

УДК 004.942

А. Я. Аноприенко (канд. техн. наук., проф.),
С.В. Иваница (асп.)

Донецкий национальный технический университет
anoprien@cs.dgtu.donetsk.ua

ИНТЕРВАЛЬНЫЕ ВЫЧИСЛЕНИЯ И ПЕРСПЕКТИВЫ ИХ РАЗВИТИЯ В КОНТЕКСТЕ КОДО- ЛОГИЧЕСКОЙ ЭВОЛЮЦИИ

Рассмотрены эволюция интервальных методов и современное состояние классической интервальной арифметики. Предложена возможность гиперкодового представления чисел с плавающей точкой для сохранения информации об исходной точности чисел и предоставления достоверной информации о точности полученных результатов.

интервальные вычисления, тетракоды

Мотивация использования интервальных методов

Интервальный анализ как научное направление сформировался относительно недавно, в основном как метод автоматического контроля ошибок округления на ЭВМ, обусловленный тем, что во многих вычислительных задачах возникла потребность не только вычисления приближенных решений, но и гарантированных оценок их близости к точным решениям. Ценность интервальных решений заключается в том, что они в целом позволяют получать наиболее достоверные решения исходных задач, учитывающие возможные диапазоны изменения исходных и вычисляемых значений [1].

Из классической математики известно, что замкнутый числовой промежуток можно представить в виде интервала. Например, интервал между $x_1 \in R$ и $x_2 \in R$ содержит все вещественные числа из множества R между x_1 и x_2 , включая их самих, и обозначается как $[x_1, x_2]$. Соответственно, интервальную неопределенность можно понимать как состояние неполного (частичного) знания о какой-либо величине, когда возможно лишь указание ее принадлежности к данному интервалу. Иными словами, можно обозначить лишь границы возможных значений рассматриваемой величины (либо пределы ее изменения), и ширина получающегося интервала является естественной мерой интервальной неопределенности (неоднозначности). Такая математическая дисциплина, т. е. дисциплина, которая изучает задачи с интервальными неопределенностями и неоднозначностями в данных и методы их решения, называется интервальным анализом, а подход, основанный на

представлении вещественных чисел в ЭВМ в виде интервалов (интервальный тип данных), называется интервальным подходом.

Выполнение арифметических операций над величинами, имеющими интервальную неопределенность, приводит к интервальной неопределенности в ответе, и интервал результата должен содержать все возможные результаты выполнения операции над элементами исходных интервалов. Это значит, что в результате интервальных вычислений получающийся интервал гарантированно содержит множество всевозможных ответов «точечных» задач, данные к которым содержались в исходных интервалах.

Вместе с тем для сложных задач применение интервального анализа часто дает неудовлетворительные результаты из-за чрезмерной длины получаемых интервалов. Чаще всего это происходит из-за того, что «пессимистические» оценки точности оказываются на порядок хуже, чем реально достигаемая точность результатов [2, 3]. Кроме этого, возникает естественное противоречие между относительно большим диапазоном интервальных значений, отражающим низкую точность соответствующих значений, и предельно точным заданием границ интервалов.

В то же время все чаще возникают ситуации, когда возможности классических бинарных ЭВМ (ориентированных на использование бинарной логики и арифметики только с двумя возможными логическими состояниями «0» и «1») не отвечают возрастающей сложности и масштабам современных вычислительных задач и компьютерного моделирования. Одно из возможных решений этой проблемы лежит в переходе к постбинарным вычислениям в контексте кодо-логической эволюции. При этом появляется возможность максимально использовать тот научно-технический задел, который накоплен в рамках интервальных вычислений, но при этом устранить или минимизировать некоторые проблемы интервального анализа, связанные, например, с противоречием между чрезмерно точным представлением границ интервальных значений и фактической неточностью тех данных, которые они представляют.

Эволюция интервальных методов

Первая монография, полностью посвященная интервальному анализу, была опубликована Р. Е. Муром в 1966 г. [4]. В этой монографии практически впервые были последовательно изложены основы нового направления в вычислительной математике, а также высказана точка зрения, что первым «интервальщиком» следует считать Архимеда, широко использовавшего в своих расчетах двусторонние приближения, в частности, для определения границ числа π – отношения длины окружности к ее диаметру. Предложенные Муром новые интересные

постановки задач и поучительные применения интервальной техники оказали решающее влияние на становление и развитие нового научного направления во всем мире.

Началом широкого распространения интервальных методов в компьютерных технологиях можно считать первый международный симпозиум по интервальному анализу, прошедший в Великобритании в январе 1968 года [5]. На русском языке первая достаточно известная работа по интервальным вычислениям была опубликована Ю.И. Шокиным в 1981 году [6].

«Интервальная идея» начала развиваться в XX веке в тесной связи с развитием и распространением практических инженерных вычислений. Но оформление интервального анализа в самостоятельную научную дисциплину стало возможным лишь с широким распространением ЭВМ. Последующие исследования показали, что методы интервального анализа могут служить не только для учета ошибок округления на ЭВМ, но и являются достаточно эффективными аналитическими методами для теоретических исследований [7].

Еще в 1931 году Р. Янг (Великобритания) [8, с.260-290] была предложена арифметика для вычислений с множествами чисел. Американский ученый П. Двайер в 1951 г. рассматривал специальный случай замкнутых интервалов в связи с необходимостью учета погрешностей в численном анализе [9]. В 1956-58 гг. в Польше были опубликованы работы М. Вармуса [10, с. 253-259] и Т. Сунаги в Японии [11], предлагавшие классическую интервальную арифметику и намечавшие ее приложения. При этом в работе Т. Сунаги [11, с. 547-564] впервые были использованы и современные термины «интервал», «интервальный». Кроме того, им были заложены основы интервального алгебраического формализма и даны весьма нетривиальные примеры применений новых методов, к примеру, в численном решении алгебраических уравнений и задачи Коши для обыкновенных дифференциальных уравнений (см. также работу [12, с. 167-188]). В 1959-м году начал публиковать свои работы в области интервальных вычислений также и Р. Э. Мур, на базе чего к 1966 году была издана упомянутая выше монография [5].

В России и Советском Союзе «интервальную» историю можно отсчитывать с 20-х годов прошлого века, и связана она с именем видного русского советского математика Владимира Модестовича Брадиса, который широко известен своими математическими таблицами. В. М. Брадис предложил так называемый метод границ – способ организации вычислений, приводящий к достоверным двусторонним границам точного значения вычисляемого результата, фактически аналогичный

интервальной арифметике. Работая в Тверском Педагогическом институте, он опубликовал целый ряд работ на эту тему [13-15].

В 1962-м году в одном из первых выпусков «Сибирского математического журнала» была опубликована статья Леонида Витальевича Канторовича [16, с.701-709], обозначившего эту тематику как одну из приоритетных для активно набирающей обороты вычислительной науки. В 1982 г. было издано учебное пособие Т. Н. Назаренко, Л.В. Марченко по интервальным методам [17], а в 1986 г. – монография С. Л. Калмыкова, Ю. И. Шокина, З. Х. Юлдашева [18]. Обширная и подробная библиография по интервальному анализу и вычислениям имеется, в частности, в работах [19-22].

К настоящему времени разработаны различные приемы интервальных вычислений [19, 21-24] и множество пакетов прикладных программ и алгоритмических макроязыков, реализующих элементы интервального анализа на машинном уровне для нескольких типов ЭВМ [23, 24]. В настоящее время готовится также соответствующий стандарт для интервальных вычислений [25, 26].

Современное состояние классической интервальной арифметики

Одной из причин использования интервальных методов является то, что современные классические ЭВМ не учитывают, степень неточности большинства исходных данных. Даже невинно выглядящее дробное число $1/10$ может порождать в определенных случаях вычислительную проблему, т. к. компьютер не может выполнять точные вычисления с этим числом [27]. В той мере, в какой точные вычислительные результаты используются для принятия критических решений, неучтенные ошибки вычислений означают повышенный риск. Очевидно, что чем сильнее зависимость точности входных данных от точности вычисляемых значений, чем более важной для последних является их корректность, и тем больше допускаемый риск. Например, широко известен такой классический пример, как катастрофа с американской зенитной ракетой Patriot 25 февраля 1991 года в Дхаране (Саудовская Аравия) [28]. Он показывает, что может произойти, если существующие вычисления с плавающей точкой будут и далее некритично применяться к новым задачам. В тот день батарея ракет Patriot не смогла перехватить иракскую ракету Scud, по официально названной причине: неадекватное вычисление в формате с плавающей точкой. Дело в том, что система управления ракеты Patriot имеет внутренние системные часы, отсчитывающие время в десятых долях секунды, т. е. для перевода времени в формат с целыми секундами компьютер просто умножает данные на $1/10$. Однако, как уже

упоминалось выше, на современных классических ЭВМ дробь $1/10$ не имеет точного внутреннего представления, и должна быть приближена подходящей двоичной дробью. В качестве такого приближения американские разработчики взяли 24-битное двоичное число 0.00011001100110011001100, которое меньше, чем $1/10$, примерно на одну миллионную. Эта на первый взгляд ничтожно малая погрешность постепенно накапливалась и после четырех дней непрерывной работы расхождение системного времени с точным достигло $1/3$ секунды, что, в конечном счете, привело к ошибке наведения в 700 метров. В результате ракета, выпущенная на перехват Skud, попала в помещение с американскими военнослужащими, убив 28 человек [28].

Вышеприведенный и подобные ему (не столь катастрофичные) случаи, наглядно демонстрируют, что являющиеся основой современных цифровых ЭВМ числа в формате с плавающей точкой, оказываются не вполне адекватными как реальному физическому миру, так и его математическим моделям, в частности, математическому понятию вещественного (действительного) числа. Основные недостатки современного представления чисел с плавающей точкой заключаются в следующем:

- большинство чисел вещественной оси не могут быть представлены точно числами с плавающей точкой, имеющими конечную длину мантиссы, и, соответственно, свойства арифметических операций над числами с плавающей точкой отличаются (из-за неизбежных округлений) от свойств идеальных математических операций над вещественными числами;

- число в формате с плавающей точкой не несет никакой информации о точности той величины, которую оно представляет.

Получается, что существующая модель вычислений с плавающей точкой не предназначена ни для адекватного представления исходных значений, ни для отслеживания вычислительных ошибок. В связи с этим постепенно усиливается тенденция к переходу от точечных значений к интервальным, что влечет за собой стремительное развитие интервальной арифметики.

Интервальный тип данных и интервальная арифметика реализуются на современных ЭВМ, как правило, с помощью представления интервала в виде пары чисел – одного для левого конца интервала, а другого для правого. При этом существующее аппаратное обеспечение, в частности, арифметика чисел с плавающей точкой, используются без каких-либо изменений, так как корректность получающейся интервальной арифметики может быть обеспечена так называемыми направленными округлениями. Например, там, где в задачах внешнего интервального оценивания в процессе вычислений требуется округление результата, нижняя граница

интервала должна округляться вниз, а верхняя граница интервала – вверх. Таким образом, даже неизбежные ошибки округления при вычислениях с плавающей точкой будут строго и систематически учитываются в процессе выполнения интервальной программы. В качестве примера, на рис. 1 показано, как иррациональные числа $\sqrt{2}$ и π в различных числовых шкалах представлены в виде различных интервалов (числовых промежутков), причем наглядно проиллюстрировано изменение «ширины» этих интервалов. Интервалы чисел, представленных в вещественном формате (шкала floats) являются достаточными, так как в границы интервалов включены и ошибки округления исходных иррациональных чисел: $\sqrt{2} \in [1.25, 1.5]$; $\pi \in [3.0, 3.5]$.

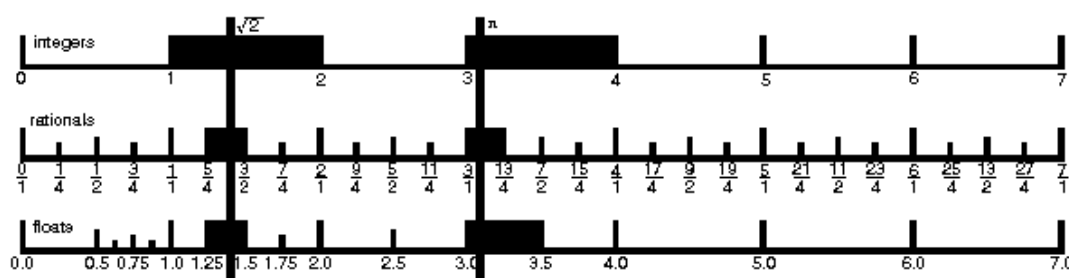


Рисунок 1 – Представление иррациональных чисел в виде интервалов (integers – целые числа, rationals – рациональные числа, floats – вещественные числа) [27, с. 485]

Данный пример отражает основные положения интервального подхода к вычислениям на ЭВМ: исходные данные и промежуточные результаты представляются граничными значениями, над которыми и производятся все операции. При этом сами операции (прежде всего арифметические) определяются таким образом, что результат соответствующей точной операции обязательно лежит внутри вычисляемых границ.

Иллюстрацией интервального подхода могут служить следующие правила выполнения операций вещественной интервальной арифметики:

$$[x_1, x_2] + [x_3, x_4] = [x_1 + x_3, x_2 + x_4];$$

$$[x_1, x_2] - [x_3, x_4] = [x_1 - x_4, x_2 - x_3];$$

$$[x_1, x_2] \times [x_3, x_4] = [\min(x_1x_3, x_1x_4, x_2x_3, x_2x_4),$$

$$\max(x_1x_3, x_1x_4, x_2x_3, x_2x_4)];$$

$$[x_1, x_2] / [x_3, x_4] = \left[\min\left(\frac{x_1}{x_3}, \frac{x_1}{x_4}, \frac{x_2}{x_3}, \frac{x_2}{x_4}\right), \max\left(\frac{x_1}{x_3}, \frac{x_1}{x_4}, \frac{x_2}{x_3}, \frac{x_2}{x_4}\right) \right],$$

где $0 \notin [x_3, x_4]$.

Возникающая при вычислении границ погрешность учитывается с помощью направленных округлений: меньшая из вычисленных границ получается округлением до ближайшего машинного числа с недостатком, а большая – с избытком. Таким образом, интервальный подход позволяет единообразным способом учесть все виды погрешностей вычислительного процесса: приближенно известные исходные данные заключаются в гарантированно содержащие точное значение границы. Погрешности округлений лишь несколько расширяют границы промежуточных результатов, а сам вычислительный метод строится так, чтобы его погрешность также включалась в вычисленные границы конечного результата [29].

Постбинарные методы реализации интервальных вычислений в компьютерном моделировании

Традиционно аппаратное обеспечение компьютеров поддерживает две числовые системы: целые числа и числа с плавающей точкой. Целочисленная арифметика оперирует конечным подмножеством множества целых чисел и позволяет безошибочно осуществлять адресные вычисления, компиляцию и другие формы трансляции, а также реализовать различные алгоритмы типа поиска и сортировки [30].

Произвольное вещественное число представляется бесконечной систематической (например, десятичной или двоичной) дробью. На практике в научных и инженерных вычислениях вещественные числа приходится представлять в компьютере конечными дробями, чаще всего числами с плавающей точкой. Арифметика чисел с плавающей точкой поддерживается аппаратным обеспечением компьютеров и поэтому выполняется очень быстро, однако каждая операция с вещественным числом может вносить погрешности, накопление которых может существенно исказить результат [31].

Современные ЭВМ практически полностью базируются на двоичной логике и арифметике, обеспечивающих до недавнего времени практически все потребности компьютерных вычислений. Однако в 90-е годы прошлого века произошли качественные изменения, как в развитии логических основ, так и в области компьютерных технологий, которые обусловили актуальность соответствующих изменений как в кодо-логическом [32], так и в алгоритмическом [33] базисе современных компьютерных технологий. Суть данных изменений может быть сведена к переходу от преобладания фиксированной точечной определенности к эволюционирующей множественности и неопределенности, а также к рассмотрению новых перспектив использования многомерного кодо-логического базиса в вычислительном моделировании и представлении знаний.

Одним из вариантов усовершенствования традиционной двоичной логики является переход к многозначной логике, т. е. двумерное логическое пространство может быть продуктивно расширено до трехмерного путем введения третьего измерения, соответствующего возможной недостоверности и/или «вариабельности» (т. е. возможной изменчивости) логических значений двумерного пространства. В качестве простейших частных случаев использования вариабельности могут рассматриваться тетралогика, позволяющая реализовать свойство адаптивности в рамках подходов, характерных для традиционной бинарной логики.

Аналогично тому, как бинарная логика является основой двоичной системы счисления, на базе тетралогии может быть построена соответствующая система кодирования количественной информации. При этом в рассмотрение могут быть введены вариабельные (модифицируемые) тетракоды, функциональные возможности которых существенно превышают возможности классической двоичной (бинарной) системы. Тетралогика включает в себя кроме классических «1» (истина) и «0» (ложь) разные пары комбинаций следующих значений, а именно: «М» как «множественность»; «А» как «возможность», «равновероятность» и некоторые другие [34].

Каждый разряд тетракода может представлять собой двухбитную комбинацию, которая соответствует одному из четырех состояний тетралогии. Таким образом суммарное количество бит для представления числа увеличивается в 2 раза, но при этом осуществляется качественно важное изменение кода: из точечного (нульмерного) числа он превращается в одномерное, что позволяет эффективно использовать все пространство числовой оси. Тетракоды позволяют перенести интервальные вычисления на уровень реализации элементарных операций и в связи с этим представляются наиболее перспективным направлением развития интервальных подходов в современных компьютерных вычислениях, позволяющим на определенном этапе перейти на аппаратную реализацию соответствующих процессоров [35].

Наиболее универсальной формой компьютерного представления тетракодов является формат с плавающей запятой, в котором представленные соответствующим образом мантисса и порядок должны быть дополнены значениями смещений, представленных обычными бинарными кодами, обеспечивающими совместимость с традиционными двоичными операциями машинной арифметики. Другими словами, если обычные натуральные числа представляются в компьютерных системах в формате $m \cdot 2^e$, где m – мантисса, e – порядок, кодируемые традиционно в виде бинарных кодов, то для универсального представления тетракодов целесообразен формат $(d+m') \cdot 2^{(k+e')}$, где m' – мантисса и e' – порядок,

представленные соответственно в виде целочисленных тетракодов, а d и k – смещения, представленные в виде классического бинарного кода. При этом смещения, с одной стороны, позволяют достаточно просто решить проблему совместимости гиперкодового представления с традиционной машинной арифметикой, а, с другой стороны, открывают возможности для дальнейшего усложнения структуры и многофункциональности численного представления путем введения элементов гиперкодового кодирования в представление смещений.

Использование подобного гиперкодового представления позволяет объединить традиционные вычисления с плавающей запятой с возможностью сохранения информации об исходной точности чисел и позволит иметь достоверную информацию о точности полученных результатов.

Заключение

Существующая модель вычислений с плавающей точкой не предусматривает текущего учета и анализа точности процесса вычислений и получаемых результатов, что в ряде случаев может приводить к так называемым «вычислительным катастрофам» [36]. В качестве одного из перспективных направлений решения данной проблемы представляется использование постбинарной логики и арифметики (например, в виде тетралогии и тетракодов). Объединение нынешних средств и методов компьютеринга с системой достоверных (интервальных) вычислений становится все более актуальным в компьютерном моделировании сложных динамических систем и при разработке нового постбинарного поколения компьютерных технологий. Именно поэтому на факультете компьютерных наук и технологий ДонНТУ инициированы исследования в данном направлении.

Список литературы

1. Добронев Б. С. Интервальная математика: Учеб. Пособие. Краснояр. гос. ун-т. – Красноярск. 2004. – 216 с.
2. Марчук Г.И. Методы вычислительной математики. – М.: Наука. 1977. – 456 с.
3. Самарский А.А. Введение в теорию разностных схем. – М.: Наука, 1971. – 552 с.
4. Moore R.E. Interval analysis. Englewood Cliffs / R.E. Moore – N.J.:Prentice-Hall, 1966.
5. Hansen E. Topics in Interval Analysis. – London: Oxford University Press, 1969.
6. Шокин Ю. И. Интервальный анализ. / Ю. И. Шокин – Новосибирск: Наука, 1981. – 111 с.

7. Интервальный анализ и его приложения. Исторические заметки, <http://www.sbras.ru/interval/index.php?j=Introduction/history>.
8. Young R.C. Algebra of many-valued quantities // *Mathematische Annalen* / R. C. Young, 1931. S. 260-290.
9. Dwyer P.S. *Linear Computations* / P. S. Dwyer – New York: John Wiley & Sons, 1951. – 36 p.
10. Warmus M. Calculus of approximations // *Bull. Acad. Polon. Sci./ M. Warmus* – 1956, Cl. III, vol. IV, No. 5.
11. Sunaga T. Theory of an interval algebra and its application to numerical analysis // *RAAG Memoirs*. – Vol. 2, Misc. II, 1958.
12. Markov S., Okumura K. The contribution of T.Sunaga to interval analysis and reliable computing // *Developments in Reliable Computing* / Cendes T., ed. – Dordrecht: Kluwer Academic Publishers, 1998.
13. Брадис В.М. Опыт обоснования некоторых практических правил действий над приближенными числами // *Известия Тверского педагогического института*. 1927. – Вып. 3.
14. Брадис В.М. Теория и практика вычислений. Пособие для высших педагогических учебных заведений. – Москва: Учпедгиз, 1937.
15. Брадис В.М. Средства и способы элементарных вычислений. – Москва: Издательство Академии педагогических наук РСФСР, 1948.
16. Канторович Л.В. О некоторых новых подходах к вычислительным методам и обработке наблюдений // *Сибирский Математический Журнал*. – 1962. – Т. 3, No. 5.
17. Назаренко Т.И., Марченко Л.В. Введение в интервальные методы вычислительной математики. / Т.И. Назаренко, и др. – Иркутск: Изд-во Иркут. ун-та. 1982.
18. Калмыков С.Л., Шокин Ю.И., Юлдашев З.Х. Методы интервального анализа. / О.Л. Калмыкин и др. – Новосибирск: Наука. 1986. – 224 с.
19. Алефельд Г., Херцбергер Ю. Введение и интервальные вычисления. / Г. Алефельд и др. – М.: Мир, 1987. – 360 с.
20. Шарый С.П. Конечномерный интервальный анализ. – Новосибирск, Институт вычислительных технологий СО РАН, 2009. – 569 с.
21. Beierbaum, F., Schwierz. K.P. A bibliography on interval mathematics // *J. Comput. Appl. Math.* V. 4, N 1. P.59-86.
22. Moore R.E. *Methods and Applications of Interval Analysis*. SIAM. Philadelphia 1979.
23. Клатте Р., Кулиш У., Неага М., Рац Д., Ульрих Х. PASCAL-XSC Язык численного программирования. / Р. Клатте и др. – М.: ДМК Пресс, 2000.
24. Агеев М. П., Алик И. П., Марков Ю. И. Алгоритм 616. Процедуры интервальной математики // Библиотека алгоритмов 516-11/06. / М. П. Агеев и др. – М.: Сов. Радио, 1976.
25. Interval arithmetic - From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Interval_arithmetic
26. IEEE Interval Standard Working Group - P1788. <http://grouper.ieee.org/groups/1788/>
27. Hayes B. A Lucid Interval. A reprint from *American Scientist* the magazine of Sigma Xi, the Scientific Research Society, Volume 91, Number 6, November–December, 2003, p.484-488.
28. Строгий учет ошибок округлений на цифровых ЭВМ, <http://www.sbras.ru/interval/index.php?j=Introduction/RusIntro>.

29. Кулиш У., Рац Д., Хаммер Р., Хокс М. Достоверные вычисления. Базовые численные методы М.: «Регулярная и хаотическая динамика», 2005.

30. Кнут Д. Е. Искусство программирования, том 3. Сортировка и поиск, 3-е издание / Д. Е. Кнут – Спб.: Диалектика, 2005.

31. Аноприенко А.Я. Тетралогия и тетракоды. // Сборник трудов факультета вычислительной техники и информатики. Вып.1. – Донецк: ДонГТУ. – 1996. С. 32-43.

32. Аноприенко А.Я. Расширенный кодо-логический базис компьютерного моделирования / В кн. «Информатика, кибернетика и вычислительная техника» (ИКВТ-97). Сборник научных трудов ДонГТУ. Выпуск 1. Донецк, ДонГТУ, 1997, с. 59-64.

33. Аноприенко А.Я. Эволюция алгоритмического базиса вычислительного моделирования и сложность реального мира // Научные труды Донецкого национального технического университета. Выпуск 52. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2002): Донецк: ДонНТУ, 2002. – С. 6-9.

34. Соболева А.Г. Когнитивная визуализация данных с помощью лиц Чернова // Збірка тез доповідей II Міжнародної наукової конференції студентів, аспірантів та молодих вчених «Комп'ютерний моніторинг та інформаційні технології 2006», 15-17 травня 2006 р. – Донецьк: ДонНТУ, 2006.

35. Аноприенко А.Я. Обобщенный кодо-логический базис в вычислительном моделировании и представлении знаний: эволюция идеи и перспективы развития // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2005) выпуск 93: – Донецк: ДонНТУ, 2005. С. 289-316.

36. Юровицкий В.М. О компьютерной «вычислительной катастрофе», <http://www.yur.ru/science/computer/Comcat.htm3>

Надійшла до редакції 02.11.2009 р. Рецензент: к.т.н., доц. Зінченко Ю.Є.

О.Я. Аноприенко, С.В. Іваниця

Донецький національний технічний університет

Інтервальні обчислення і перспективи їх розвитку в контексті кодо-логічної еволюції. Розглянуто еволюцію інтервальних методів і сучасний стан класичної інтервальної арифметики. Запропоновано можливість гіперкодового представлення чисел з рухомою крапкою для збереження інформації про вихідну точність чисел і надання достовірної інформації про точність отриманих результатів.

інтервальні обчислення, тетракоди

A. Anoprienko, S. Ivanitsa

Donetsk National Technical University

Interval calculations and perspectives of their development in context code-logical evolutions. Consider evolution of interval methods and modern condition of classical interval arithmetic. Propose possibility of hypercode's conception of floating point numbers for preservations information about entrance accuracy data and granting of trustworthy information about of received results.

interval calculation, tetracodes