International Journal of Pure and Applied Mathematics

Volume 45 No. 2 2008, 193-204

APPROXIMATE RATIONAL ARITHMETICS AND ARBITRARY PRECISION COMPUTATIONS FOR UNIVERSAL ALGORITHMS

G.L. Litvinov[1 §], A.Ya. Rodionov[2], A.V. Tchourkin[3]

[1]Department of Mathematics
Independent University of Moscow
Bolshoi Vlasievskii Per., 11, Moscow, 119002, RUSSIA
e-mail: glitvinov@gmail.com

[2]Research Institute for Nuclear Physics
Moscow State University
Vorobjevy Gory, Moscow, 119992, RUSSIA
e-mail: ayarodionov@yahoo.com

[3]Department of Physics
Moscow State University
Vorobjevy Gory, Moscow, 119992, RUSSIA
e-mail: churandr@mail.ru

Abstract: We will describe an approximate rational arithmetic with round-off errors (both absolute and relative) controlled by the user. The rounding procedure is based on the continued fraction expansion of real numbers. Results of computer experiments are given in order to compare efficiency and accuracy of different types of approximate arithmetics and rounding procedures. Relations with universal algorithms and generic programming are briefly discussed.

AMS Subject Classification: 11Y65, 65G50, 68Q25, 68Q65
Key Words: approximate rational arithmetic, continued fractions, round-off errors, multiple and arbitrary precision computations, universal algorithms

Received: February 12, 2008

§Correspondence author

1. Introduction

1.1. Floating Point Arithmetics: Disadvantages

Problems of validity and reliability of calculations (including the analysis of round-off errors) have become more and more important recently, partly due to the steady growth of computer power. Roughly speaking, the main disadvantage of the standard floating point arithmetic is that relative round-off error only can be controlled during calculations. In some cases (e.g., for summation of series and subtraction of nearly equal numbers) this disadvantage can lead to a loss of accuracy and even to absolutely incorrect results. So, if the result of calculations depends critically on the errors in input data and round-off errors (for example, in the case of solving ill-posed equations, the study of stability of solutions etc.), then it is reasonable to use calculations with multiple and even arbitrary precision.

In fact it is often desirable to perform computations with variable (and arbitrary) accuracy. For this purpose, algorithms are required to be independent of the accuracy of computations and of particular computer representations of numbers. Moreover, many important algorithms are not only independent of computer representations of numbers, but also of concrete mathematical (algebraic) operations on data. In this case, operations may be considered as variables. Such algorithms are implemented by *generic programs* based on the *abstract data types* technique (abstract data types are defined by the user, in addition to predefined types of the language used), see, e.g., [1], [14].

Algorithms of this type (*universal algorithms*) are typical for the so-called idempotent (tropical) mathematics, see, e.g., [8], [6], [7]. In this paper we will discuss an approximate rational arithmetic of arbitrary precision. This arithmetic is useful and convenient for implementations of universal algorithms.

1.2. Universal Algorithms

An algorithm is called *universal* if it is independent of a particular numerical domain and (or) of its computer representation. A typical example of a universal algorithm is computation of the scalar product (x, y) of two vectors $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ by the formula $(x, y) = x_1y_1 + \cdots + x_ny_n$. This algorithm (formula) is independent of a particular domain and its computer implementation, since the formula is defined for any semiring. It is clear that one algorithm can be more universal than another. For example, the sim-

plest rectangular formula provides the most universal algorithm for numerical integration; indeed, this formula is valid even for idempotent integration (over any idempotent semiring [8], [6], [7]). Other quadrature formulas (e.g., combined trapezoid or Simpson formulas) do not depend on computer arithmetics and can be used (e.g., in the iterative form) for computations with arbitrary accuracy. In contrast, algorithms based on Gauss–Jacobi formulas are designed for fixed accuracy computations: they include constants (coefficients and nodes of these formulas) defined with fixed accuracy.

As a rule, iterative algorithms (beginning with the successive approximation method) for solving differential equations (e.g., methods of Euler, Euler–Cauchy, Runge–Kutta, Adams, a number of important versions of the difference approximation method, and the like), methods for calculating elementary and some special functions based on the expansion in Taylor's series and continuous fractions (Padé approximations) and others do not depend on the computer representation of numbers.

Computer algebra algorithms used in such systems as *Mathematica, Matlab, Maple, REDUCE*, and others are highly universal. Standard algorithms used in linear algebra can be presented as universal, see, e.g., [8].

1.3. Rational Arithmetics

An appealing way to improve the accuracy of calculations is to use different versions of rational arithmetics, which work with rational numbers of the form $\frac{p}{q}$, where p and q are integer numbers ($q > 0$). It is possible to use the exact rational arithmetic (see, e.g., [2]), but, as a rule, it leads to an explosive growth of both calculation time and storage space since magnitudes (and lengths) of numerators and denominators of computed numbers grow very fast. The approximate rational arithmetics with fixed slash (maximum of lengths of the numerator and the denominator is fixed) or floating slash (the sum of these lengths is fixed) were investigated in detail earlier (see [11], [12]). These rounding procedures use the representation of continued fractions. There also exist rougher rounding procedures, which use only a fixed number of top digits (the other digits are replaced by zeros) but it sometimes leads to relatively large rounding errors.

Here we suggest a new modification of approximate rational arithmetic with a more natural and accurate rounding procedure. The user defines values Δ and δ of absolute and relative error such that $0 \le \Delta \le \infty$, $0 \le \delta \le \infty$. In particular, in the case $\Delta = \delta = 0$ we obtain the exact rational arithmetic. If

$\delta = \infty$, then only absolute rounding error Δ is fixed. If only relative error δ is fixed, then we obtain approximately the same picture as for the floating-point arithmetic. This rounding procedure is applied to a fraction if lengths of its numerator and denominator exceed a number M as specified by the user. In this case the initial rational number is replaced by its best approximation in the form of a convergent of a continued fraction within given errors Δ and δ. The result of rounding is always an uncancellable fraction, and sometimes it can coincide with the initial number (this is our crucial point).

This type of arithmetic was originally implemented by means of the *REDUCE* computer algebra system and was used for constructing arbitrary rational approximations to functions of one variable [4]. In the present paper an analysis of accuracy and efficiency of different modifications of approximate rational arithmetics is based on computer experiments, which are implemented by means of the C^{++} language. Note that Yu.V. Matijasevich suggested applying an approximate rational arithmetic of arbitrary precision for his *a posteriori interval analysis* [9], [10].

Below we describe an algorithm of rounding and constructing of best approximations (using the convergents of continued fractions). This method is compared with other methods by means of a computer experiment. We give also an estimation for the number of components of continued fractions depending on the accuracy of rounding. In particular, we show that the increasing of calculation accuracy does not lead to an explosive growth in calculation time. The main result of computer experiments is that this algorithm of rounding provides significantly higher accuracy of calculations in comparison with other modifications of rational arithmetics that require comparable time.

2. Continued Fractions and the Approximate Rational Arithmetic

Recall some basic notions of the theory of continued fractions. Denote by $[a_0; a_1, a_2, \dots]$ a continued fraction of the form

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}. \tag{1}$$

Any non-negative rational number $\frac{p}{q}$ has a unique canonical representation in the form of a finite continued fraction

$$\frac{p}{q} = [a_0; a_1, a_2, \dots, a_n], \tag{2}$$

where all a_i $(i = 0, \dots, n)$ are non-negative integers, $a_i \geq 1$ if $i = 1, \dots, n-1$ and $a_n \geq 2$ if $n \geq 1$. Irrational numbers can also be represented in the form (1)

as infinite continued fractions. *A convergent* $\frac{p_k}{q_k}$ *of the order* k *of a continued fraction* is defined for the decomposition (2) by the equality

$$\frac{p_k}{q_k} = [a_0; a_1, a_2, \ldots, a_k], \tag{3}$$

where $k \leq n$. It is clear from (3) that the convergent of a continued fraction $\frac{p_n}{q_n}$ coincides with $\frac{p}{q}$. From the theory of continued fractions [3] it is well known that the convergent of a continued fraction (3) is a best approximant to the number (2) in the following sense: for any fraction $\frac{r}{s}$ such that $0 < s \leq q_k$ and $\frac{r}{s} \neq \frac{p_k}{q_k}$, the following inequality holds:

$$\left|\frac{r}{s} - \frac{p}{q}\right| > \left|\frac{p_k}{q_k} - \frac{p}{q}\right|.$$

The only (trivial) counterexample is $\frac{p}{q} = a_0 + \frac{1}{2}$; in this case a_0 and $a_0 + 1$ approximate the number p/q equally well. For any convergent $\frac{p_k}{q_k}$ in the case of $k \neq 0$ and $k < n$, the following inequality holds:

$$\frac{1}{q_k(q_k + q_{k+1})} < \left|\frac{p}{q} - \frac{p_k}{q_k}\right| \leq \frac{1}{q_k q_{k+1}}\,. \tag{4}$$

A similar property is correct for infinite continued fractions of the form (1) corresponding to irrational numbers. Using the inequality (4) we obtain an efficient algorithm for estimation of the rounding error. These properties of convergents make continued fractions an ideal tool for construction of approximate rational arithmetics.

Recall the following algorithm for constructing of convergents of a continued fraction [3]. Let the initial fraction be $\frac{p}{q}$, where p, q are integer numbers, $p > 0$, $q > 1$ (if $p/q < 0$, then we work with the fraction $|p/q|$ and then multiply result by -1).

— Let the initial condition be defined by $b_{-2} = p$, $b_{-1} = q$, $p_{-2} = 0$, $p_{-1} = 1$, $q_{-2} = 1$, $q_{-1} = 0$.

— For $i = 1, 2, \ldots,$ the values of a_i and b_i are consecutively computed as the quotient and the remainder obtained when b_{i-2} is divided by b_{i-1} respectively:

$$b_{i-2} = a_i b_{i-1} + b_i\,.$$

— The numerator and the denominator of the convergent of order i of the continued fraction are given in the recurrent form:

$$p_i = a_i p_{i-1} + p_{i-2},$$

$$q_i = a_i q_{i-1} + q_{i-2}.$$

— If $b_i = 0$, then the convergent of the continued fraction coincides with the initial fraction $\frac{p}{q}$ and the procedure terminates.

— At each step (at each $i = 0, 1, ..$), a criterion of accuracy (see below) is checked, and if the result satisfies the criterion of accuracy then the procedure terminates, otherwise we perform the next step with $i := i + 1$.

As a criterion of accuracy we can choose one of the following conditions:

1) the absolute error is less than Δ;

2) the relative error is less than δ;

3) both conditions 1) and 2) are satisfied.

Note that inequality (4) makes it possible to check the absolute error without a direct comparison with the initial number. This algorithm of rounding can be applied to a result of any arithmetic operation if lengths of its numerator and denominator exceed a given threshold M. Of course, the values of parameters Δ, δ, M are given by the user.

3. Estimations for Round-Off Errors

As a consequence of the recurrence formula for the denominator the values q_k are minimal for each fixed k, if $a_i = 1$ for $i = 0, 1, \ldots, k$, i.e. $q_i = q_{i-1} + q_{i-2}$, $q_{-2} = 1$, $q_{-1} = 1$, $q_0 = q_{-1} + q_{-2} = 1$. Therefore for any convergent $\frac{p_k}{q_k}$ we obtain the inequality

$$q_k \geq F_{k+1}, \tag{5}$$

where F_k are Fibonacci numbers defined by the recurrence formulas $F_k = F_{k-1} + F_{k-2}$ $(k \geq 2)$, where $F_0 = 0$, $F_1 = 1$. It is well-known that Fibonacci numbers are expressed by the formula:

$$F_k = \frac{1}{\sqrt{5}}(\Phi^k - \hat{\Phi}^k), \tag{6}$$

where $\Phi = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$ ("golden section"), $\hat{\Phi} = \frac{1}{2}(1 - \sqrt{5}) \approx -0.618$. It is also well-known that the convergence of the continued fraction expansion of Φ is the slowest among other numbers. From (5) and (6) it follows that

$$q_k q_{k+1} \geq F_{k+1}F_{k+2} = \frac{1}{5}(\Phi^{k+1} - \hat{\Phi}^{k+1})(\Phi^{k+2} - \hat{\Phi}^{k+2})$$
$$= \frac{1}{5}(\Phi^{2k+1} - (\Phi\hat{\Phi})^{k+1}(\Phi + \hat{\Phi}) + \hat{\Phi}^{2k+1}) = \frac{1}{5}(\Phi^{2k+3} + (-1)^k + \hat{\Phi}^{2k+3}),$$

since $\Phi\hat{\Phi} = -1$ and $\Phi + \hat{\Phi} = 1$. Using the values of Φ and $\hat{\Phi}$, we obtain for arbitrary k the estimations $q_k q_{k+1} > \frac{1}{5}\Phi^{2k+2}$, while for even k we have $q_k q_{k+1} > \frac{1}{5}\Phi^{2k+3}$. From these estimates and inequalities (4) it follows that the

absolute round-off error is less than Δ if

$$k \geq \frac{1}{2}\log_\Phi \frac{5}{\Delta} - 1$$

for even k, and

$$k \geq \frac{1}{2}\log_\Phi \frac{5}{\Delta} - \frac{3}{2}$$

for odd k.

4. Estimations for the Number of Iterations

These relations lead to upper estimations for the number of iterations required for the approximation of a rational fraction with an absolute error smaller than Δ. Let $\frac{p_k}{q_k}$ be the convergent of a continued fraction within the required error and the convergent $\frac{p_{k-1}}{q_{k-1}}$ does not give the required accuracy yet. Thus k is the number of iterations necessary to obtain the required accuracy.

For any real number r, the symbol $\lfloor r \rfloor$ will denote the floor of r and $\lceil r \rceil$ will denote the ceiling of r. So $\lceil r \rceil$ is the least integer that is greater or equal to r; similarly, $\lfloor r \rfloor$ is the integer part of r, i.e. the largest integer that is less or equal to r. Hence if r is an integer number, then $\lfloor r \rfloor = \lceil r \rceil$ and otherwise $\lceil r \rceil = \lfloor r \rfloor + 1 = \lfloor r + 1 \rfloor$. Then the required upper estimate has the following form:

$$k \leq \lceil \frac{1}{2}\log_\Phi \frac{5}{\Delta} - 1 \rceil \leq \lfloor \frac{1}{2}\log_\Phi \frac{5}{\Delta} \rfloor. \tag{7}$$

But if the number $\lceil \frac{1}{2}\log_\Phi \frac{5}{\Delta} - \frac{3}{2} \rceil$ is even, the estimation (7) can be strengthened:

$$k \leq \lceil \frac{1}{2}\log_\Phi \frac{5}{\Delta} - \frac{3}{2} \rceil \leq \lfloor \frac{1}{2}\log_\Phi \frac{5}{\Delta} - \frac{1}{2} \rfloor \tag{8}$$

In the case of $\Delta = 10^{-N}$ the estimations (7) and (8) give the following result.

Theorem 1. *If an absolute error specified in the criterion of accuracy of rounding has the form* $\Delta = 10^{-N}$, *then*

$$k \leq \lfloor a + bN \rfloor, \tag{9}$$

where $a = \frac{1}{2}\log_\Phi 5 \approx 1.672$ *and* $b = \frac{1}{2}\log_\Phi 10 \approx 2.392$. *If the number* $\lceil a + bN - \frac{3}{2} \rceil$ *is even, the estimation (9) can be strengthened:*

$$k \leq \lfloor a - \frac{1}{2} + bN \rfloor. \tag{10}$$

For example, if $N = 8$, then the estimation (9) shows that $k \leq 20$; for $N = 9$ this estimation gives $k \leq 23$, but in this case the estimation (10) is applicable, so $k \leq 22$.

Note that these estimations depend only on the absolute error Δ and do not

depend on the initial (i.e. rounded) numbers. In fact (see below), the number of iterations is usually much less than right-hand sides of these inequalities. Since the number of iterations is estimated by a linear function of logarithm of absolute error, an increase in the accuracy of calculations does not lead to an explosive growth of calculation time.

Heuristically, it is easy to estimate the *mean value* $\overline{k}$ of parameter k for a fixed absolute error Δ. Consider a convergent of a continued fraction $\frac{p_k}{q_k}$ as an approximation to a real number x. A.Ya. Khinchin investigated the convergents of continued fraction $\frac{p_k}{q_k}$ for real numbers and proved that for almost all x the equation $\lim_{k\to\infty} \sqrt[k]{q_k} = \gamma$ is valid, where γ is a constant (see [3]). P. Levy (see [5], p. 320), showed that $\ln\gamma = \frac{\pi^2}{12\ln 2} \approx 1.18657\ldots$, i.e. $\gamma \approx 3.27582\ldots$ Roughly speaking, this result means that if values of k are sufficiently large, then the denominator q_k of a continued fraction is "close" to γ^k.

Leaving mathematical rigor aside for a moment, substitute the quantities γ^k and γ^{k+1} into (4) for of q_k and q_{k+1} in order to estimate a mean order of the convergent with a given approximation error Δ. As an upper bound we obtain the number $\frac{\ln(1/\Delta)}{2\ln\gamma} - \frac{1}{2}$, and the lower bound differs from the upper bound by the value $\frac{\ln(1+1/\gamma)}{2\ln\gamma} \approx 0.11$. Thus, the mean value of k (not necessarily integer) is close to $\frac{\ln(1/\Delta)}{2\ln\gamma}$. If $\Delta = 10^{-N}$, then

$$\overline{k} \sim \frac{\ln(1/\Delta)}{2\ln\gamma} = \frac{N\ln 10}{2\ln\gamma} \approx 0,97\cdot N \sim N. \tag{11}$$

This estimation becomes realistic only for large values of N, otherwise $\overline{k}$ is much less than N.

5. Different Rational Arithmetics: A Comparison

To compare different variants of rational arithmetics consider a classical example of a numerical calculation of the function $\sin x$ at points $x_m = \frac{\pi}{6} + 2\pi m$ by summation of its Taylor series. The sum is calculated until the absolute value of a summand becomes less than 10^{-7}. The number π is replaced by its rational approximation $\frac{355}{113}$ with an absolute error $2.7\cdot 10^{-7}$.

A single core *Intel x86* family processor was used for calculations. Different variants of rational arithmetics were created using the arbitrary precision arithmetic, implemented by means of the C^{++} programming language (implementations by means of the *REDUCE* system and *Mathematica* system give similar results).

We consider the following variants of approximate rational arithmetic:

I) The exact rational arithmetic (without rounding).

II) Approximate rational arithmetic described in Section 2 with $M = 9$, $\Delta = 10^{-8}$, $\delta = \infty$ (so only absolute round-off error $\Delta = 10^{-8}$ is fixed).

III) The same arithmetic with $M = 9$, $\Delta = \delta = 10^{-8}$.

IV) The same arithmetic with $M = 9$, $\Delta = \infty$, $\delta = 10^{-8}$ (so only relative round-off errors are fixed).

V) Fixed slash arithmetic [11], [12], where the maximum length L of numerator and denominator is fixed by $L = 6$.

VI) The same arithmetic with $L = 9$.

VII) The same arithmetic with $L = 12$.

VIII) Floating-slash arithmetic [11], [12], where the maximum sum S of lengths of numerator and denominator is fixed by $S = 12$.

IX) The same arithmetic with $S = 15$.

X) The same arithmetic with $S = 18$.

XI) A "reductive" arithmetic, where the numbers of correct first digits D of numerator and denominator are fixed by $D = 9$ (the other digits are replaced by zeros).

The results of numerical computations are presented in Table 1. For calculation of the function $\sin(\frac{\pi}{6} + 2\pi m)$ at points $m = 0, 1, 2, 3, 4, 5, 6$ an absolute errors of the result ε (the relative error equals 2ε), time of calculations t in seconds and the sum of the lengths of numerator and denominator s for the result are specified in Table 1. All values of errors are rounded to the first digits to fit them in the format of the table.

It follows from Table 1 that in the case of the exact rational arithmetic the quick increase of the parameter s leads to the explosive growth of calculations time. Curiously enough, the exact rational arithmetic does not always give the most accurate result.

This phenomenon can be partly explained by the inaccuracy of the presentation of the number π, but it is also a consequence of the unusually simple form of the rational fraction $\sin(\frac{\pi}{6} + 2\pi n) = \frac{1}{2}$. The nature of this effect is discussed in [11]. If the relative error of rounding is only fixed (variant IV), the situation similar to the floating point arithmetic (variant XI) is repeated completely: starting at $m = 4$ the errors are larger then half of computed values (see [13], Part 3).

For other types of rational arithmetic the increase of the difficulty of calculations damages the accuracy of the result essentially, in contrast to variants

	m	0	1	2	3	4	5	6
I	ε	$4\cdot10^{-8}$	$4\cdot10^{-7}$	$8\cdot10^{-7}$	10^{-6}	$2\cdot10^{-6}$	$3\cdot10^{-6}$	$5\cdot10^{-6}$
	s	62	214	372	504	650	810	980
	t	0.007	0.09	0.24	0.95	3.3	5.7	17
II	ε	$2\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	10^{-6}	$2\cdot10^{-6}$	$2\cdot10^{-6}$	$3\cdot10^{-6}$
$\Delta=10^{-8}$	s	16	13	12	12	12	12	11
	t	0.007	0.08	0.15	0.21	0.28	0.34	0.42
III	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	10^{-6}	$2\cdot10^{-6}$	$2\cdot10^{-6}$	$3\cdot10^{-6}$
$\Delta=10^{-8}$	s	15	13	12	12	12	12	11
$\delta=10^{-8}$	t	0.012	0.09	0.16	0.23	0.32	0.37	0.46
IV	ε	$4\cdot10^{-8}$	$3\cdot10^{-7}$	$3\cdot10^{-4}$	0.21	0.6	0.8	1.17
$\delta=10^{-8}$	s	15	13	9	9	9	8	8
	t	0.014	0.06	0.14	0.18	0.25	0.29	0.34
V	ε	0	0	10^{-3}	0.7	1.0	1.4	3.4
L=6	s	2	2	12	12	12	11	12
	t	0.017	0.12	0.18	0.19	0.21	0.23	0.28
VI	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	0.008	0.08	0.3	0.6
L=9	s	17	18	17	18	18	16	18
	t	0.025	0.21	0.29	0.31	0.33	0.36	0.39
VII	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	10^{-6}	10^{-6}	$2\cdot10^{-4}$	0.007
L=12	s	24	24	23	23	24	23	24
	t	0.05	0.29	0.49	0.56	0.64	0.65	0.67
VIII	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	$2\cdot10^{-6}$	10^{-4}	0.04	0.06	0.8
S=12	s	11	11	10	11	11	10	11
	t	0.013	0.18	0.34	0.48	0.51	0.52	0.54
IX	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	$6\cdot10^{-6}$	$2\cdot10^{-3}$	0.01	0.4
S=15	s	14	14	13	13	14	13	13
	t	0.05	0.21	0.37	0.41	0.56	0.61	0.64
X	ε	$4\cdot10^{-8}$	$5\cdot10^{-7}$	10^{-6}	$2\cdot10^{-6}$	$3\cdot10^{-6}$	$3\cdot10^{-5}$	0.01
S=18	s	17	17	15	17	16	17	17
	t	0.023	0.31	0.51	0.68	0.75	0.85	0.87
XI	ε	0	$4\cdot10^{-5}$	0.04	0.1	0.2	0.4	0.9
D=9	s	18	18	18	18	18	18	18
	t	0.008	0.04	0.12	0.17	0.26	0.33	0.42

Table 1: Comparison of rational arithmetics

II and III of approximate rational arithmetic. On the other hand, the time of calculations is comparable in all cases except of the case I (the exact rational arithmetic).

Of course, for practical calculations of Taylor series (this is not our aim in this paper) it is better to compute the sum from small final terms to initial terms. This way leads to similar results in the spirit of Table 1.

Table 2 presents the dependence of the mean number of iterations $\bar{k}$ (see above, Section 4) on the absolute rounding error $\Delta = 10^{-N}$ for $N = 16, 18, \ldots, 36$.

N	16	18	20	22	24	26	28	30	32	34	36
$\bar{k}$	13.9	16.4	18.9	20.9	22.7	24.7	26.5	28.4	30.9	33.0	34.9

Table 2: Dependence of the mean number of iterations on the accuracy of rounding

The calculations were implemented for the model example described above with $M = 9$. It follows from Table 2 that an estimation (11) is realistic for $\bar{k}$ if N is sufficiently large.

6. Conclusion

The approximate rational arithmetic described in Section 2 provides a sufficiently higher degree of accuracy of calculations in a time comparable to other types of rational arithmetic. This result is illustrated by the above calculations quite clearly. It is particularly important that the round-off errors can be controlled by the user on each step of the calculation procedure. This allows us to control the inaccuracy of rounding, estimate the maximum computing error beforehand, and guarantee (in particular, in terms of interval analysis) the required accuracy of calculations. The approximate rational arithmetic of arbitrary precision is useful and convenient for implementations of universal algorithms.

Acknowledgements

The paper was supported by the Russian Foundation for Basic Research, grants RFBR 08-01-00601-a and RFBR-CNRS 05-01-02807.

References

[1] R. Backhouse, P. Jansson, J. Jeuring, L. Meertens, Generic programming - an introduction, *Lect. Notes Comput. Sci.*, **1608** (1999), 28-115.

[2] J. Howell, R.T. Gregory, Solving linear equations using residue arithmetic. II, *Nordisk Tidskr. Inf.*, **10** (1970), 23-37.

[3] A.Ya. Khinchin, *Continued Fractions*, Moscow (1961), In Russian.

[4] A.P. Kryukov, G.L. Litvinov, A.Ya. Rodionov, Construction of rational approximations by means of *REDUCE*, In: *Proc. of the (1986), Symposium*

on Symbolic and Algebraic Computation. Symsac'86 (July 21-23 (1986), Waterloo, Ontario), Univ. of Waterloo (1986), 31-33.

[5] P. Levy, *Théorie de l'Addition des Variables Aléatoires*, Paris (1937).

[6] G.L. Litvinov, V.P. Maslov, The correspondence principle for idempotent calculus and some computer applications, In: *Idempotency* (Ed. J. Gunawardena), Cambridge, I. Newton Institute, Cambridge Univ. Press (1998), 420-443; e-print math. GM/0101021, http://arXiv.org.

[7] G.L. Litvinov, V.P. Maslov, Ed-s, *Idempotent Mathematics and Mathematical Physics*, Amer. Math. Soc., *Contemporary Mathematics*, **377** (2005).

[8] G.L. Litvinov, E.V. Maslova, Universal numerical algorithms and their software implementation, *Programming and Computer Software*, **26**, No. 5 (2000), 53-62; e-print math. SC/0102114, http://arXiv.org.

[9] Yu. Matijasevich, Real numbers and computers, In: *Kibernetika i Vychislitelnaya Technika*, **2** (1986), 104-133, In Russian.

[10] Yu. Matijasevich, A posteriori version of interval analysis, In: *Topics in the Theoretical Basis and Applications of Computer Sciences. Proc. Fourth Hung. Computer Sci. Conf.* (Ed-s: M. Arató, I. Kátai, L. Varga), Budapest, Acad. Kiado (1986), 339-349.

[11] D.W. Matula, P. Kornerup, Approximate rational arithmetic systems: analysis of recovery of simple fractions during expression evaluation, *Lecture Notes in Computer Science*, **72** (1979), 383-397.

[12] D.W. Matula, P. Kornerup, Finite precision rational arithmetic: slash number systems, *IEEE Transaction on Computers*, **C-34**, No. 1 (1985), 3-18.

[13] D.D. McKracken, W.S. Dorn, *Numerical Methods and FORTRAN Programming with Applications in Engineering and Science*, Wiley, Int. Edition, New York (1977).

[14] I. Pohl, *Object-Oriented Programming Using C++*. Reading: Addison-Wesley (1997).