 \cdot 10^{-1}$

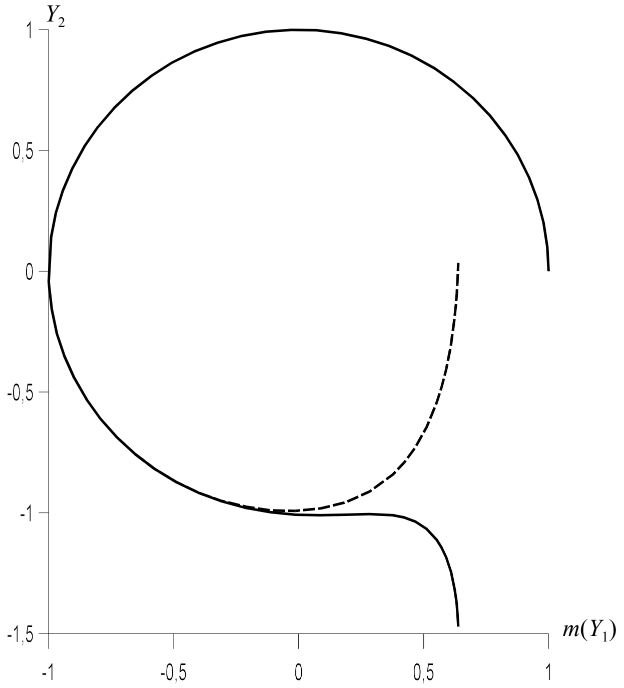


Figure 4.6. The wrapping effect caused by the method (4.73) for the problem (2.5) with $M = 0$ and $\kappa = 1$

Example 4.7

In Example 3.5 we tried to solve the problem of the motion of a simple pendulum by the interval version of Alexander's method (3.62). It appears that we get an interval solution only for about 10% of the period. But any of interval versions of Runge-Kutta type method can be used to obtain starting points for interval multistep methods. Applying the two-stage semi-implicit interval method (3.59) for the same data as in Example 3.5 and for $h = 0.001$ we have get additional starting intervals as follows:

$$Y_1(0.001) = [-5.1347415402712304\text{E}-0003,\ -5.1347415400712303\text{E}-0003],$$
$$Y_2(0.001) = [\ 5.2359620822533077\text{E}-0001,\ \ 5.2359620822553078\text{E}-0001].$$

Then, by the two-step implicit interval methods (4.83) and (4.111), after 7–8 iterations in each step we obtained interval solutions presented in Tables 4.11 and 4.12, respectively. In both of these interval methods we assumed the accuracy 10^{-18} in the iteration, and took $h = 0.001$,

$$\Delta_t = [0, 2],\quad \Delta_{y_1} = \left[\underline{-1.8}, \overline{1.8}\right],\quad \Delta_{y_1} = \left[\underline{-0.6}, \overline{0.6}\right],$$

and used interval extensions of $\overline{\psi}_s(t, y) \equiv y_s^{(4)}(t)$ $(s = 1, 2)$, where – on the basis of (2.12) –

$$\overline{\psi}_1(t, y) = u^4 y_1,\quad \overline{\psi}_2(t, y) = u^4 y_2.$$

We can observe that for each t the exact solution belongs to the interval solutions obtained (compare Table 2.4) and that the interval method of Milne-Simpson type (4.111) gives a somewhat better solution than the interval method of Adams-Moulton type (4.83). ■

Table 4.11. The interval solutions of the problem (2.12) – (2.13) obtained by the two-step method (4.83)

$t_k \in T_k$	$Y_s = Y_{sk}$	*Width of* Y_s
0.5	$Y_1 = [-1.6396588322988759\text{E}+0000, -1.6396588321649396\text{E}+0000]$	$\approx 1.34\cdot 10^{-10}$
	$Y_2 = [\ 2.6272853320688279\text{E}-0003,\ 2.6272853745701990\text{E}-0003]$	$\approx 4.25\cdot 10^{-11}$
1.0	$Y_1 = [-1.6454781641330172\text{E}-0002, -1.6454780680840558\text{E}-0002]$	$\approx 9.60\cdot 10^{-10}$
	$Y_2 = [-5.2357240965344030\text{E}-0001, -5.2357240934706097\text{E}-0001]$	$\approx 3.06\cdot 10^{-10}$
1.5	$Y_1 = [\ 1.6394936973823808\text{E}+0000,\ 1.6394937034708161\text{E}+0000]$	$\approx 6.09\cdot 10^{-9}$
	$Y_2 = [-7.8815924351058335\text{E}-0003, -7.8815904912303674\text{E}-0003]$	$\approx 1.94\cdot 10^{-9}$
2.0	$Y_1 = [\ 3.2907886190368976\text{E}-0002,\ 3.2907924096365751\text{E}-0002]$	$\approx 3.79\cdot 10^{-8}$
	$Y_2 = [\ 5.2349330780937726\text{E}-0001,\ 5.2349331991355161\text{E}-0001]$	$\approx 1.21\cdot 10^{-8}$

Table 4.12. The interval solutions of the problem (2.12) – (2.13) obtained by the two-step method (4.111)

$t_k \in T_k$	$Y_s = Y_{sk}$	*Width of* Y_s
0.5	$Y_1 = [-1.6396588322884904\text{E}{+}0000, -1.6396588321755220\text{E}{+}0000]$	$\approx 1.13 \cdot 10^{-10}$
	$Y_2 = [\ 2.6272853322057914\text{E}{-}0003,\ 2.6272853678587899\text{E}{-}0003]$	$\approx 3.57 \cdot 10^{-11}$
1.0	$Y_1 = [-1.6454781465753352\text{E}{-}0002, -1.6454780815256611\text{E}{-}0002]$	$\approx 6.50 \cdot 10^{-10}$
	$Y_2 = [-5.2357240960387409\text{E}{-}0001, -5.2357240939672969\text{E}{-}0001]$	$\approx 2.07 \cdot 10^{-10}$
1.5	$Y_1 = [\ 1.6394936988159553\text{E}{+}0000,\ 1.6394937020381708\text{E}{+}0000]$	$\approx 3.22 \cdot 10^{-9}$
	$Y_2 = [-7.8815919673964566\text{E}{-}0003, -7.8815909391754222\text{E}{-}0003]$	$\approx 1.02 \cdot 10^{-9}$
2.0	$Y_1 = [\ 3.2907897336904612\text{E}{-}0002,\ 3.2907912867531363\text{E}{-}0002]$	$\approx 1.55 \cdot 10^{-8}$
	$Y_2 = [\ 5.2349331138252507\text{E}{-}0001,\ 5.2349331634096158\text{E}{-}0001]$	$\approx 4.96 \cdot 10^{-8}$

The main conclusion from the examples presented and many other carried out by the author (concerning not only the multistep interval methods, but also the interval methods of Runge-Kutta type) is such that the interval methods for solving the initial value problem executed in floating-point interval arithmetic yield solutions in the form of intervals which **contain all possible numerical errors**, i.e. representation errors, rounding errors and errors of methods. Other conclusions concerning the multistep interval methods are as follows:

- for the same number of steps explicit interval methods of Nyström type are somewhat better than the methods of Adams-Bashforth type,
- for the same number of steps implicit interval methods of Milne-Simpson type give somewhat better results than the methods of Adams-Moulton type,
- the implicit interval methods based on backward interval differences give somewhat worse results than the methods based only on the combinations of interval function values at different points,
- the application of an explicit interval multistep method as the predictor for an implicit one significantly reduces the number of iterations involved,
- for each particular problem one should choose the appropriate step size and the number of method steps to obtain the interval solution with the smallest width (for a given step size there exists the optimal number of method steps, and for a given number of method steps there exists the best step size).

Chapter 5

Other Interval Methods for Solving the Initial Value Problem

5.1. Known Methods

The interval methods presented in Chapters 3 and 4 are not the only interval methods that one can use for solving the initial value problem. In 1965, R. E. Moore described an interval method for ordinary differential equations using interval arithmetic for the first time (see [131] and [133]).

The *Moore method* concerns the initial value problem of the form

$$y' = f(y(t)), \quad y(0) = y_0, \tag{5.1}$$

where the function f is defined on an interval $\Delta_y = \left[\underline{b}, \overline{b}\right]$ and $y_0 \in \left(\underline{b}, \overline{b}\right)$. Let us assume that the function $f(y)$ has an interval extension $F(Y)$ and, further, that:

- the function $F(Y)$ is defined and continuous for all $Y \subset \Delta_y$,
- the function $F(Y)$ is monotonic with respect to inclusion,
- for each $Y \subset \Delta_y$ there exists a constant $\Lambda > 0$ such that $w(F(Y)) \le \Lambda w(Y)$.

Since $y_0 \in \left(\underline{b}, \overline{b}\right)$, there exists $h > 0$ such that for $t \in [0, h]$ we have

$$y_0 + tF(\Delta_y) \subset \Delta_y.$$

Let r be a positive integer and let us partition the interval $[0, h]$ into subintervals

$$T_q = \left[\frac{(q-1)h}{r}, \frac{qh}{r}\right], \quad q = 1, 2, \ldots, r.$$

Let $Y_0 \subset \Delta_y$ be such an interval that $y_0 \in Y_0$. For each r let us define an interval function $Y_r(t)$, where $t \in [0, h]$, as follows:

$$Y_r(0) = y_0,$$

$$Y_r(t_q) = Y_r(t_{q-1}) + hF(Z_q), \quad t_q = \frac{qh}{r}, \quad q = 1, 2, \ldots, r, \tag{5.2}$$

where

$$Z_q = Y_r(t_{q-1}) + \left[0, \frac{h}{r}\right] F(\Delta_y). \tag{5.3}$$

For $t_{q-1} < t < t_q$ let

$$Y_r(t) = Y_r(t_{q-1}) + (t - t_{q-1})F(Z_q). \tag{5.4}$$

Since $t_q - t_{q-1} = h / r,$ then the equations (5.2) – (5.4) with the condition $Y_p(0) = y_0$ define a piecewise regulated, continuous interval function $Y_r(t)$ for all $t \in [0, h]$.

It can be proved (see e.g. [133] or [167]) that

$$w(Y_r(t)) \le \frac{3h}{r} \left| F(\Delta_y) \right| \left(\exp(h\Lambda) - 1 \right)$$

and

$$y(t) \in \bigcap_{p=1}^{\infty} Y_r(t)$$

for all $t \in [0, h]$, where $y(t)$ is the exact solution of the problem (5.1).

The formulas (5.2) – (5.4) determine a so-called *Moore's method of the first order*. In order to construct the *Moore method of order p* it is sufficient to assume that the function $y(t)$ has continuous derivatives up to the order p. Thus, from the Taylor theorem we have

$$y(t) = y(0) + \sum_{i=1}^{p-1} \frac{f^{(i-1)}(0)}{i!} t^i + \frac{f^{(p-1)}(\theta)}{p!} t^p,$$

where $\theta \in [0, h]$. If interval extensions $F^{(i)}$ of the functions $f^{(i)}$ are determined for the interval Δ_y and $y(0) \in Y(0) \subset \Delta_y$, then $y(t) \in Y(t)$, where

$$Y(t) = Y(0) + \sum_{i=1}^{p-1} \frac{F^{(i-1)}(Y(0))t^i}{i!} + \frac{F^{(p-1)}(\Delta_y)t^p}{p!},$$

under the condition that $Y([0, h]) \subset \Delta_y$. Moreover, if

$$w(F^{(i)}(Y)) \le \Lambda_i w(Y),$$

where $\Lambda_i > 0$ are constants, then

$$w(Y(t)) \le \frac{\Lambda_{p-1} w(\Delta_y) h^p}{p!} + \left(1 + \sum_{i=1}^{p-1} \frac{\Lambda_{i-1} h^i}{i!}\right) w(Y(0)).$$

In 1969, F. Krückeberg published a method (see [98]), which he called *three-process method* or *3PM*, where he considered the N-dimensional initial value problem of the form

$$y' = f(t, y(t)), \quad y(t_0) = y_0, \tag{5.5}$$

for which it is known that the exact solution $\tilde{y}(t; t_0, y_0)$ exists and is unique. If the initial conditions are such that $y_0 \in Y_0$, where Y_0 is an interval in $\mathbf{R}^N$, then the solution to (5.5) is a set of solutions that can be denoted by

$$\tilde{Y}(t) = \{z\colon z = \tilde{y}(t; t_0, y_0),\ y_0 \in Y_0\},$$

so that $\tilde{Y}(t_0) = Y_0$.

For such an additional requirement to the problem (5.5) we have to find an interval $Y_1^* \supseteq \tilde{Y}(t_1)$, where $t_1 = t_0 + h$, $h > 0$. In the first process we determine a step length $h > 0$ and an interval polynomial with vector coefficients

$$\hat{P}(t - t_0) = \sum_{i=0}^{k} P_i(t - t_0),$$

where k is a given integer, so that for all $t \in [t_0, t_1]$ we have

$$\hat{P}(t - t_0) \supseteq \tilde{Y}(t).$$

In most cases it is sufficient to construct $\hat{P}(t - t_0)$ for $k = 0$. An implementation of this was proposed by Moore in [134] and we obtain $\hat{P}(t - t_0) = P_0$ such that

$$\tilde{Y}(t) \subseteq P_0$$

for all $t \in [t_0, t_1]$.

In the second process we find an interval solution to (5.5) for a point $z_0 \in Y_0$. Using the Taylor series through terms of second order we have

$$Z_1 = z_0 + hF(t_0, z_0) + \frac{h^2}{2} F'([t_0, t_1], \hat{P}([0, h])). \tag{5.6}$$

Of course, $z_1 = \tilde{y}(t_1; t_0, z_0) \in Z_1$, because in the remainder of (5.6) the whole set of solutions $\tilde{Y}(t)$ is included in $\hat{P}([0, h])$.

The third process is an interval variant of the perturbation method. Let us denote by $U_0 = Y_0 - z_0$ a set of all perturbations of the initial value z_0. Let us write our problem in the form

$$\frac{d(u+z)}{dx} = f(t, u+z), \quad u(t_0) = u_0 \in U_0. \tag{5.7}$$

We know that $u(t_1) = z_1 = \tilde{y}(t_1; t_0, z_0)$ and we should determine

$$u_1 = \tilde{y}(t_1; t_0, u_0).$$

By linearization of the given differential equation in the neighborhood of

$$z_1 = \tilde{y}(t_1; t_0, z_0)$$

we get

$$\frac{du}{dt} = u\frac{\partial f}{\partial y}, \quad u(t_0) = u_0, \tag{5.8}$$

where in the N-dimensional case, $\partial f / \partial y$ is a matrix with elements $\partial f_i / \partial y_j$ $(i, j = 1, 2, \ldots, N)$. Thus, the interval $U_1 = QU_0$, where

$$Q = F(t_0, z_0) + \sum_{i=1}^{\infty} \frac{h^i}{i!}(L([t_0, t_1], P_0))^i,$$

and $L(t, y)$ denotes the matrix with elements $\partial f_i(t, y) / \partial y_j$, is the unknown interval solution of the problem (5.8) and (5.7). It is obvious that

$$Y_1^* = U_1 + Z_1 \supseteq Y_1.$$

In [167] Yu. I. Shokin proposed a method of the second order for the initial value problem (5.1), in which one should assume that the function f is defined and has bounded derivatives of the first and the second order in $\Delta_y = [\underline{b}, \overline{b}]$. Moreover, there exists an interval $Y_0 \subset \Delta_y$ (proper inclusion) such that $y_0 \in Y_0$. Let $F(Y)$ be an interval extension of f(y) with the same properties as in the Moore method and additionally let there exists an interval extension $\Psi(Y)$ of the function

$$f(y)\left[f(y)f''(y) + \left(f'(y)\right)^2\right],$$

and let this extension be monotonic with respect to inclusion.

Since the interval Y_0 is properly included in Δ_y, for a given number $h_0 > 0$ it is possible to find $\xi > 0$ such that

$$Y_0 + \xi\left(F(\Delta_y) - \frac{h_0^2}{12}\Psi(\Delta_y)\right) \subset \Delta_y.$$

An interval solution is constructed on the interval $[0, \xi]$. This interval is partitioned into m parts by the points $t_k = kh$ $(k = 0, 1, \ldots, m)$, where $h = \xi / m < h_0$.

It can be proved (see [87] or [167]) that if the intervals $Y_k = Y(t_k)$ are determined by the following formulas:

$$\begin{aligned} &Y_0 = Y(t_0) = Y(0),\\ &Y_{k+1} = Y_k + \frac{h}{2}\Big(F(Y_i) + F(Y_i + h[0,h]F(\Delta_y)))\Big) -\\ &-\frac{h^3}{12}\Psi(Y_i + [0,h]F(\Delta_y)),\\ &k = 0, 1, \ldots, m-1, \end{aligned} \tag{5.9}$$

then for the exact solution $y(t)$ of the problem (5.1) such that $y(0) \in Y_0$ we have $y(t_k) \in Y_k$ for $k = 1, 2, \ldots, m$, and

$$w(Y_k) \le Nh^2 + Mw(Y_0),$$

where M and N are some nonnegative constants independent of h. The method (5.9) is called the *Shokin method*.

In [167] Shokin presented yet another method based on the *Simpson formula*

$$\int_a^b g(t)dt = \frac{b-a}{6}\left[g(a) + 4g\left(\frac{a+b}{2}\right) + g(b)\right] - \frac{(b-a)^5}{2880}g^{(4)}(\eta), \tag{5.10}$$

where $\eta \in [a, b]$. The initial value problem

$$y' = f(t, y(t)), \quad y(0) = y_0 \in Y_0 \tag{5.11}$$

is solved on the interval $[0, \xi]$ partitioning by the points $t_i = ih$, where $h = \xi / m$ and $i = 0, 1, \ldots, m$. From (5.10) we have

$$\begin{aligned} \int_{t_i}^{t_{i+1}} y'(t)dt &= \frac{h}{6}\left[y'(t_i) + 4y'\left(\frac{t_i + t_{i+1}}{2}\right) + y'(t_{i+1})\right] - \frac{h^5}{2880}y^{(5)}(t_i + \vartheta h) =\\ &= \frac{h}{6}\left[f(t_i, y(t_i)) + 4f\left(\frac{t_i + t_{i+1}}{2}, y\left(\frac{t_i + t_{i+1}}{2}\right)\right) + f(t_{i+1}, y(t_{i+1}))\right] -\\ &-\frac{h^5}{2880}\psi(t_i + \vartheta h, y(t_i + \vartheta h)), \end{aligned} \tag{5.12}$$

where $\eta \in [t_i, t_{i+1}]$, and $\psi(t, y(t)) = y^{(5)}(t)$ can be expressed by $f(t, y(t))$ and its partial derivatives.

Let $F(T, Y)$ and $\Psi(T, Y)$ be interval extensions of $f(t, y(t))$ and $\psi(t, y(t))$, respectively, and let these extensions have the following properties:

- the function $F(T, Y)$ is defined and continuous for all $T \subset \Delta_t$ and $Y \subset \Delta_y$, where
$$\Delta_t = \{t \in \mathbf{R}\colon 0 \le t \le a\}, \quad \Delta_y = \{y \in \mathbf{R}\colon \underline{b} \le y \le \overline{b}\},$$
- 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 $Y \subset \Delta_y$ there exists a constant $\Lambda > 0$ such that
$$w(F(T, Y)) \le \Lambda(w(T) + w(Y)),$$
- the function $\Psi(T, Y)$ is defined for all $T \subset \Delta_t$ and $Y \subset \Delta_y$,
- the function $\Psi(T, Y)$ is monotonic with respect to inclusion.

The interval solution of the initial value problem (5.11), following from (5.12), is of the form:

$$Y(0) = Y_0,$$
$$Y_{k+1} = Y(t_{k+1}) = Y_k +$$
$$+\frac{h}{6}\left[F(T_k, Y_k) + 4F\left(\frac{T_k + T_{k+1}}{2}, \tilde{Y}\left(\frac{t_k + t_{k+1}}{2}\right)\right) + F(T_{k+1}, \tilde{Y}(t_{k+1}))\right] -$$
$$-\frac{h^5}{2880}\Psi(T_k + [0, h]F(\Delta_t, \Delta_y)),$$
$$k = 0, 1, \ldots, m-1.$$

The values $\tilde{Y}((t_k + t_{k+1})/2)$ and $\tilde{Y}(t_{k+1})$ should be calculated from T_k, T_{k+1} and Y_k by any method of the third order. It appears that $y(t_k) \in Y_k$ for $k = 1, 2, \ldots, m$, and

$$w(Y_k) \le Aw(Y_0) + B \max_{j=1,2,\ldots,m} w(T_j) + Ch^4,$$

where A, B and C are nonnegative constants independent of h.

In recent years a lot of studies have been conducted on a variety of the interval methods based on the high-order Taylor series. Below we outline a traditional method on the basis of [144]. For further reading we refer the reader to the relevant papers given in References [10 – 14, 16 – 21, 28 – 30, 35 – 40, 46, 59, 70, 71, 92, 107, 110, 111, 113 – 115, 140, 142, 145, 148, 152, 157 – 161]. We will use a notation usually applied in interval methods based on the Taylor series.

Let us consider the initial value problem of the form

$$y'(t) = f(y), \quad y(t_0) = y_0, \quad y \in \mathbf{R}^N, \ t \in \mathbf{R}. \tag{5.13}$$

The initial condition can be in an interval vector Y_0, i.e. $y_0 \in Y_0$. If we denote the solution to (5.13) by $y(t; t_0, y_0)$, we denote by $y(t; t_0, Y_0)$ the set of solutions originating from each initial condition in Y_0:

$$y(t; t_0, Y_0) = \{y(t; t_0, y_0) \colon y_0 \in Y_0\}.$$

We wish to find intervals that are guaranteed to contain the exact solution of (5.13) at points $t_0 < t_1 < \ldots < t_m$, i.e. we want to find Y_k such that

$$y(t_k; t_0, Y_0) \subseteq Y_k, \quad k = 1, 2, \ldots, m.$$

Let us suppose that we have computed Y_k at some point t_k. The interval solution at the next point in time, i.e. at t_{k+1}, we find in two phases.

At first, we try to find an interval $[t_k, t_{k+1}]$ and an a priori enclosure $\widetilde{Y}_k$ such that the problem

$$y' = f(y), \quad y(t_k) = y_k \tag{5.14}$$

has a unique solution for all $y_k \in Y_k$ and all $t \in [t_k, t_{k+1}]$, and

$$y(t; t_k, Y_k) \subseteq \widetilde{Y}_k \tag{5.15}$$

for all $t \in [t_k, t_{k+1}]$. Proving the existence and uniqueness, and finding $[t_k, t_{k+1}]$ and $\widetilde{Y}_k$, is usually based on applying a fixed-point theorem.

In the second phase we use $\widetilde{Y}_k$ to enclosure the truncation error of the method and compute a tighter enclosure Y_{k+1} at t_{k+1} such that

$$y(t_{k+1}; t_0, Y_0) \subseteq Y_{k+1} \subseteq \widetilde{Y}_k. \tag{5.16}$$

Let us introduce a convenient notation for the Taylor coefficients, denoting

$$f^{[0]}(y) = y,$$
$$f^{[i]}(y) = \frac{1}{i}\left(\frac{\partial f^{[i-1]}}{\partial y} f\right)(y), \quad i \geq 1.$$

Given the initial value problem (5.14), we have for the i-th Taylor coefficient of its solution

$$\frac{y^{(i)}(t_k)}{i!} = f^{[i]}(y_k).$$

In order to compute a priori bounds we use the following procedure: if y_k is in the interior of $\widetilde{Y}_k$, and

$$y_k + \sum_{i=1}^{j-1} (t - t_k)^i f^{[i]}(y_k) + (t - t_k)^j f^{[j]}(\widetilde{Y}_k) \subseteq \widetilde{Y}_k, \quad j \geq 1, \tag{5.17}$$

for all $t \in [t_k, t_{k+1}]$ and all $y_k \in Y_k$, then there exists a unique solution to (5.14) for all $y_k \in Y_k$ and

$$y(t;t_k,y_k) \in y_k + \sum_{i=1}^{j-1}(t-t_k)^i f^{[i]}(y_k) + (t-t_k)^j f^{[j]}(\widetilde{Y}_k)$$

for all $t \in [t_k, t_{k+1}]$ and all $y_k \in Y_k$.

In order to compute tight bounds we use $\widetilde{Y}_k$ and we wish to find an interval Y_{k+1} such that (5.16) holds. Writing a Taylor series expansion, we can compute

$$Y_{k+1} = Y_k + \sum_{i=1}^{j-1} h_k^i f^{[i]}(Y_k) + h_k^j f^{[k]}(\widetilde{Y}_k),$$

where $h_k = t_{k+1} - t_k$, which contains the true solution, but the width of Y_{k+1} is

$$w(Y_{k+1}) \geq w(Y_k),$$

and usually

$$w(Y_{k+1}) > w(Y_k).$$

To obtain a scheme that could follow contracting solutions, the mean-value evaluation is applied: for any $y_k, \hat{y}_k \in Y_k$,

$$\begin{aligned} & y_k + \sum_{i=1}^{j-1} h_k^i f^{[i]}(y_k) + h_k^j f^{[j]}(\widetilde{Y}_k) \subseteq \\ & \subseteq \hat{y}_k + \sum_{i=1}^{j-1} h_k^i f^{[i]}(\hat{y}_k) + h_k^j f^{[j]}(\widetilde{Y}_k) + \left(I + \sum_{i=1}^{j-1} h_k^i \frac{\partial f^{[i]}}{\partial y}(Y_k) \right)(Y_k - \hat{y}_k), \end{aligned} \tag{5.18}$$

where I denotes the $n \times n$ identity matrix. On the basis of (5.18) we can find an interval Y_{k+1} consisting of:

- a point approximation

$$u_{k+1} = \hat{y}_k + \sum_{i=1}^{j-1} h_k^i f^{[i]}(\hat{y}_k),$$

- an enclosure

$$z_{k+1} = h_k^j f^{[j]}(\widetilde{Y}_k)$$

of the truncation error which can be viewed as the excess introduced on the current integration step over the true solution, called *local excess*,

- an enclosure $S_k(Y_k - \hat{y}_k)$, where

$$S_k = I + \sum_{i=1}^{j-1} h_k^j \frac{\partial f^{[i]}}{\partial y}(Y_k),$$

of the propagated *global excess* to t_{k+1}.

Thus, finally we get

$$Y_{k+1} = u_{k+1} + z_{k+1} + S_k(Y_k - \hat{y}_k).$$

If all calculations are executed in floating-point interval arithmetic, the above interval is a rigorous enclosure on the exact solution of the problem (5.13), enclosing a rounding error, too. In [144] a procedure for reducing the wrapping effect is also presented.

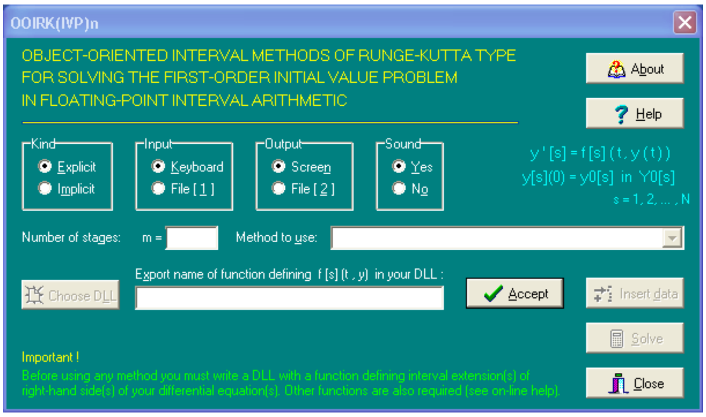
5.2. Software Overview

In conclusion we would like to mention a number of software packages for computing the bounds on the solution of the initial value problem on the basis of many approaches to the Taylor methods. In Table 5.1 we present a list of such packages, together with relevant references.

For the interval methods of Runge-Kutta type (presented in Chapter 3) and for the interval methods of Adams type (discussed in Chapter 4) the packages OOIRK in Delphi Pascal and IMM in C++, respectively, have been developed at the Institute of Computing Science of the Poznań University of Technology. Information on these packages can be found in [82] and [121]. Both packages are available for free from the authors, and their main windows are presented in Figures 5.1 and 5.2.

Table 5.1. Packages for computing bounds in the initial value problem

Package	*Year*	*Reference*	*Language*
AWA	1988	[106]	PASCAL-XSC
ADIODES	1997	[169]	C++
COSY INFINITY	1997	[23, 27, 31, 109]	Fortran, C++ interface
VNODE	2001	[141]	C++
VODESIA	2003	[49]	Fortran-XSC
VSPODE	2005	[104]	C++
ValEncIA-IVP	2005	[9]	C++
VNODE-LP	2006	[143]	C++



OBJECT-ORIENTED INTERVAL METHODS OF RUNGE-KUTTA TYPE
FOR SOLVING THE FIRST-ORDER INITIAL VALUE PROBLEM
IN FLOATING-POINT INTERVAL ARITHMETIC

Kind: ● Explicit ○ Implicit

Input: ● Keyboard ○ File [1]

Output: ● Screen ○ File [2]

Sound: ● Yes ○ No

y ' [s] = f [s] (t , y (t))
y[s](0) = y0[s] in Y0[s]
s = 1, 2, ... , N

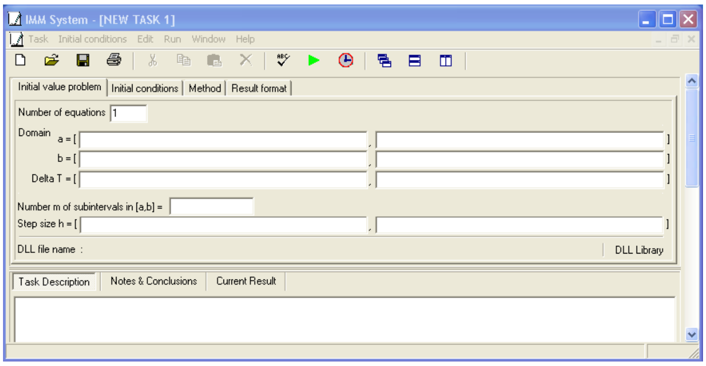
Number of stages: m =

Method to use:

Export name of function defining f [s] (t , y) in your DLL :

Important !
Before using any method you must write a DLL with a function defining interval extension(s) of right-hand side(s) of your differential equation(s). Other functions are also required (see on-line help).

Figure 5.1. The main window of the OOIRK package



Number of equations 1

Domain
a = [,]
b = [,]
Delta T = [,]

Number m of subintervals in [a,b] =

Step size h = [,]

DLL file name :

DLL Library

Figure 5.2. The main window of the IMM package

References

[1] Adams, E., Periodic Solutions: Enclosure, Verification, and Applications, in: Ch. Ullrich (ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical Methods*, Academic Press, San Diego 1990, 199–245.

[2] Adams, E., The Reliability Question for Discretizations of Evolution Problems, Part I: Theoretical Consideration on Failures, Part II: Practical Failures, in: E. Adams, U. Kulisch (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, San Diego 1993, 423–463, 465–526.

[3] Adams, E., Ames, W. F., Kühn, W., Rufeger, W., Spreuer, H., Computational Chaos May Be Due to a Single Local Error, *Journal of Computational Physics* 104 (1993), 241–250.

[4] Adams, E., Scheu, G., Zur numerischen Durchführung eines Iterationsverfahrens für monotone Schrankenfolgen bei nichtlinearen gewöhnlichen Rand- oder Anfangswertaufgaben, *Zeitschrift für Angewandte Mathematik und Mechanik* 56 (1976), T270–T272.

[5] Alefeld, G., Herzberger, J., *Introduction to Interval Computations*, Academic Press, New York 1983.

[6] Ames, W. F., Ginsberg, M., Bilateral Algorithms and their Applications, in: A. Dold, B. Eckmann (eds.), *International Conference on Computational Methods in Nonlinear Mechanics*, Lecture Notes in Mathematics Vol. 461, Springer, Berlin 1974, 1–32.

[7] Ascher, U. M., Petzold, L. R., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia 1998.

[8] Auer, E., Kecskeméthy, A., Tandl, M., Traczinski, H., Interval Algorithms in Modeling of Multibody Systems, in: *Numerical Software with Result Verification*, Lecture Notes in Computer Science Vol. 2991, Springer, Berlin 2004, 132–159.

[9] Auer, E., Rauh, A., Hofer, E. P., Luther, W., Validated Modeling of Mechanical Systems with SmartMOBILE: Improvement of Performance by ValEncIA-IVP, in: P. Hertling, C. H. Hoffmann, W. Luther, N. Revol (eds.), *Reliable Implementation of Real Number Algorithms: Theory and Practice*, Lecture Notes in Computer Science Vol. 5045, Springer, Berlin 2008, 1–27.

[10] Avenhaus, J., Ein Verfahren zur Einschließung der Lösung des Anfangswertproblems, *Computing* 8 (1971), 182–190.

[11] Bachmann, K.-H., Untersuchungen zur Einschließung der Lösungen von Systemen gewöhnlicher Differentialgleichungen, *Beitrage zur Numerischen Mathematik* 1 (1974), 9–42.

[12] Bachmann, K.-H., Fehlereinschließung für Naherungslösungen von Systemen gewöhnlicher Differentialgleichungen, *Zeitschrift für Angewandte Mathematik und Mechanik* 63 (1983), 63–65.

[13] Barrio, R., Performance of the Taylor Series Method for ODEs/DAEs, *Applied Mathematics and Computation* 163 (2005), 525–545.

[14] Barrio, R., Sensitivity Analysis of ODES/DAES Using the Taylor Series Method, *SIAM Journal of Scientific Computing* 27 (2006), 929–1947.

[15] Bashforth, F., Adams, J. C., *An Attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops Fluid, with an Explanation of the Method of*

Integration Employed in Constructing the Tables which Give the Theoretical Forms of Such Drops, Cambridge University Press, Cambridge 1883.

[16] Bauch, H., Zur Lösungseinschließung bei Anfangswertaufgaben gewöhnlicher Differentialgleichungen nach der Defektmethode, *Zeitschrift für Angewandte Mathematik und Mechanik* 57 (1977), 387–396.

[17] Bauch, H., On the Iterative Inclusion of Solutions in Initial-Value Problems for Ordinary Differential Equations, *Computing* 22 (1979), 339–354.

[18] Bauch, H., Zur iterativen Lösungseinschließung bei Anfangswertproblemen mittels Intervallmethoden, *Zeitschrift für Angewandte Mathematik und Mechanik* 60 (1980), 137–145.

[19] Bauch, H., Jahn, K.-U., Oelschlägel, D., Süsse, H., Wiebigke, V., *Intervallmathematik: Theorie und Anwendungen*, Teubner, Leipzig 1987.

[20] Bauch, H., Kimmel, W., Solving Ordinary Initial Value Problems with Guaranteed Bounds, *Zeitschrift für Angewandte Mathematik und Mechanik* 69 (1989), T110–T112.

[21] Bendsten, C., Stauning, O., *TADIFF, a Flexible C++ Package for Automatic Differentiation Using Taylor Series*, Technical Report 1997-x5-94, Department of Mathematical Modeling, Technical University of Denmark, DK-2800, Lyngby 1997.

[22] Berz, M., Differential Algebraic Description of Beam Dynamics to Very High Orders, *Particle Accelerators* 24 (1989), 109.

[23] Berz, M., *COSY INFINITY Version 8 Reference Manual*, Technical Report MSUCL-I088, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1997.

[24] Berz, M., Hoffstätter, G., Exact Bounds of the Long Term Stability of Weakly Nonlinear Systems Applied to the Design of Large Storage Rings, *Interval Computations* 2 (1994), 68–89.

[25] Berz, M., Bischof, C., Corliss, G., Griewank, A. (eds.), *Computational Differentiation: Techniques, Applications and Tools*, SIAM, Philadelphia 1996.

[26] Berz, M., Hoffstätter, G., Wan, W., Shamseddine, K., Makino, K., COSY INFINITY and its Applications to Nonlinear Dynamics, in: M. Berz, C. Bischof, G. Corliss, A. Griewank (eds.), *Computational Differentiation: Techniques, Applications and Tools*, SIAM, Philadelphia 1996, 363–365.

[27] Berz, M., *COSY INFINITY Version 8 – Reference Manual*, Technical Report MSUCL-1088, National Supercomputing Cyclotron Laboratory, Michigan State University, East Lansing 1997.

[28] Berz, M., Hoffstätter, G., Computation and Application of Taylor Polynomials with Interval Remainder Bounds, *Reliable Computing* 4 (1) (1998), 83–97.

[29] Berz, M., Makino, K., Verified Integration of ODEs and Flows with Differential Algebraic Methods on Taylor Models, *Reliable Computing* 4 (4) (1998), 361–369.

[30] Berz, M., Hoefkens, J., Verified High-Order Inversion of Functional Dependencies and Superconvergent Interval Newton Methods, *Reliable Computing* 7 (5) (2001), 379–398.

[31] Berz, M., Makino, K., *COSY INFINITY Version 8.1 – Reference Manual*, Technical Report MSUCL-1195, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing 2001.

[32] Berz, M., Makino, K., Hoetkens, J., Verified Integration of Dynamics in the Solar System, *Nonlinear Analysis* 47 (2001) 179–190.

[33] Berz, M., Hoefkens, J., Makino, K., *COSY INFINITY Version 8.1 – Programming Manual*, Technical Report MSUHEP-20703, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824 (2002).

[34] Berz, M., Makino, K., *COSY INFINITY Version 8.1 – User's Guide and Reference Manual*, Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824 (2002).

[35] Berz, M., Makino, K., Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: Long-term Stabilization by Shrink Wrapping, *International Journal of Differential Equations and Applications* 10 (4) (2005), 385–403.

[36] Berz, M., Makino, K., *performance of Taylor Model Methods for Validated Integration of ODEs*, Lecture Notes in Computer Science Vol. 3732, Springer, Berlin 2005.

[37] Brouver, D., Clemence, G. M., *Methods of Celestial Mechanics*, Academic Press, New York 1961.

[38] Butcher, J. C., *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, Wiley, Chichester 1987.

[39] Chang, Y. F., Corliss, G. F., Solving Ordinary Differential Equations Using Taylor Series, *ACM Transactions on Mathematical Software* 8 (1982), 114–144.

[40] Chang, Y. F., Corliss, G. F., ATOMFT: Solving ODEs and DAEs Using Taylor Series, *Computers & Mathematics with Applications* 28 (1994), 209–233.

[41] Collatz, L., *The Numerical Treatment of Differential Equations*, 3rd Edition, Springer, Berlin 1959.

[42] Collatz, L., *Functional Analysis and Numerical Mathematics*, Academic Press, New York 1966.

[43] Corless, R. M., Corliss, G. F., Rationale for Guaranteed ODE Defect Control, in: L. Atanassova, J. Herzberger (eds.), *Computer Arithmetic and Enclosure Methods*, North-Holland, Amsterdam 1992, 3–12.

[44] Corliss, G. F., Survey of Interval Algorithms for Ordinary Differential Equations, *Applied Mathematics and Computation* 31 (1989), 112–120.

[45] Corliss, G. F., *Proposal for a basic interval arithmetic subroutines library (BIAS)*, Technical Report, Marquette University Department of Mathematics, Statistics and Computer Science, Milwaukee 1991.

[46] Corliss, G. F., Rihm, R., Validating an a Priori Enclosure Using High-Order Taylor Series, in: *Scientific Computing, Computer Arithmetic, and Validated Numerics*, Akademie, Berlin 1996, 228–238.

[47] Dahlquist, G., Convergence and Stability in the Numerical Integration of Ordinary Differential Equations, *Mathematica Scandinavica* 4 (1956), 33–53.

[48] Davey, D. P., Stewart, N. F., Guaranteed Error Bounds for the Initial Value Problem Using Polytope Arithmetic, *BIT* 16 (1976), 257–268.

[49] Dietich, S., *Adaptive verifizierte Lösung gewöhnlicher Differentialgleichungen*, Ph. D. Thesis, University of Karlsruhe, Karlsruhe 2003.

[50] Dormand, J. R., *Numerical Methods for Differential Equations – A Computational Approach*, CRC, Boca Raton 1996.

[51] Duboshin, G. N., *Celestial Mechanics. Analytical and Qualitative Methods* [in Russian], Nauka, Moskva 1978.

[52] Eijgenraam, P., *The Solution of Initial Value Problems Using Interval Arithmetic*, Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, Amsterdam 1981.

[53] Erhel, J., Philippe, B., Design of a toolbox to control arithmetic reliability, in: L. Atanassova, J. Herzberger (eds.), *Computer Arithmetic and Enclosure Methods*, North-Holland, Amsterdam 1992.

[54] Gajda, K., Marciniak, A., Szyszka, B., Three- an Four-Stage Implicit Interval Methods of Runge-Kutta Type, *Computational Methods in Science and Technology* 6 (2000), 41–59.

[55] Gajda, K., *Interval Methods of Symplectic Runge-Kutta Type* [in Polish], Ph. D. Thesis, Poznań University of Technology, Poznań 2004.

[56] Gajda, K., Marciniak, A., Symplectic Interval Methods for Solving Hamiltonian Problems, *Pro Dialog* 22 (2007), 27–37.

[57] Gajda, K., Jankowska, M., Marciniak, A., Szyszka, B., A Survey of Interval Runge-Kutta and Multistep Methods for Solving the Initial Value Problem, in: R. Wyrzykowski, J. Dongmara, K. Karczewski, J. Wasniewski (eds.), *Parallel Processing and Applied Mathematics*, Lecture Notes in Computer Science Vol. 4967, Springer, Berlin 2007, 1361–1371.

[58] Gambill, T. N., Skeel, R. D., Logarithmic Reduction of the Wrapping Effect with Application to Ordinary Differential Equations, *SIAM Journal on Numerical Analysis* 25 (1) (1988), 153–162.

[59] Griewank, A., Walther, A., On the Efficient Generation of Taylor Expansions for DAE Solutions by Automatic Differentiation, in: J. Dongarra, K. Madsen, J. Wasniewski (eds.), *PARA '04, State-of-the-Art in Scientific Computing*, Lecture Notes in Computer Science Vol. 3732, Springer, Berlin 2006, 1103–1111.

[60] Guderley, K. G., Keller, C. L., A Basic Theorem in the Computation of Ellipsoidal Error Bounds, *Numerische Mathematik* 19 (3) (1972), 218–229.

[61] Hammer, R., Hocks, M., Kulisch, U., Ratz, D., *Numerical Toolbox for Verified Computing I. Basic Numerical Problems*, Springer, Berlin 1993.

[62] Heirer, E., Nørsett, S. P., Wanner, G., *Solving Ordinary Differential Equations I – Nonstiff Problems*, Springer, Berlin 1987.

[63] Heirer, E., Wanner, G., *Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems*, Springer, Berlin 1991.

[64] Hansen, E. R., *Topics in Interval Analysis*, Oxford University Press, London 1969.

[65] Hansen, E. R., *Global Optimization Using Interval Analysis*, Marcel Dekker , New York 1992.

[66] Hayes, W., Jackson, K. R., Rigorous Shadowing of Numerical Solutions of Ordinary Differential Equations by Containment, *SlAM Journal on Numerical Analysis* 42 (5) (2003), 1948–1973.

[67] Henrici, P., *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York 1962.

[68] Henrici, P., *Error Propagation in Difference Methods*, Wiley, New York 1963.

[69] Hill, G. W., Researches in the Lunar Theory, *American Journal of Mathematics* 1 (1) (1878), 5–26.

[70] Hoetkens, J., Berz, M., Makino, K., *Efficient High-Order Methods for ODEs and DAEs*, Springer, New York 2001.

[71] Hoetkens, J., Berz, M., Makino, K., *Verified High-Order Integration of DAEs and Higher-Order ODEs*, Kluwer, Dordrecht 2001.

[72] Hofkens, J., Berz, M., Makino, Computing Validated Solutions of Implicit Differential Equations, *Advances in Computational Mathematics* 19 (2003), 231–253.

[73] Hunger, S., Intervallanalytische Defektabscha.tzung zur Lösung mit exakter Fehlererfassung bei Anfangswertaufgaben für Systeme gewöhnlicher Differentialgleichungen, *Zeitschrift für Angewandte Mathematik und Mechanik* 52 (1972), T208–T209.

[74] Jackson, K. R., Nedialkov, N. S., Some Recent Advances in Validated Methods for IVPs for ODEs, *Applied Numerical Analysis and Computational Mathematics* 42 (2002), 269–284.

[75] Jackson, L. W., *A Comparison of Ellipsoidal and Interval Arithmetic Error Bounds, Numerical Solutions of Nonlinear Problems*, SIAM, Philadelphia 1968.

[76] Jackson, L. W., Interval Arithmetic Error-Bounding Algorithms, *SIAM Journal on Numerical Analysis* 12 (2) (1975), 223–238.

[77] Jain, M. K., *Numerical Solution of Differential Equations*, Wiley, New York 1979.

[78] Jankowska, J., Jankowski, M., *Survey of Numerical Methods and Algorithms. Part I* [in Polish], WNT, Warsaw 1981.

[79] Jankowska, M., Marciniak, A., Implicit Interval Multistep Methods for Solving the Initial Value Problem, *Computational Methods in Science and Technology* 8 (1) (2002), 17–30.

[80] Jankowska, M., Marciniak, A., On Explicit Interval Methods of Adams-Bashforth Type, *Computational Methods in Science and Technology* 8 (2) (2002), 46–57.

[81] Jankowska, M., Marciniak, A., On Two Families of Implicit Interval Methods of Adams--Moulton Type, *Computational Methods in Science and Technology* 12 (2) (2006), 109–114.

[82] Jankowska, M., Marciniak, A., Preliminaries of the IMM System for Solving the Initial Value Problem by Interval Multistep Methods [in Polish], *Pro Dialog* 10 (2005), 117–134.

[83] Jankowska, M, *Interval Multistep Methods of Adams Type and Their Implementation in the C++ Language*, Ph. D. Thesis, Poznań University of Technology, Poznań 2006.

[84] Jankowska, M., Marciniak, A., An Interval Version of the Backward Differentiation (BDF) Method, in: *SCAN 2006 Conference Post-Proceedings*, IEEE-CPS Product No. E2821 (2007).

[85] Jaulin, L., Kieffer, M., Didrit, O., Walter, É., *Applied Interval Analysis*, Springer, London 2001.

[86] Kalmykov, S. A., Shokin, Ju. I., Juldashev, E. C., *Methods of Interval Analysis* [in Russian], Nauka, Novosibirsk 1986.

[87] Kalmykov, S. A., Shokin, Ju. I., Juldashev, E. C., *Solving Ordinary Differential Equations by Interval Methods* [in Russian], Doklady AN SSSR Vol. 230, No. 6, 1976.

[88] Kaucher, E. W., Miranker, W. L., *Self-Validating Numerics for Function Space Problems*, Academic Press, Orlando 1984.

[89] Kerbl, M., Stepsize Strategies for Inclusion Algorithms for ODE's, in: E. Kaucher, S. Markov, G. Mayer (eds.), *Computer Arithmetic, Scientific Computation, and Mathematical Modeling*, IMACS Annals on Computing and Applied Mathematics 12, J. C. Baltzer, Basel 1991.

[90] Kieffer, M., Walter, E., Nonlinear Parameter and State Estimation for Cooperative Systems in a Bounded-Error Context, in: *Numerical Software with Result Verification*, Lecture Notes in Computer Science Vol. 2991, Springer, Berlin 2004, 107–123.

[91] Kincaid, D., Cheney, W., *Numerical Analysis – Mathematics of Scientific Computing*, Third Edition, Brooks/Cole, 2002.

[92] Kirlinger, G., Corliss, G. F., On Implicit Taylor Series Methods for Stiff ODEs, in: L. Atanassova, J. Herzberger (eds), *Computer Arithmetic and Enclosure Methods*, North-Holland, Amsterdam 1992, 371–379.

[93] Klatte, R., Kulisch, U., Neaga, M., Ratz, D., Ullrich, Ch., *Pascal-XSC: Language Reference with Examples*, Springer, Berlin 1992.

[94] Klatte, R., Kulisch, U., Lawo, C., Rauch, M., Wiethoff, A., *C-XSC – A C++ Class Library for Extended Scientific Computing*, Springer, Heidelberg 1993.

[95] Kletting, M., Rauh, A., Aschemann, H., Hofer, E., Consistency Tests in Guaranteed Simulation of Nonlinear Uncertain Systems with Application to an Activated Sludge Process, *Journal of Computational and Applied Mathematics* 199 (2) (2007), 213–219.

[96] Krämer, W., von Gudenberg, J. W. (eds.), *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer, New York 2001.

[97] Krupowicz, A., *Numerical Methods of Initial Value Problems of Ordinary Differential Equations* [in Polish], PWN, Warsaw 1986.

[98] Krückeberg, F., Ordinary Differential Equations, in: E. Hansen (ed.), *Topics in Interval Analysis*, Clarendon Press, Oxford 1969, 91–97.

[99] Krückeberg, F., Leisen, R., Solving Initial Value Problems of Ordinary Differential Equations to Arbitrary Accuracy with Variable Precision Arithmetic, in: *Proceedings of the 11th IMACS World Congress on System Simulation and Scientific Computing*, Vol. 1, Oslo 1985.

[100] Knüppel, O., PROFIUBIAS – A Fast Interval Library, *Computing* 53 (3–4) (1994), 277–287.

[101] Kutta, W., Beitrag zur näherungsweisen Integration totaler Differentiagleichungen, *Zeitschrift für Mathematik und Physik* 46 (1901), 435–453.

[102] Lawo, C., C-XSC, A Programming Environment for Verified Scientific Computing and Numerical Data Processing, in: E. Adams, U. Kulisch (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, Orlando 1992.

[103] Lerch, M., Tischler, G., von Gudenberg, J. W., *FILIB++ – Interval Library Specification and Reference Manual*, Technical Report 279, Universität Würzburg, 2001.

[104] Lin, Y., Stadtherr, M. A., Validated Solution of Initial Value Problems for ODEs with Interval Parameters, in: R. L. Muhanna, R. L. Mullen (eds.), *Proceedings of 2nd NSF Workshop on Reliable Engineering Computing*, Savannah 2006.

[105] Lohner, R. J., Enclosing the Solution of Ordinary Initial and Boundary Value Problems, in: E. Kaucher, U. Kulisch, Ch. Ullrich (eds.), *Computer Arithmetic: Scientific Computation and Programming Languages*, B.G. Teubner, Stuttgart 1987, 255–286.

[106] Lohner, R. J., *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Ph. D. Thesis, University of Karlsruhe, Karlsruhe 1988.

[107] Lohner, R. J., Computations of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems, in: I. G. J. Cash (ed.), *Computational Ordinary Differential Equations*, Clarendon Press, Oxford 1992, 425–435.

[108] Lohner, R. J., Interval Arithmetic in Staggered Correction Format, in: E. Adams, U. Kulisch, (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, San Diego 1993, 301–321.

[109] Makino, K., Berz, M., COSY INFINITY Version 7, in: *Fourth Computational Accelerator Physics Conference*, AIP Conference Proceedings, 1996.

[110] Makino, K., Berz, M., Remainder Differential Algebras and Their Applications, in: *Computational Differentiation: Techniques, Applications, and Tools*, SIAM, Philadelphia 1996.

[111] Makino, K., Berz, M., Efficient Control of the Dependency Problem Based on Taylor Model Methods, *Reliable Computing* 5 (1) (1999), 3–12.

[112] Makino, K., Berz, M., *The Method of Shrink Wrapping for the Validated Solution of ODEs*, Technical Report MSUHEP-20510, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2002.

[113] Makino, K., Berz, M., Taylor Models and Other Validated Functional Inclusion Methods, *International Journal of Pure and Applied Mathematics* 6(3) (2003), 239–316.

[114] Makino, K, Bertz, M., Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: Long-term Stabilization by Preconditioning, *International Journal of Differential Equations and Applications* 10 (4) (2005), 353–384.

[115] Makino, K., Berz, M., Suppression of the Wrapping Effect by Taylor Model-based Verified Integrators: The Single Step, *International Journal of Differential Equations and Applications* 36 (2) (2006), 175–197.

[116] Mannshardt, R., Enclosing a Solution of an Ordinary Differential Equation by Sub- and Superfunctions, in: Ch. Ullrich (ed.), *Contributions to Computer and Self-Validating Numerical Methods*, IMACS Annals on Computing and Applied Mathematics 7, J. C. Baltzer, Basel 1990, 319–328.

[117] Marciniak A., *Numerical Solutions of the N-body Problem*, Reidel, Dordrecht 1985.

[118] Marciniak A., *Selected Numerical Methods for Solving the N-body Problems* [in Polish], Publishig House of Poznań University of Technology, Poznań 1989.

[119] Marciniak, A., 0.1 [in Polish], *Pro Dialog* 5 (1997), 55–82.

[120] Marciniak A., Marlewski, A., On Interval Representation of Non-Machine Numbers in Object Pascal [in Polish], *Pro Dialog* 7 (1998), 75–100.

[121] Marciniak, A., Gajda K., Marlewski A., Szyszka B., A Layout of an Object-Oriented System for Solving the Initial Value Problem by Interval Methods of Runge-Kutta Type [in Polish], *Pro Dialog* 8 (1999), 39–62.

[122] Marciniak, A., Szyszka, B., One- and Two-Stage Implicit Interval Methods of Runge-Kutta Type, *Computational Methods in Science and Technology* 5 (1999), 53–65.

[123] Marciniak, A., Finding the Integration Interval for Interval Methods of Runge-Kutta Type in Floating-Point Interval Arithmetic, *Pro Dialog* 10 (2000), 35–45.

[124] Marciniak, A., Szyszka, B., On Representation of Coefficients in Implicit Interval Methods of Runge-Kutta Type, *Computational Methods in Science and Technology* 10 (1) (2004), 57–71.

[125] Marciniak, A., Implicit Interval Methods for Solving the Initial Value Problem, *Numerical Algorithms* 37 (2004), 241–251.

[126] Marciniak A., On Computational Complexity of Some Interval Methods for Solving the Initial Value Problem [in Polish], *Pro Dialog* 20 (2005), 55–70.

[127] Marciniak, A., Multistep Interval Methods of Nyström and Milne-Simpson Types, *Computational Methods in Science and Technology* 13 (1) (2007), 23–40.

[128] Marciniak, A., On Multistep Interval Methods for Solving the Initial Value Problem, *Journal of Computational and Applied Mathematics* 199 (2) (2007), 229–238.

[129] Milne, W. E., Numerical Integration of Ordinary Differential Equations, *The American Mathematical Monthly* 33 (1926), 455–460.

[130] Milne, W. E., *Numerical Solution of Differential Equations*, Wiley, New York 1953.

[131] Moore, R. E., The Automatic Analysis and Control of Error in Digital Computation Based on the Use of Interval Numbers, in: L. B. Rall (ed.), *Error in Digital Computation*, Vol. 1, Wiley, New York 1965, 61–130.

[132] Moore, R. E., Automatic Local Coordinate Transformations to Reduce the Growth of Error Bounds in Interval Computation of Solutions of Ordinary Differential Equations, in: L. B. Rall (ed.), *Error in Digital Computation*, Vol. 2, Wiley, New York 1965, 103–140.

[133] Moore, R. E., *Interval Analysis*, Prentice-Hall, Englewood Cliffs 1966.

[134] Moore, R. E., *Methods and Applications of Interval Analysis*, SIAM, Philadelphia 1979.

[135] Moore, R. E., A Survey of Interval Methods for Differential Equations, in: *Proceedings of the 23rd Converence on Decision and Control*, Las Vegas 1984, IEEE 1984, 1529–1535.

[136] Mukundan, H., Ko, K. H., Maekawa, T., Sakkalis, T., Patrikalakis, N. M., Tracing Surface Intersections with a Validated ODE System Solver, in: G. Elber, G. Taubin (eds.), *Proceedings of the Ninth EG/ACM Symposium on Solid Modeling and Applications*, Eurographics Press, 2004.

[137] Nedialkov, N. S., Jackson, K. R., Corliss, G. F., Validated Solutions of Initial Value Problems for Ordinary Differential Equations, *Applied Mathematics and Computation* 105 (1) (1999) 21–68.

[138] Nedialkov, N. S., Jackson, K. R., An Interval Hermite-Obreschkoff Method for Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation, *Reliable Computing* 5 (3) (1999), 289–310.

[139] Nedialkov, N. S., Jackson, K. R., A New Perspective on the Wrapping Effect in Interval Methods for Initial Value Problems for Ordinary Differential Equations, in: A. Facius, U. Kulisch, R. Lohner (eds.), *Perspectives on Enclosure Methods*, Springer, Vienna 2001, 219–264.

[140] Nedialkov, N. S., Jackson, K. R., Pryce, J. D., An Effective High-Order Interval Method for Validating Existence and Uniqueness of the Solution of an IVP for an ODE, *Reliable Computing* 7 (2001), 449–465.

[141] Nedialkov, N. S., Jackson, K. R., *The Design and Implementation of a Validated Object--Oriented Solver for IVPs for ODEs*, Technical Report 6, Software Quality Research Laboratory, Department of Computing and Software, McMaster University, Harnilton 2002.

[142] Nedialkov, N. S., Pryce, J. D., Solving Differential-Algebraic Equations by Taylor Series (I): Computing Taylor Coefficients, *BIT* 45 (2005), 561–591.

[143] Nedialkov, N., S., *VNODE-LP – A Validated Solver for Initial Value Problems in Ordinary Differential Equations*, Technical Report CAS 06-06-NN, Department of Computing and Software, McMaster University, Harnilton 2006.

[144] Nedialkov, N. S., *Interval Tools for ODEs and DAEs*, Technical Report CAS 06-09-NN, Department of Computing and Software, McMaster University, Hamilton 2006.

[145] Nedialkov, N. S., Pryce, J. D., Solving Differential-Algebraic Equations by Taylor Series (II): Computing the System Jacobian, *BIT* 47 (1) (2007), 121–135.

[146] Neumaier, A., *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge 1990.

[147] Nickel, K. L. E., Intervall-Mathematik, *Zeitschrift für Angewandte Mathematik und Mechanik* 58 (1978), T72–T85.

[148] Nickel, K. L. E., How to Fight the Wrapping Effect, in: K. L. E. Nickel (ed.), *Interval Analysis*, Lecture Notes in Computer Science Vol. 212, Springer, Berlin 1985, 121–132.

[149] Nickel, K. L. E., Using Interval Methods for the Numerical Solution of ODE's, *Zeitschrift für Angewandte Mathematik und Mechanik* 66 (1986), 513–523.

[150] Nyström, E. J., Über die numerische Integration von Differentialgleichungen, *Acta Societas Scientiarum Fennicae* 50 (13) (1925).

[151] Poincaré, A., *New Methods of Celestial Mechanics*, Vol. 1, American Institute of Physics, New York 1993.

[152] Pryce, J. D., Solving High-Index DAEs by Taylor Series, *Numerical Algorithms* 19 (1998), 195–211.

[153] Pryce, J. D., A Simple Structural Analysis Method for DAEs, *BIT* 41 (2) (2001), 364–394.

[154] Rall, L. B., *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science Vol. 120, Springer, Berlin 1981.

[155] Rall, L. B., Optimal Implementation of Differentiation Arithmetic, in: E. Kaucher, U. Kulisch, Ch. Ullrich (eds.), *Computer Arithmetic: Scientific Computation and Programming Languages*, B. G. Teubner, Stuttgart 1987, 287–295.

[156] Rall, L. B., Differentiation Arithmetics, in: Ch. Ullrich (ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical Methods*, Academic Press, San Diego 1990, 73–90.

[157] Revol, N., Makino, K., Berz, M., Taylor Models and Floating Point Arithmetic: Proof That Arithmetic Operations Are Bounded in COSY, *Journal of Logic and Algebraic Programming* 64 (2005), 135–154.

[158] Rihm, R., LösungseinschlieBung bei gewöhnlichen Differentialgleichungen mit Unstetigkeiten in der rechten Seite, *Zeitschrift für Angewandte Mathematik und Mechanik* 71 (1991), T795–T797.

[159] Rihm, R., Enclosing Solutions with Switching Points in Ordinary Differential Equations, in: L. Atanassova, J. Herzberger (eds.), *Computer Arithmetic and Enclosure Methods*, North-Holland, Amsterdam 1992, 419–425.

[160] Rihm, R., Einschlie6ung von Lösungen mit Schaltpunkten bei gewöhnlichen Anfangswertproblemen, *Zeitschrift für Angewandte Mathematik und Mechanik* 73 (1993), T815–T817.

[161] Rihm, R., Über eine Klasse von Einschlie6ungsverfahren für gewöhnliche Anfangswertprobleme, *Zeitschrift für Angewandte Mathematik und Mechanik* 74 (1994), T685–T687.

[162] Rihm, R., Interval Methods for Initial Value Problems in ODEs, in: J. Herzberger (ed.), *Topics in Validated Computations*, Elsevier, Amsterdam 1994, 173–207.

[163] Rihm, R., Implicit Methods for Enclosing Solutions of ODEs, *Journal of Universal Computer* Science 4 (2) (1998), 202–216.

[164] Rufeger, W., Adams, E., A Step Size Control for Lohner's Enclosure Algorithm for Ordinary Differential Equations with Initial Conditions, in: E. Adams, U. Kulisch (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, San Diego 1993, 283–299.

[165] Runge, C., Ueber die numerische Auflösung von Differentialgleichungen, *Mathematische Annalen* 46 (1985), 167–178.

[166] Sanz-Serna, J. M., Calvo, M. P., *Numerical Hamiltonian Problems*, Chapman & Hall, London 1994.

[167] Shokin, Yu. I., *Interval Analysis* [in Russian], Nauka, Novosibirsk 1981.

[168] Siegel, C. L., Moser, J. K., *Lectures on Celestial Mechanics*, Springer, Berlin 1971.

[169] Stauning, O., *Automatic Validation of Numerical Solutions*, Ph. D. Thesis, Technical University of Denmark, Lyngby 1997.

[170] Stetter, H. J., *Analysis of Discretization Methods for Ordinary Differential Equations*, Springer, Berlin 1973.

[171] Stetter, H. J., Algorithms for the Inclusion of Solutions of Ordinary Initial Value Problems, in: J. Vosmanský, M. Zlámal (eds.), *Equadiff 6: Proceedings of the International Conference on Differential Equations and their Applications*, Lecture Notes in Mathematics Vol. 1192, Springer, Berlin 1986, 85–94.

[172] Stetter, H. J., Validated Solution of Initial Value Problems for ODE, in: Ullrich, Ch. (Ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical Methods*, Academic Press, San Diego 1990, 171–186.

[173] Stewart, N. F., A Heuristic to Reduce the Wrapping Effect in the Numerical Solution of $x' =$ $= f(t, x)$, *BIT* 11 (1971), 328–337.

[174] Stoer, J., Bulirsch, R., *Introduction to Numerical Analysis*, Springer, Berlin 1983.

[175] Szyszka, B., *Implicit Interval Methods of Runge-Kutta Type* [in Polish], Ph. D. Thesis, Poznań University of Technology, Poznań 2003.

[176] *The IEEE-754 Standard for Binary Floating-Point Arithmetic*, Institute of Electrical and Electronics Engineers, 1985.

[177] Ullrich, Ch. (ed.), *Computer Arithmetic and Self-Validating Numerical Methods*, Academic Press, San Diego 1990.

[178] Ullrich, Ch. (ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical Methods*, IMACS Annals on Computing and Applied Mathematics 7 , J. C. Baltzer, Basel 1990.

[179] Walter, W .V., ACRITH-XSC: A FORTRAN-like Language for Verified Scientific Computing, in: E. Adams, U. Kulisch (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, Orlando 1992.

[180] Wierzbiński, S., *Celestial Mechanics* [in Polish], PWN, Warsaw 1973.

Index