ISSN 2278-3091
Volume 2, No.4, July - August 2013
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://warse.org/pdfs/2013/ijatcse03242013.pdf

Postbinary calculations as a machine-assisted realization of real interval calculations

A. Anopriyenko[1], S. Ivanitsa[2], A. Hamzah[3]
[1] Donetsk National Technical University, Ukraine, anoprien@cs.dgtu.donetsk.ua
[2] Donetsk National Technical University, Ukraine, isv@cs.dgtu.donetsk.ua
[3] Donetsk National Technical University, Ukraine

ABSTRACT

The following article deals with confidence, reliability and precision of computer calculations, and the ways of achievement thereof. Postbinary calculations are regarded as perspective ones since they are capable of comprising the realization of the interval analysis methods. Here are the examples of postbinary code (tetracode) decoding to the borders of real interval. The modified floating-point formats of numbers used for storage and processing of tetracodes are suggested, they enable providing the reliability of computer calculations.

Keywords: Postbinary calculations, tetralogic, tetracodes, floating-point formats, interval analysis, postbinary computing.

1. INTRODUCTION

Confidence, reliability and precision of computer-aided calculations are of a great importance for calculations connected with the engineering of high-end technology machines and mechanisms. While analyzing the whole range of the emergencies and disasters [1] happened recently, it was discovered, that the calculating methods, used while engineering and realized with the help of modern and cutting-edge software systems, have fundamental defects. The main defects is the wrong results of computer calculations without the notice for user about their ambiguity.

The mathematical modeling of industrial goods and projects cannot be carried out without computer calculations.

Four main faults sources can be singled out in the course of realization and receiving of results [2]:

1) faults connected with the inaccurate (or incorrect) basic data input;

2) faults connected with the incorrect choice of numerical methods and algorithms;

3) faults connected with coding and processing of numerical information with the limitation of a digit grid of the computer.

When receiving results of mathematical modeling it is necessary to differentiate three types of computer calculations errors:

1. The reliability of the received results of modeling which corresponds to qualitatively correct results of mathematical modeling of test practical and mathematical tasks.

2. Mathematical accuracy of results of practical tasks modeling which quantitatively estimates the error of mathematical modeling (usually mathematical accuracy is specified percentagewise in relation to known, knowingly exact solution of these tasks).

3. Computer precision of the results of test mathematical tasks solution which quantitatively estimates the error of mathematical modeling and is specified by the number of right significant decimal Figures in the solution of test tasks.

The present article casts light upon the questions of computer precision of the received results while solution of computing tasks by means of computer modeling. Today the set of the decisions pointed at the providing of reliable results during realization of computer calculations over numbers of the various precision in floating point formats has been created. Thus the most significant and perspective directions are:

1) real interval calculations [3; 4];
2) postbinary calculations [5; 6];
3) equivalent transformations which affect the correctness of the task under solution [1; 7].

However the use of intervals implies the appearance of uncertainty and ambiguity at the very beginning, being at the same time an integral part of a problem definition. As a result, interval methods can act only as supportive means for the solution of non-interval tasks.

The application of equivalent transformations leads to the analysis and considerable preliminary work on modification or the radical change of algorithms of the whole computing process. A number of difficulties connected with the reprogramming of already existing and well established computing algorithms is found during the solution of tasks on computer subject to equivalent transformations.

When doing a postbinary calculation the modified floating-point formats of numbers are used. At the same time input parameters can be presented in post-binary formats in such a way that the round-off error of material number is taken into consideration while coding. The post-binary number received as a result of calculation can be reduced to an interval, representing a numerical interval, as the main data object. Thus, the transition to postbinary calculations is optimal when doing the solution of tasks by means of computer modeling with receiving reliable results.

2. POSTBINARY CODING

The quaternary coding system of quantitative information (tetracode) can be determined by the various combinations from four categories (tetryts) which code the conditions of two-dimensional logical space in various combinations [5, page 28]. However the tetracode $C_4 = \{0, \text{A}, \text{M}, 1\}$ attained wider application in a number of researches [8, 9]. Therefore the tetracode will be further understood as a way of data presentation in one digit in the form of a four-sign combination specified by the Figures 0, 1 and letters ‘A’, ‘M’. The tetracode is a positional code, that’s why the number of combinations (codes) of a *k*-digital tetracode is equal to number of arrangements with repetitions:

$$\tilde{A}_4^k = 4^k, \quad (1)$$

where *k* — the number of tetracode digits, i.e. the number of tetrits.

For example, it is possible to code 16 different combinations using 2 tetrits: 00 0A 0M 01 A0 AA AM A1 M0 MA MM M1 10 1A 1M 11; 3 digits — 64: 000 00A 00M 001 … 110 11A 11M 111, etc.

Thus, **postbinary calculations** are calculations which use tetracodes comprising numerical data, and **postbinary coding** is tetracoding, i.e. coding of quantitative information with the help of tetracodes.

In interval calculations the transition to tetracodes is conditioned, first of all, by the possibility of coding of the set of digits in one data field. Moreover, such coding system of quantitative information, possesses a set of qualitative advantages in comparison with traditional systems.

In [10] the process of conversion from a decimal interval into a tetracode is considered by the example of calculation of irrational number by means of Bailey-Borwein-Plouffe’s formula [11]:

$$\pi = f(n) = \sum_{k=0}^{n} \frac{1}{16^k} \cdot \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right). \quad (2)$$

Therewith an initial decimal interval x = [x1, x2] ($x \in \mathbb{IR}$, $x_1, x_2 \in \mathbb{R}$) is composed of 8-digit numbers received at the first and 100-th steps of calculations of the form (2):

$$x_1 = f(1) = 3.1414225; \quad x_2 = f(100) = 3.1415927.$$

Since the received values differ, starting from one ten-thousandth, the accuracy of the four first Figures of the result is guaranteed, when using this interval in the course of the calculations, which make it possible to carry out the automatic accounting of all the types of errors, i.e. in this case numbers.

Binary mappings of the numbers o x_1 and x_2, presented for clearness by 32-digit binary fractions without rounding off, enable to express an interval $\boldsymbol{x}$ = [3.1414225, 3.1415927] by one tetracode *T*:

$x_1 = 3{,}1414225_{10}$ = 11.00100100001101000100001111010b,

$x_2 = 3{,}1415927_{10}$ = 11.00100100001111110110101101001b,

T = 11.001001000011MMMMMAAAAAAAAAAAAA.

The position of the first (senior position) digit M is defined by the first discrepancy of bit values in the corresponding categories (these categories are underlined in the example). The number of digits of “many-valuedness” of ‘M’ is conditioned by the provision of the necessary density of “counting” within the interval borders, that provides approach of at least one of a set of calculated values to initial number with a set precision.

The use of the middle (center) of an interval mid $\boldsymbol{x} = 2^{-1} \cdot (x_1 + x_2)$ — the value, equidistant from boundary values of the interval, — is necessary and sufficient condition for provision of the necessary density of “counting” within the borders of the interval $\boldsymbol{x}$. The process of formation of the tetracode *T* using the binary x_1, x_2 and mid $\boldsymbol{x}$ values is illustrated in the Fig. 1.

The position of the high ‘M’ is defined by the first discrepancy of values of the same weight. The number of values of ‘M’ therewith is determined by the number of value-incoincident digits. The following coincidence of digits values of the same weight means the beginning of group of non-significant digits which are expressed by the digit ‘A’ in a tetracode.

The decoding of the tetracode *T* to the resultant interval $\boldsymbol{t} = [t_{\min}, t_{\max}]$ is achieved by digit-by-digit functions of absolute minimization and maximizing of ‘M’ for receiving the left and the right borders of an interval respectively [5]. From the position of ambiguity of ‘A’ for the borders of the interval $\boldsymbol{t}$ such values are considered, by which the interval will have the minimum width. It is obvious that the case when all the categories of ‘A’ will accept value ‘1’ will ensure the minimum width of an interval, and for the upper bound — when all categories of ‘A’ will accept value ‘0’ (Fig. 1).

Thus, the binary values of borders of a resultant interval $\boldsymbol{t}$:

$t_{\min}$ = MIN_M(MAX_A(*T*)) =
= 11.00100100001100000111111111111b;
$t_{\max}$ = MAX_M(MIN_A(*T*)) =
= 11.00100100001111110000000000000b.

The operation of minimizing and maximizing functions in such a combination will be considered in detail in the monograph [5, page 172].

The received decimal values:

$$t_{min} = 3{,}1411365, \quad t_{max} = 3{,}1415939$$

make it possible to claim that the tetracode T codes the borders of the interval $\boldsymbol{t}$, which in its turn is sure to possess all the values of the initial interval $\boldsymbol{x}$. Actually, the received results correspond to the inequalities $t_{min} < x_1$ and $t_{max} > x_2$. Therefore, values tmin and tmax reduced from T fix all the whole set of values of function $f(n)$ of the expression (2). It is easy to be convinced of it, as the true value is $\pi = 3{,}1415926536\ldots$

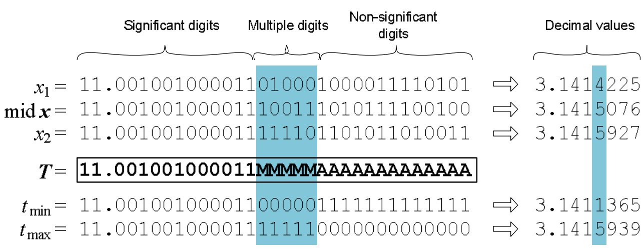

Figure 1: Coding of the interval x with the help of tetracode T with the subsequent obtaining of borders tmin, tmax of the resultant interval

On the ground of the received results it is possible to claim that while coding an initial decimal interval $\boldsymbol{x}$ by tetracode T and the subsequent reduction of this tetracode to a resultant decimal interval $\boldsymbol{t}$, the following proportion is correct

$$\boldsymbol{t} \supseteq \boldsymbol{x}. \qquad (3)$$

The condition (3) underlies post-binary coding of decimal numbers.

The example of formation of the tetracode T from the initial interval $\boldsymbol{y} = [222{,}123;\ 222{,}124]$ with the subsequent getting of resultant intervals $\boldsymbol{t}$ and $\boldsymbol{t}'$ is illustrated in Fig. 2. And $\boldsymbol{t}'$ is got in accordance with the tetracoding principles [5, pages 155–173]. Therefore the dependence received from inequalities shown in Fig. 2 does

$$\boldsymbol{t}' \supset \boldsymbol{t} \supset \boldsymbol{y}. \qquad (4)$$

not conflict with the condition (3).

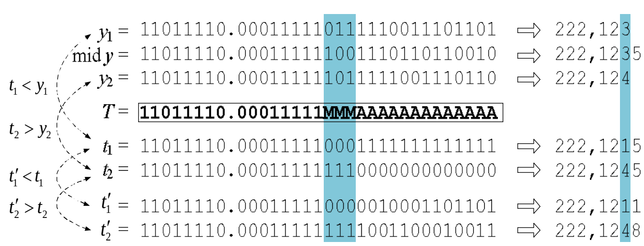

Figure 2: Coding of the initial $\boldsymbol{y}$ by the tetracode T with the subsequent decoding to the intervals $\boldsymbol{t}$ and $\boldsymbol{t}'$

The representation of intervals borders by a tetracode gives the chance of a flexible task practically of all sets of the numerical values included in this interval. In spite of the fact that the number of the bits demanded for "understanding" of tetracodes by modern computers, is doubled in comparison with a binary code, the increase of self-descriptive degree of tetracodes, received for the account of this, is worth coding costs.

3. FLOATING-POINT FORMATS MODIFICATIONS

In the sphere of logical and calculating components of the modern computing the problem of floating-point formats of numbers modification is especially topical. Modification should make it possible to avoid calculating errors considered earlier. The most perspective option of modification is the expansion of the existing standard floating-point formats and operating algorithms by the way of implementation of the following changes:

1. Apart from the standard fields of a mantissa and the exponent, the field of the , for which the parts of the lower order of a mantissa may be involved in standard formats (Fig. 3) and which generally consists of the **cf (format code)** and the **mf (format modifier)**, is introduced for ensuring the flexible use of various floating-point formats of numbers [5, page 198].

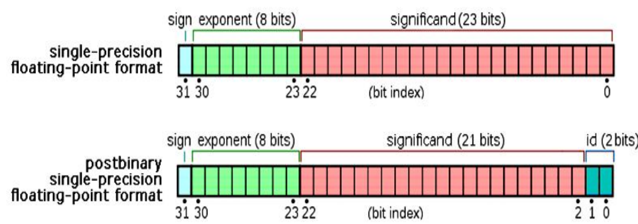

Figure 3: Correlation of the standard (upwardly) and the modified (downwardly) floating-point formats

2. The identifiers of a format specify the digit capacity and the form of representation of floating-point numbers. Therewith up to the 128 digit capacity (binary 128) the **compatibility with the existing formats** and, starting from the digit capacity binary 256, the **successive doubling of digit capacity**, as required with the observance of rules of increasing the digit capacity of the modifier and a format code, are provided.

3. Except the traditional representation of numerical values, **"pair formats"** are introduced additionally. In "pair formats" by the way of doubling of each digit in the form of a single **"postbinary value"** the pairs of traditional pointwise values are presented, **which are interpreted either as a numerator and a denominator of an ordinary fraction or as boundary values of intervals** (in support of interval calculations).

4. **The tetracode, on which basis the representation of "normalized intervals"** and the peculiar concealment of really significant digits on the account of use of "uncertainty" value for digits which are not set and not received in an explicit form can be realized, is regarded as the main postbinary format.

5. Such changes which will enable automatically increasing (or reducing) of the digit capacity as required are made in execution algorithm of arithmetic operations. It will enable the implementing the **arithmetic's "without roundings"**, and, therefore, without loss of precision.

6. The bases for the change of digit capacity can be both, the increase of demanded digit capacity of a mantissa, for instance, while multiplying and exponentiations, and increase of demanded digit capacity of the exponent. It will make it possible to avoid peculiar for floating-point calculations underflow and overflow of exponent.

Table 1: Modified floating-point formats of numbers

Format	Precision	Digit capacity the number mantissa (significant + CF + MF)	Digit capacity of exponent	Shifting of exponent	Code CF
pbinary32	Single	24 (21+1+1)	8	+127	0
pbinary64	Double	53 (48+2+2)	11	+1023	01
pbinary128	Quadruple	112 (104+3+5)	15	+16383	011
pbinary256	Octuple	235 (219+4+12)	20	+524287	0111

4. CONCLUSION

The methods and approaches suggested in this work are the beginning of transition into the new numerical era. The transition into postbinary calculations begins with the post-binary computer logic [8] and finishes with the essential expansion of forms and formats of computer representation of numbers; on the basis of this the post-binary computing should be created, including existing today computing as a particular case.

In the context of transition into postbinary computing, the diversity of presentation of logical and numerical values will inevitably increase due to the necessity of developing both, logical and algorithmic bases of computer technologies, and the concept of number itself, including the level of basic formats of information display. The system of modification of data computer formats suggested in the research may be considered as a prototype of the basic elements for the new generation of the computer architecture, corresponding to the initial stage of a post-binary computing.

REFERENCES

1. Petrov Yu. P. **Obespechenie dostovernosti i nadezhnosti komp'yuternyx raschetov**. / Yu. P. Petrov — SPb.: BXV-Peterburg, 2012. — 160 s.: il.
2. Manichev V. B. **Metody chislennogo resheniya obyknovennyx differencial'nyx uravnenij, nelinejnyx i linejnyx algebraicheskix uravnenij s garantirovannoj dostovernost'yu i tochnost'yu** MGTU. — 32 s.: http://pa10.ru/doc/manichev-oap-lection.pdf.
3. Anopriyenko A. Ya., Ivanitsa S. V. **Interval'nye vychisleniya i perspektivy ix razvitiya v kontekste kodo-logicheskoj e'volyucii** // Nauchnye trudy Doneckogo nacional'nogo texnicheskogo universiteta. Seriya «Problemy modelirovaniya i avtomatizacii proektirovaniya dinamicheskix sistem» (MAP-2010). Vypusk 8 (168): Doneck: DonNTU, 2010. S. 150–160.
4. Ivanitsa S. V., Alrababa Xamza, Anopriyenko A. Ya. **Interval'nyj analiz i ego primenenie v ramkax tradicionnyx i postbinarnyx vychislenij** // Informatika i komp'yuternye texnologii / Materialy VII mezhdunarodnoj nauchno-texnicheskoj konferencii studentov, aspirantov i molodyx uchenyx — 22–23 noyabrya 2011 g. T.1. Doneck, DonNTU. — 2011. S. 249–255.
5. Anopriyenko A. Ya., Ivanitsa S. V. **Postbinarnyj komp'yuting i interval'nye vychisleniya v kontekste kodo-logicheskoj e'volyucii**. — Doneck, DonNTU, UNITEX, 2011. — 248 s.
6. Anopriyenko A. Ya., Ivanitsa S. V. **Tetralogika, tetravychisleniya i nookomp'yuting**. — Doneck, DonNTU, UNITEX, 2012. — 308 s.
7. Petrov Yu. P., Petrov L. Yu. **Neozhidannoe v matematike i ego svyaz' s avariyami i katastrofami**. / Yu. P. Petrov, L. Yu. Petrov — Chetvertoe izdanie, rasshirennoe i dopolnennoe. — SPb.: BXV-Peterburg, 2005. — 224 s.
8. Ivanitsa S. V., Anopriyenko A. Ya. **Osobennosti realizacii operacij tetralogiki //** Nauchnye trudy Doneckogo nacional'nogo texnicheskogo universiteta. Seriya: «Informatika, kibernetika i vychislitel'naya texnika» (IKVT-2011). Vypusk 13 (185). — Doneck: DonNTU, 2011. S. 134–140.
9. Anopriyenko A. Ya., Ivanitsa S. V. **Osobennosti realizacii postbinarnyx logicheskix operacij** // «Iskusstvennyj intellekt», № 2, 2011. S. 110–121.
10. Anopriyenko A. Ya., Ivanitsa S. V. **Osobennosti postbinarnogo kodirovaniya na primere interval'nogo predstavleniya rezul'tatov vychislenij po formule Be'jli-Borue'jna-Plaffa** // Nauchnye trudy Doneckogo nacional'nogo texnicheskogo universiteta. Seriya: «Informatika, kibernetika i vychislitel'naya texnika» (IKVT-2010). Vypusk 11 (164). — Doneck: DonNTU, 2010. S. 19–23.
11. Bailey David H., Borwein Peter B., Plouffe Simon **On the Rapid Computation of Various Polylogarithmic Constants** // Mathematics of Computation. — 1997. — B. 218., T. 66. — p. 903–913.