

DOI: <https://doi.org/10.37129/2313-7509.2020.14.2.41-50>

УДК 004.4

І.В. Шаріпова¹**С.Г. Трутнев**¹**А.О. Левченко**¹, к.т.н., доц.**О.В. Головка**²<https://orcid.org/0000-0003-0521-1299><https://orcid.org/0000-0002-6107-7920><https://orcid.org/0000-0001-5550-0027>¹Військова академія (м. Одеса), Україна²Одеський національний університет імені І.І. Мечникова, Україна

ВИТОКИ ПОМИЛОК ПРОГНОЗУВАННЯ СИТУАЦІЙ В КОМП'ЮТЕРНИХ СИСТЕМАХ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Сучасні тенденції розвитку теорії і практики наукових досліджень показують, що одними із визначальних елементів визначення достовірності результатів досліджень є комп'ютерне моделювання. З розвитком мов програмування високого рівня та доступністю технічних засобів значна кількість науковців використовують програмні засоби моделювання як інструмент дослідження. За допомогою систем імітаційного моделювання приймаються рішення, що впливають на людські життя, планування бойових дій, напрямки розвитку видів Збройних сил та галузей промисловості і навіть перспективи державотворення. Але фахівці в відповідних галузях ґрунтуючись на підсумках комп'ютерних розрахунків не враховують особливості точності розрахунків у комп'ютері з двійковими числами та вплив двійкової арифметики на результати імітаційного моделювання. У статті авторами використано підхід визначення витоків помилок комп'ютерного моделювання за допомогою фрагментів програмного коду визначеного діапазону десятичних чисел. За допомогою фрагментів програмного коду показано відмінності результатів десятичних та двійкових обчислень з особливостями представлення двійкових чисел з плаваючою комою. Виявлено, що на результат обчислень найбільш впливають помилки денормалізації числа, помилки циклічності операції суми та помилки округлення в залежності від опису точності представлення змінних безпосередньо в програмному коді. Застосування такого підходу надає можливість вже під час реалізації програмних продуктів виключати, чи визначати діапазони числових значень представлення яких в двійковому вигляді може привести до спотворення результатів моделювання.

Ключові слова: представлення чисел з плаваючою комою, числові значення моделювання, математична модель, архітектура комп'ютера, помилки моделювання, помилки округлення, помилки денормалізації.

Числові значення параметрів, які використовуються в комп'ютерних системах імітаційного моделювання відрізняються різнорозмірністю [1]. Поняття різнорозмірності для комп'ютерних обчислень вперше розтлумачено в [2]. Під різнорозмірністю запропоновано мати на увазі різну кількість елементів розрядної сітки представлення числових значень сукупності числових значень в комп'ютері. Слід згадати результати останніх досліджень присвячених проблемі розрахунків з представленням чисел з плаваючою комою [3]. Думка, що у розрахунках з плаваючою комою підсумок не перевищує межі відносної похибки представлення найбільшого числового значення виявляється помилковою. В комп'ютері, з урахуванням сучасної архітектури побудови апаратної частини [4], помилки розрахунків накладаються одна на одну без урахування знаку, а базові арифметичні тотожності не працюють. Наявність одночасно малих та великих значень параметрів в комп'ютерних розрахунках з плаваючою комою, у підсумку реалізацій алгоритмів бінарних розрахунків з обмеженою розрядною сіткою, роблять помилки неприпустимими [1], [5].

Відсутність визначення витоків помилок прогнозування ситуацій в комп'ютерних системах імітаційного моделювання ставить під сумнів саму доцільність застосування згаданих систем з великими ризиками в разі прийняття нераціональних рішень.

Постановка проблеми

Основним форматом подання даних в сучасних комп'ютерах є формат подання двійкових чисел в плаваючою крапкою. Формат подання двійкових чисел в плаваючою крапкою введено стандартом IEEE 754 [3].

В техніці та програмуванні як правило використовують формати 32 і 64 біта. Наприклад в мові програмування: VB типи числових даних представляють як single (32 біта) і double (64 біта), а в Cі, Cі++ – float (32 біта) і double (64 біта). Інші формати представлення чисел є версією single-precision. Для представлення числа форматі single-precision відповідно до IEEE 754 необхідно привести число до двійкового нормалізованого вигляду. Процедура перетворення числа в 32 бітний формат IEEE 754 реалізує наступні кроки:

- 1 біт відводиться для позначення знаку числа. Це старший біт в 32 бітній послідовності;
- трансформуються біти представлення експоненти, для цього виділяють 1 байт (8 біт). Експонента може бути, як і число, со знаком + чи -, однак сутність цих знаків для числа, експоненти та мантиси абсолютно відмінний. Для визначення знаку експоненти, щоб не вводити ще один біт знака, додають зсув до експоненти в половину байта. Зсувній експоненті відводять 8 біт;
- 23 біти відводять для мантиси. Але, у нормалізованій двійковій мантисі перший біт завжди дорівнює 1, так як число знаходиться в діапазоні $1 \leq M < 2$.

Числа у форматі з плаваючою комою представляють кінцеву множину, на якій слід відобразити нескінчену множину чисел. Тому вихідне числове значення в системах імітаційного моделювання вже представляється х помилкою.

Ситуація з розвитком мов прогнозування високого рівня та доступністю комп'ютерних засобів викликає занепокоєність. За допомогою оцінки ситуацій в системах імітаційного моделювання приймаються рішення, що впливають на людські життя [6, 7], або навіть на напрямки розвитку держав [8] та галузей промисловості [9]. Але фахівці в відповідних галузях ґрунтуючись на підсумках комп'ютерних розрахунків не враховують особливості саме точності розрахунків у комп'ютері з двійковими числами.

Аналіз останніх досягнень і публікацій

Аналіз останніх досліджень, публікацій у різних галузях науки та техніки свідчить, що сьогодні питання застосування систем імітаційного моделювання бурхливо використовується як інструмент наукових досліджень. Як приклади різноманітності галузей відповідних робіт наведемо роботи: [6]–[10]. У згаданих авторах матеріалах, як і у більшості інших в різноманітних галузях досліджень, посилання на результати моделювання та розробка математичних моделей здійснюється без дослідження тотожності математичних алгоритмів їх математичним аналогам. Тоді як алгоритми обчислень за допомогою яких отримуються підсумки комп'ютерного моделювання трансформовано відповідно до архітектури комп'ютерних систем [4]. В сучасних роботах перед початком моделювання фахівці в прикладних галузях досліджень не враховують відповідність множини числових значень необхідних для моделювання множині чисел відображених комп'ютером.

Виключенням є замкнуте середовище професійних програмістів де проблема обговорюється таким чином, щоб не вплинути на фінансову стабільність ІТ індустрії. В [12] для прикладу наведено баг №53632 для мови програмування PHP, що обумовив паніку на початку 2011 року. До здійсненого факту відмови всіх програмних додатків на мові PHP у світовій спільноті ніхто навіть не замислювався про вразливість ресурсів, обумовленою особливостями подання конкретного числа в десятинній та двійковій системах обчислення.

Професійне комп'ютерне середовище зреагувало на проблему специфічно – в мережі Internet з'являються платні сервіси з конвертації чисел [13], а галузь комерційних хмарних послуг з високоточних обчислень набула фінансових показників у кілька десятків мільярдів доларів прибутку у рік.

Постановка завдання

Метою системних досліджень авторів є вивчення методів забезпечення точності результатів комп'ютерних обчислень у розрахунках з різнорозмірними даними. Метою статті окреслено аналіз особливостей подання числових значень, що впливають на витoki помилки прогнозування ситуацій в комп'ютерних системах імітаційного моделювання.

Виклад основного матеріалу дослідження

Сучасні мови програмування реалізують типи даних для роботи з великими раціональними й цілими числами. Типом даних, що описує найбільший діапазон значень цілих чисел доступних нам (C++ і Pascal) є тип даних `int64`, який вміщує 8-байтове ціле число і може представляти цілі числа від 0 до $18446744073709551615(2^{64})$, або від -9223372036854775808 до $+9223372036854775807$. Такого типу даних вистачає, якщо мова не йде про малі раціональні числа (наприклад розрахунки похибок в системах технічної діагностики), або вирішення математичних задач з дуже високою точністю (наприклад реалізація чисельних методів в інформаційних технологіях СППР з прогнозуванням) [2].

На сьогодні існує цілий ряд задач, які вимагають підвищеної точності розрахунків, серед яких маємо:

- високо нелінійні обчислення;
- розрахунки за участю лінійних систем з поганою обумовленістю;
- обчислення даних з дуже великим динамічним діапазоном;
- великі обчислення на високопаралельних комп'ютерних системах;
- розрахунки, в яких числова чутливість в даний час не є серйозною проблемою, але потрібно періодичне тестування, щоб гарантувати надійність результатів;
- дослідницькі проблеми математики і математичної фізики, пов'язані з розпізнаванням констант і виявленням цілочисельних відносин.

Саме ці галузі активно «компютерізуються» та дослідження у яких ґрунтуються на результатах імітаційного моделювання. Таким чином, високоточна арифметика набуває критичного значення як засіб від серйозної помилки чисельного округлення. Характер математичних задач та алгоритмів, що закладено у предметну область, визначає необхідність впровадження методів підвищення номінальної точності розрахунків. Одним з таких прикладів є модуляція клімату та погоди, що є принципово хаотичною – якщо мікроскопічні зміни вносяться до поточного стану, незабаром майбутній стан стане зовсім іншим. На практиці обчислювальні результати змінюються, навіть якщо в код або систему внесені незначні зміни. Ця чисельна варіація є основною неприємністю для обслуговування системи. У [14] виявили, що завдяки використанню арифметичних операцій зі словами подвоєною довжини для реалізації ключового внутрішнього циклу добутку більшість цих числових змін зникли.

Як інший приклад моделювання, ключове питання планетарної теорії полягає в тому, чи стабільна Сонячна система протягом космологічних часових рамок (мільярди років). Вчені вивчали це питання, виконуючи дуже тривале моделювання рухів планет. Ця проблема, як відомо, демонструє хаос. Моделювання, як правило, добре спрацьовує протягом тривалого періоду часу, але потім виходить з ладу на певних ключових місцях, якщо не вжити спеціальних заходів. [15] виявили, що арифметичні операції зі словами подвоєною довжини необхідні, щоб уникнути серйозних числових неточностей, навіть якщо застосовуються інші методи.

Дослідники з LBNL [16] використовують квантову точку для визначення нелокальних термодинамічних рівноважних популяцій заліза та інших атомів в атмосферах наднових зірок. Залізо може існувати в кількох видах, тому необхідно знайти рішення для всіх видів одночасно. Оскільки відносна щільність будь-якого стану з домінуючого стану пропорційна експоненті енергії іонізації, динамічний діапазон цих значень може бути дуже великим. Частина арифметичних операцій зі словами подвоєною довжини тепер домінує в усіх обчисленнях. Прикладом критичного значення арифметики підвищеної точності у ітеративному методі та функціональному аналізі є обчислення електромагнітного розсіювання, де ключовою операцією є пошук точок відгалуження асимптотичного розширення сфероїдальної хвильової функції. Схеми, засновані на ітераціях Ньютона-Рафсона, що використовують стандартну точність машини, мають обмеження точності. Тому зараз впроваджуються спеціальні пакетні рішення для значного розширення діапазону хвильових функцій, які можна вивчити за допомогою цих моделювань.

Представлення в пам'яті ЕОМ раціональних типів даних обумовлює обмеження на точність їх представлення, що є критичним питанням відносно вищезазначених задач. Наприклад, тип, `double` має

діапазон значень від $1.7e-308$ до $1.7e+308$. Однак він може зберігати лише 16 значущих цифр числа і його порядок. Саме тому цей тип представлення даних не придатний для згаданих задач, де необхідний точний результат. При реалізації відомих алгоритмів проведення арифметичних операцій в ЕОМ внаслідок неминучих округлень результату обчислень вже при складанні системи рівнянь та обчисленні поліномів 4 ступеню починають накопичуватися помилки комп'ютерних розрахунків які приводять до виродження матриць і унеможливають отримання результату обчислень [1, 2]. Ці помилки пов'язані з обмеженою довжиною розрядної сітки ЕОМ (що зазвичай дорівнює 32 або 64 біта).

IEEE 754 – широко розповсюджений стандарт представлення чисел та арифметики у ЕОМ. Майже усі сучасні процесори мають апаратне прискорення розрахунків з типами цього стандарту у вигляді FPU чи наборів інструкцій таких як AMD 3DNow! та сімейства IntelSSE.

Основні представлення займають у пам'яті 4 (одинарна точність) та 8 (подвійна точність) байт – що у роки, коли стандарт було представлено здавалось достатнім для ЕОМ тих часів, які у середньому мали 1 Мб ОЗП, з яких програмам було доступно лише 0,64 Мб. У разі використання типу `longdouble`, що має бути як менш таким же точним як `double` та за наявності FPU, може стати доступною можливість використання представлення у 12 байт, але проблема полягає у тому, що навіть це не завжди захищає від нижче оглянутих похибок, а в залежності від компілятора та його конфігурації, мови програмування тип може займати як 8, так і 12 байт (у деяких випадках 16 чотирьох кратна точність) – що робить програму написану з використанням цього типу дуже непередбаченою – результати розрахунків можуть відрізнятись на різних ЕОМ і при цьому не бути вірними. Існуюче у стандарті 32 байт – непередставлення дуже рідкісне і немає відомих апаратних реалізацій.

Зараз апаратні компоненти комп'ютерних систем імітаційного моделювання мають на порядки більші об'єми пам'яті, але ще все користуються стандартом комп'ютерів 1985-го року. Дійсно, для більшості бізнес-задач типів IEEE 754-го стандарту достатньо – фінанси, кількість одиниць товару, але для наукових та інженерних задач навіть якщо діапазон буде достатнім, точність представлення та результатів операцій вже може бути незадовільною.

Оскільки саме представлення чисел у IEEE 754 має похибку, вони існують у будь яких розрахунках [17]. Можливо вчислити максимальну величину цієї похибки – для цього потрібно мінімальне число, яке може зберігати тип поділити на 2 – для `float` це 2-150 (для ненормалізованих чисел) і 2-151 (для нормалізованих чисел), 2-1075 і 2-1076 відповідно для типу `double`.

```
intmain()
{
float a = 123456789;
float b = 123456788;
cout<<a - b;
return 0;
}
```

У результаті виконання наведеного коду у консоль буде виведено число 8, але ясно що різниця чисел 123456789 і 123456788 буде дорівнювати 1.

Комп'ютерне обчислення, без попереднього вивчення діапазону числових значень, призвело до похибки у розмірі 800%.

Точність результату масивно знижується у розрахунках де модуль кінцевого значення в рази менший кожного з аргументів. Але, як було зазначено вище, максимальна абсолютна величина похибки не є дуже великою, тому частіше нею зневажають. Іноді, коли є можливість, цієї похибки можливо уникнути використанням більш точного типу – наприклад, використовуючи тип `double` у попередньому коді. Наведе-ний код слід трансформувати у наступний: у результаті виконання якого у консоль буде виведено число 1:

```
intmain()
{
double a = 123456789;
double b = 123456788;
cout<<a - b;
return 0;
}
```

Зазвичай однакові числа представлені різними типами не будуть рівними між собою:

```
intmain()
{
float a = 123456789.123457;
double b = 123456789.123457;
double c = a - b;
cout<< c ;
return 0;
}
```

У результаті виконання цього коду у консоль буде виведено число 2.87654, але ясно що різниця чисел 123456789.123457 і 123456789.123457 буде дорівнювати 0.

В разі використання однакових типів, результат буде дійсним.

```
intmain()
{
double a = 123456789.123457;
double b = 123456789.123457;
double c = a - b;
cout<< c ;
return 0;
}
```

Слід зазначити також те, що тип результату роботи оператора «/» залежить від мови програмування, чи її компілятора/інтерпретатора. Наприклад якщо зазначити, що операцію проводиться над раціональними числами, дійсно, отримаємо раціональний результат, але в наслідок того що це результат типу double виникає похибка.

```
intmain()
{
float a = 1;
float b = 3;
float c = a / b;
cout<< c - (float)1 / 3;
return 0;
}
```

Що одним з витоків втрати працездатності програмного забезпечення систем імітаційного моделювання є циклічність розрахунків. У результаті виконання розрахунків з числами, що відрізняються у величині, коли різниця експонент приближується до кількості значущих цифр типу, якщо ця різниця більша, то результат розрахунків буде повністю втрачено.

Це означає, що проводити операції можливо лише між числами що знаходяться у деякому діапазоні, що менший ніж діапазон чисел які може представляти формат.

Для чисел існує границя кількості операцій складання, досягши якої результат вже не буде змінюватись.

У матеріалах з професійного програмування існують приклади не є правильними, хоч і виглядають коректними, і досвідчений спеціаліст не буде просто сумувати величини у циклі. Отже, виявляється, що не достатньо лише правильно математично реалізувати алгоритм, а й врахувати похибки що виникають у процесі роботи самої програми.

```
intmain()
{
    floatdetail = 0.000015;
    float detail1 = 0.0;
    floatbuffer = 500.0;
    float buffer1 = 0.0;
    floatoffset = 0.0;

    for (int i = 0; i < 10000000; i++)
    {
        detail1 = detail - offset;
        buffer1 = buffer - detail1;
        offset = (buffer - buffer1) - detail1;
        buffer = buffer - detail1;
    }

    cout<<buffer;
    return 0;
}
```

У результаті виконання цього коду у консоль буде виведено число 350, що є вірним результатом. У прикладі коду використовується алгоритм Кэхэна. У кожній ітерації вище приведеного прикладу враховується похибка попередньої ітерації. Так як IEEE 754 – арифметика не володіє основними властивостями алгебраїчних операцій, це є необхідністю.

Округлення в програмних системах імітаційного моделювання виконується у двох випадках:

- число, що повертається як результат розрахунку округлюється;
- число вводиться або виводиться.

У розрахунках виконується округлення по стандарту IEEE 754(їх чотири виду), стандартне старший розряд, до якого менша різниця.

Навіть якщо складання виконується над однаковими числами, де результат операції буде без похибки, на точність остаточного результату вплине округлення.

В разі виводу числа, наприклад, у Windowsfloat округлюється до 7 знаків, а double до 15. В результаті округлень виникає положення – користувач чи сама програма може вважати нулем число, яке нулем не є.

```
intmain()
{
    double a = 0.1234567890123451234567890;
    cout.precision(7);
    cout<< a;
    cout<<"\n";
    cout.precision(15);
    cout<< a;
    return 0;
}
```

У результаті виконання цього коду у консоль буде виведено числа:0.1234568 та 0.123456789012345. У перший раз (вивід 7 знаків після десятинної коми) число округлюється у більшу сторону, у другий (вивід 15 знаків після десятинної коми) число округлюється у меншу сторону.

У розгалуженнях програмних продуктів таке округлення може вплинути на обраний шлях алгоритму.

Слід також згадати і наведений випадок наведений у [12]. Цей приклад не демонструє один з шляхів виникнення помилок у розрахунках систем імітаційного моделювання, але демонструє ймовірність повної втрати працездатності програмно-апаратного комплексу в разі спроби обчислень з двійковими числами, що знаходяться на межі нормалізованих та ненормалізованих.

IEEE 754 використовує різні способи переведення чисел стандарту у числа раціональні залежно від самого числа – це, по-перше, ускладнює архітектуру процесорів, що підтримує цей формат та вносить невпевненість у тому де визначено перехід з денормалізованого до нормалізованого числа, бо сам IEEE 754 цієї границі не визначає.

Висновки

В статті запропонований шлях до перевірки витоків помилок прогнозування ситуацій в комп'ютерних системах імітаційного моделювання. Результати, що отримані за допомогою аналізу фрагментів програмного коду дозволяють приймати важливі рішення щодо працездатності систем імітаційного моделювання. Відмовою програмних засобів систем імітаційного моделювання слід вважати відсутність тотожності в числових значеннях контрольних прикладів.

Перспективи подальших досліджень

Подальшого вивчення діапазонів числових даних в деяких системах імітаційного моделювання. Особливої уваги потребують методики математичного моделювання бойових дій. В подальшому слід визначити які саме значення параметрів моделювання в двійковій системі обчислення можуть вплинути на результат моделювання.

Список використаних джерел

1. Погрешности измерений и вычислений как причина «катастрофического феномена 1985–1986 годов» в авиационной и ракетно-космической технике /С.Ф. Левин //Контрольно-измерительные приборы и системы.– 2000. – № 3. – С. 21-25
2. Витоки втрати працездатності систем діагностики ОБТ другого роду з представленням чисел з плаваючою комою /А.О. Левченко, Р.М. Войтенков// Сб. науч. труд. Sword.– 2015. – Вип. № 1(38), Том 3. С 4 – 11
3. IEEE Standard for Binary Floating-Point Arithmetic. Copyright 1985 by The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street, New York, NY 10017, USA.
4. Архитектура компьютера / Э. Таненбаум, Т. Остин. – Питербург: Изд. 000 «Питер», 2013. – 811с.
5. Яшкардин В. IEEE 754 – стандарт двоичной арифметики с плавающей точкой // [Электронный ресурс]. – Режим доступа: <http://www.softelectro.ru/teoriy.html> (дата обращения 14.10.2020)
6. Вихідні дані методик для визначення структури системи радіоуправління технічними засобами з урахуванням електромагнітної сумісності / Ю.А. Максименко // Сб. науч. труд. Sword. — 2014. – Вип. № 4(37). – С. 49-52.
7. Визначення оптимальної стратегії радіоуправління технічними засобами у випадку одностороннього відслідковування / Ю. А. Максименко // Праці Одеського політехнічного університету. — 2015. – № 2(46). – С. 155-159.
8. Теорія ігор як метод розробки стратегії узгодження державної політики в сфері інновацій та економічної безпеки /Н.Є. Авдєєва // Сб. науч. труд. КНУТД.– 2017. – Вип. № 2. – С. 49-52.
9. Стратегічне управління інноваційним розвитком підприємства як ключовий фактор його успішного функціонування / О.В. Бойко, М.Я. Іваницька // Економічні науки. Серія “Економіка та менеджмент”: Збірник наукових праць. Луцький національний технічний університет.– Луцьк, 2011. – Випуск 8 (30). – С. 64-71.

10. Семененко О. М., Добровольський Ю. Б., Кострач В. В., Малиш А. Г. Нечітко-множинна математична модель побудови шкали оцінювання рівня воєнно-економічної безпеки // Зб. наук. пр. ЦНДІ ЗС України.– 2016. – № 2 (76). – С. 235–248.
11. Модифицированное зональное сжатие изображений при частичном странении фазовых составляющих спектра / А.В. Королев, С.В. Малахов, Н.Ф. Линник// Системы обработки информации. – Харків: НАНУ, ПАНМ, ХВУ. – 2001. – Вып. 5 (15). – С. 176 – 180.
12. Yashkardin Vladimir, IEEE 754 – binaryfloatingpointarithmeticstandard/IEEE 754 – стандарт двоичной арифметики с плавающейточкой: [Online]. – Access mode: <https://iwww.softelectro.ru> , 04.06.2012
13. Программа IEEE754 – конвертор чисел формата IEEE754 с абсолютной точностью представления результата // [Электронный ресурс]. – Режим доступа: http://www.softelectro.ru/pr_ieee754.htm
14. John L. Gustafson The End of Error: Unum Computing (Chapman & Hall/CRC Computational Science) Chapman and Hall/CRC; 1st edition (February 5, 2015), 416 p.
15. G. Lake, T. Quinnand D. C. Richardson, «From Sir Isaacto the Sloan Survey: Calculating the Structure and Chaos Duetto Gravityin the Universe» Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1997, pp. 1-10.
16. P. H. Hauschildtand E. Baron, “The Numerical Solution of the Expanding Stellar Atmosphere Problem,” Journal Computationaland Applied Mathematics, vol. 109 (1999), pp. 41-63.

References

1. Levin, S.F. (2000). Pogreshnosti izmerenij i vy`chislenij kak prichina «katastroficheskogo fenomena 1985–1986 godov» v aviacionnoj i raketno-kosmicheskoy tekhnike [Measurement and Calculation Errors as the Cause of the «Catastrophic Phenomenon of 1985-1986» in Aviation and Rocket and Space Engineering]. *Kontrol`no-izmeritel`ny`e pribory` i sistemy` – Controlled and measuring devices and systems*, 3, 21-25 [in Russian].
2. Levchenko, A.O., & Vojtenkov, R.M. (2015). Vytoky vtraty pratsezdatnosti system diahnostyky OVT drugoho rodu z predstavleniam chysel z plavaiuchoiu komoiu [Leaks of disability of diagnostic systems of weapons of the second kind with representation of numbers with a floating point]. *Sb. nauch. trud. Sword – Sat. scientific. work. Sword, 1(38), part 3*, 4-11 [in Ukrainian].
3. *EEE Standard for Binary Floating-Point Arithmetic*. Copyright 1985 by The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street, New York, NY 10017, USA. [in English].
4. Tanenbaum, E. & Ostin, T. (2013). *Arkhitektura komp`yutera [Computer architecture]*. Piterburg: Publ 000 «Piter» [in Russian].
5. Yashkardin, V. (2012). IEEE 754 – Standard binary arithmetic float. www.softelectro.ru Retrieved from http://www.softelectro.ru/teoriy_en.html [in Russian].
6. Maksymenko, Y.A. (2014). Vykhidni dani metodyk dlia vyznachennia struktury systemy radioupravlinnia tekhnichnymi zasobamy z urakhuvanniam elektromahnitnoi sumisnosti [Initial data of methods for determining the structure of the radio control system by technical means taking into account electromagnetic compatibility]. *Sb. nauch. trud. Sword – Sat. scientific. work. Sword, 4(37)*, 49-52 [in Ukrainian].
7. Maksymenko, Y.A. (2015). Vyznachennia optimalnoi stratehii radioupravlinnia tekhnichnymi zasobamy u vypadku odnobichnoho vidslidkovuvannia [Determining the optimal strategy of radio control by technical means in the case of one-way tracking]. *Pratsi Odeskoho politekhnichnoho universitetu – Proceedings of Odessa Polytechnic University, 2(46)*, 155-159 [in Ukrainian].
8. Avdieieva, N.I. (2017). Teoriia ihor yak metod rozrobky stratehii uzgodzhennia derzhavnoi polityky v sferi innovatsii ta ekonomichnoi bezpek [Game theory as a method of developing a strategy for coordinating public policy in the field of innovation and economic security]. *Sb. nauch. trud. KNUTD – Sat. scientific. work. KNUTD, 2*, 49-52[in Ukrainian].

9. Boiko, O.V., & Ivanytska, M.I. (2011). Stratehichne upravlinnia innovatsiinym rozvytkom pidpriemstva yak kliuchovy faktor yoho uspishnoho funktsionuvannia [Strategic management of innovative development of the enterprise as a key factor of its successful functioning]. *Zbirnyk naukovykh prats. Lutskiy natsionalnyi tekhnichnyi universytet – Collection of scientific works. Lutsk National Technical University*, 8(30), 64-71 [in Ukrainian].
10. Semenenko, O.M., Dobrovolskyi, Y.B., Kostrach, V.V., & Malysh, A.H. (2016). Nechitko-mnozhyhna matematychna model pobudovy shkaly otsiniuvannia rivnia voienno-ekonomichnoi bezpeky [Fuzzy-plural mathematical model for constructing a scale for assessing the level of military and economic security]. *Zb. nauk. pr. TsNDI ZS Ukrainy – Coll. Science. etc. Central Research Institute of the Armed Forces of Ukraine*, 2 (76), 235–248 [in Ukrainian].
11. Korolev, A.V., Malakhov, S.V., Lynnyk, N.F. (2001). Modyfytsirovannoe zonalnoe szhatye yzobrazhenyi pry chastychnomu stranenyy fazovykh sostavliaiushchykh spektra [Modified zonal compression of images with partial removal of the phase components of the spectrum]. *Systemy obrobky informatsii – Information processing systems. Kharkiv: NANU, PANM, KhVU*, 5(15), 176–180 [in Russian].
12. Yashkardin, V. (2012). IEEE 754 – Standard binary arithmetic float. www.softelectro.ru Retrieved from http://www.softelectro.ru/teoriy_en.html [in Russian].
13. Prohramma IEEE754 - konvertor chysel formata IEEE754 s absolutnoi tochnosti predstavleniya rezultata [The IEEE754 program is a converter of numbers in the IEEE754 format with absolute precision of the result representation]. (n.d.). www.softelectro.ru Retrieved from http://www.softelectro.ru/pr_ieee754.html [in Russian].
14. John, L. Gustafson (2015). *The End of Error: Unum Computing*. Chapman and Hall/CRC; 1st edition [in English].
15. Lake, G., Quinnand, T., & Richardson C. (1997). From Sir Isaacto the Sloan Survey: Calculating the Structure and Chaos Duetto Gravityin the Universe. *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithm. Philadelphia: SIAM* 1-10 [in English].
16. Hauschildtand, P.H., & Baron, E. (1999). The Numerical Solution of the Expanding Stellar Atmosphere Problem. *Journal Computationaland Applied Mathematics*, 41-63 [in English].

Рецензент: Максименко Ю.А., кандидат технічних наук, Військова академія (м. Одеса), Україна

ИСТОЧНИКИ ОШИБОК ПРОГНОЗИРОВАНИЯ СИТУАЦИЙ В КОМПЬЮТЕРНЫХ СИСТЕМАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

И. Шарипова, С. Трутнев, А. Левченко, А. Головки

Современные тенденции развития теории и практики научных исследований показывают, что одними из главных элементов определения достоверности результатов исследований является компьютерное моделирование. С развитием языков программирования высокого уровня и доступности технических средств, значительное количество научных сотрудников используют программные средства моделирования как инструмент исследования. С помощью систем имитационного моделирования принимаются решения, влияющие на человеческие жизни, направления развития отраслей промышленности и перспективы государства. Специалисты в соответствующих отраслях основываясь на итогах компьютерных расчетов не учитывают особенности точности расчетов в компьютере двоичными числами и влияние двоичной арифметики на результаты имитационного моделирования. В статье авторами использован подход определения источников ошибок компьютерного моделирования с помощью фрагментов программного кода. Применение такого подхода дает уже при реализации программных продуктов исключать или определять диапазоны числовых значений представления, которых в двоичном виде может привести к искажению результатов моделирования.

Ключевые слова: представления чисел с плавающей запятой, числовые значения моделирования, математическая модель, архитектура компьютера, ошибки моделирования, ошибки округления, ошибки денормализации.

SOURCES OF SITUATION FORECASTING ERRORS IN COMPUTER SIMULATION SYSTEMS

I. Sharipova, S. Trutniev, A. Levchenko, O. Golovko

Modern trends in the development of the theory and practice of scientific research show that one of the main elements of determining the reliability of research results is computer modeling. With the development of high-level programming languages and the availability of technical means, a significant number of researchers are using modeling software as a research tool. With the help of simulation systems, decisions are made that affect human lives, the directions of development of industries and the prospects of the state. Experts in the relevant industries, based on the results of computer calculations, do not take into account the peculiarities of the accuracy of calculations in a computer with binary numbers and the influence of binary arithmetic on the results of simulation. In the article, the authors used an approach to determine the sources of computer modeling errors using fragments of program code. With the help of fragments of the program code the differences of results of decimal and binary calculations with features of representation of binary numbers with a floating point are shown. It was found that the result of calculations is most influenced by errors of number denormalization, cyclicity errors of of sum operation and rounding errors depending on the description of the accuracy of representation of variables directly in the program code. The use of this approach makes it possible to exclude or determine the ranges of numerical values, the representation of which in binary form, can lead to distortion of the simulation results already during the implementation of software products.

Keywords: *floating point representations, numerical simulation values, mathematical model, computer architecture, simulation errors, rounding errors, denormalization errors.*