УДК 004.43:004.222.3

И.И. Жульковская, О.А. Жульковский

СОВРЕМЕННЫЕ СРЕДСТВА ПОВЫШЕНИЯ ТОЧНОСТИ РЕЗУЛЬТАТОВ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Аннотация. Повышение точности арифметических операций за счет увеличения разрядности чисел является основным средством выполнения расчетов, критичных к округлениям. Проведенное исследование показывает возможности современных процессоров общего назначения и компиляторов языков высокого уровня поддерживать форматы повышенной точности для повышения эффективности математического моделирования в целом.

Ключевые слова: число с плавающей запятой, стандарт IEEE 754, формат представления, граничные значения, абсолютная погрешность вычисления.

Постановка проблемы

Решение современных задач, формализованных в виде математических моделей, требует крайне сложных вычислений над огромными массивами данных, обработка которых содержит значительное число итераций. При этом точность машинных вычислений становится неудовлетворительной, а вычислительная погрешность определяет основную долю ошибки в получаемом решении. Одним из источников вычислительных погрешностей является приближенное представление действительных чисел в ЭВМ, обусловленное конечностью разрядной сетки.

Анализ публикаций по теме исследования

Действительные данные в памяти вычислительной системы представлены в формате с плавающей запятой (точкой) в соответствии со стандартом IEEE Std 754–2008 (ISO/IEC/IEEE 60559:2011).

Авторы многих публикаций [1, 2 и др.] предлагают различные подходы к стандартизации математических функций, работающих с числами с плавающей запятой в форматах IEEE 754. Наряду с этим, множество работ посвящено тестированию на соответствие стандарту [2–4], вопросам погрешностей результатов при вычислении функций

[5]. Авторами настоящей работы ранее исследован и описан алгоритм современного подхода к формированию машинного представления и хранения числовой информации в формате с плавающей запятой, рассмотрены особенности представления, а также вычислены граничные (максимальные и минимальные) значения субнормальных и нормализованных чисел [6, 7 и др.].

Формулировка целей статьи

Целью настоящего исследования является повышение достоверности результатов моделирования на основе оценки современных, наиболее часто используемых аппаратных и программных средств на предмет получения наибольшей точности вычислений и снижения погрешностей, обусловленных представлением действительных данных в памяти компьютера.

Основная часть

Как известно, значение двоичного числа с плавающей запятой определяется выражением:

$$A = \pm a_0.a_1a_2a_3...a_{n-1} \times S^e \quad (1)$$

где S – основа системы счисления; n – число значащих разрядов мантиссы; ai – цифры (0≤ ai <S); e – порядок или экспонента (не путать с числом e).

Стандарт IEEE 754 определяет основные параметры форматов представления чисел с плавающей запятой (табл.1).

Таблица 1

Параметры основных двоичных форматов чисел с плавающей запятой

формат	всего бит	бит в порядке	бит в мантиссе	смещение порядка
binary32 single	32	8	23	127
binary64 double	64	11	52	1023
binary128 quadruple	128	15	112	16383

В работе [5] описано машинное представление граничных (максимальных и минимальных) значений чисел с плавающей запятой в стандарте IEEE 754, получены формулы для вычисления и рассчитаны граничные значения для различных форматов этого стандарта (табл.2).

 ISSN 1562-9945

Таблица 2

Граничные значения нормализованных чисел с плавающей запятой

формат	Максимальное значение	Минимальное значение
binary32	±3.402823E+38	±1.175494E-38
binary64	±1.797693E+308	±2.225073E-308
binary128	±1.189731E+4932	±3.362103E-4932

Особенность представления действительных чисел со скрытой единицей заключается в том, что имеется довольно большой разрыв между нулем и ближайшим к нему представимым числом – потеря значимости (underflow) около нуля. Это обстоятельство может приводить к ошибкам при работе с малыми величинами.

Для повышения точности вычислений при работе с «маленькими» числами в стандарте предусмотрена возможность использования т.н. субнормальных (subnormal numbers), т.е. ненормализованных чисел. Использование денормализованных чисел - это способ увеличить количество представимых чисел с плавающей запятой для повышения точности вычислений.

Описание машинного представления и особенностей использования субнормальных чисел в стандарте IEEE 754, а также получение формул для вычисления и сам расчет граничных значений (максимальных и минимальных) субнормальных чисел для различных форматов (табл.3) рассмотрены авторами данной статьи в работе [8].

Таблица 3

Граничные значения субнормальных чисел с плавающей запятой

формат	Максимальное значение	Минимальное значение
binary32	±1.175494E-38	±1.401298E-45
binary64	±2.225073E-308	±4.940656E-324
binary128	±3.362103E-4932	±6.475175E-4966

Представимые в формате с плавающей запятой числа должны быть двоично-рациональными, т.е. должны иметь вид $p / 2^m$, где p и m – целые числа, причем m неотрицательно. Только в этом случае число представляется в формате с плавающей запятой без округления и, соответственно, без потери точности. Остальные же числа представлены приближенно, т.е. они округляются до точно представимых в формате с плавающей запятой чисел.

Основным показателем качества машинной арифметики с плавающей запятой считается точность (accuracy), с которой арифметика округляет действительные числа. Чем больше расстояние от исходного числа до ближайшего представимого, тем с меньшей точностью оно может быть представлено. Расстояние между соседними представимыми числами, т.е. числами с единым десятичным значением порядка и с различающимися в один бит мантиссами, называют шагом числа. Следовательно, максимальная абсолютная погрешность округления для числа в формате IEEE 754 равна половине шага, а вычисленный результат отличается от точного значения не более чем на половину единицы последнего разряда мантиссы результата (unit in the last place, ulp) [8]. Шаг чисел удваивается с увеличением экспоненты двоичного числа на единицу. Т.е. чем дальше от нуля, тем шире шаг чисел в формате IEEE 754 по числовой оси.

Авторами настоящей работы рассмотрены [9] особенности стандартных способов округления чисел с плавающей запятой при вычислениях на компьютере и получены выражения для вычисления максимальных абсолютных погрешностей округления чисел, представленных в базовых форматах стандарта IEEE 754 (табл.4).

Таблица 4

Выражения для вычисления максимальных абсолютных погрешностей округления чисел базовых форматов стандарта IEEE

формат	субнормальные числа	нормализованные числа
binary32	2^{-150}	2^{E-151}
binary64	2^{-1075}	2^{E-1076}
binary128	2^{-16495}	$2^{E-16496}$

Из всех вышеприведенных таблиц ясно видно, что для получения наиболее точного результата вычислений необходимо использовать формат четырехкратной точности (quadruple precision).

Аппаратная поддержка binary128 чрезвычайно редка, например в z/Architecture и POWER9; в SPARC V8 и V9 заявлена, но реальная аппаратная поддержка отсутствует.

Авторами работы проведено исследование современных аппаратных и программных средств, обеспечивающих реализацию алгоритмов численного экспериментирования, отличающихся повышенной точностью получаемых результатов. Установлено [10], что наиболее популярные при математическом моделировании языки програм-

мирования C/C++ согласно стандарту предоставляют три типа с плавающей запятой: float, double и long double.

Типы float и double практически всюду соответствуют binary32 и binary64 стандарта IEEE 754. Тип long double менее однозначен. Для x86/x86_64 в компиляторах GCC и Clang он по умолчанию соответствует 80-битному расширенному формату Intel x87. Для Visual Studio (начиная с 32-битных версий) и ICC он физически соответствует тому же типу что и double, хотя логически является самостоятельным типом (в Visual Studio поддержка 80-битного расширенного типа присутствовала в 16-битных версиях). Для ICC под Windows можно воспользоваться флагом /Qlong-double, который меняет long double на 80-битный x87 (тип занимает 16 байт из соображений выравнивания).

Размер типа long double в смысле значения sizeof(long double) для 80-битного типа x87 типично не равен 10 байтам из соображений выравнивания – реальные значения обычно равны 12 или 16 байт. Для GCC это значение можно изменить соответствующими опциями компиляции. Неиспользуемые байты при этом имеют произвольные значения.

На некоторых платформах long double означает binary128 либо использует пару 64-битных действительных чисел для достижения 106-битной точности, но с диапазоном обычного double. На многих платформах (в частности x86/x86_64) GCC по умолчанию поддерживает (предоставляет в качестве расширения) нестандартный тип __float128 (а также __float80). Реализация этого типа (на архитектурах, не поддерживающих binary128) – программная и, соответственно, «медленная». В некоторых версиях ICC есть поддержка 128-битного типа с плавающей запятой _Quad.

Выводы и перспективы дальнейших исследований

Т.к. действительные числа – это бесконечное множество, в то время как их машинная реализация числами с плавающей запятой представляет собой конечное множество, то для создания корректных программ, а также минимизации ошибок вычислений, важно понимать особенности представления и хранения в ЭВМ граничных значений таких чисел.

Для современных цифровых вычислительных систем очень важной проблемой является ограниченная размерная сетка, используемая для представления данных. Это, в свою очередь, приводит к

дополнительной погрешности результатов вычисления, которая может оказаться соизмеримой с величиной исходных данных.

Проведенное исследование показывает возможности современных процессоров общего назначения и компиляторов языков высокого уровня поддерживать форматы повышенной точности для повышения эффективности математического моделирования в целом.

ЛИТЕРАТУРА

1. Some Notes for a Proposal for Elementary Function Implementation in Floating-Point Arithmetic / G. Hanrot, V. Lefevre, J.-M.Muller, N. Revol and other // Proc. of Workshop IEEE 754 and Arithmetic Standardization, in ARITH-15.– 2001.

2. Proposal for a standardization of mathematical function implementation in floating-point arithmetic / D. Defour, G. Hanrot, V. Lefevre, J.-M.Muller and other // Numerical Algorithms.– 2004.– №37(1–4).– P.367–375.

3. A Test Generation Framework for Datapath Floating-Point Verification / M. Aharoni, S. Asaf, L. Fournier, A. Koifman and other // IEEE International High Level Design Validation and Test Workshop.– 2003.– P.17–22.

4. Stehle D. Searching Worst Cases of a One-Variable Function Using Lattice Reduction / D.Stehle, V.Lefevre, P. Zimmermann // IEEE Transactions on Computers.– 2005.– №54(3).– P.340–346.

5. Никонов О.Я. Оценка точности вычислений специальных функций при разработке компьютерных программ математического моделирования / О.Я.Никонов, О.В.Мнушка, В.М.Савченко // Вісник НТУ «ХПІ». Тематичний випуск: Інформатика і моделювання.– Харків: НТУ.– 2011.– №17.– С.115-121.

6. Жульковская И.И. Вычисление граничных значений субнормальных чисел в IEEE-стандарте / И.И.Жульковская, О.А.Жульковский, Р.Г.Шаганенко // Математичне моделювання. – Дніпродзержинськ: ДДТУ.– 2015.– №1 (32).– С.41-44.

7. Жульковская И.И. Вычисление граничных значений действительных числовых данных в IEEE-стандарте / И.И.Жульковская, О.А.Жульковский, Ю.В.Николаенко // Зб. наук. праць ДДТУ (технічні науки).– Дніпродзержинськ, ДДТУ.– 2015.–№.1 (26).– С.240-245.

8. Goldberg D. What Every Computer Scientist Should Know about Floating-Point Arithmetic / D. Goldberg // ACM Computing Surveys.– 1991.– №.23(1).– P.5-48.

9. Жульковская И.И., Жульковский О.А. Вычисление максимальных абсолютных погрешностей округления чисел в IEEE-стандарте / И.И.Жульковская, О.А.Жульковский // Математичне моделювання.– 2015.– №2 (33).– С.33-36.

10.Жульковская И.И. Современные средства аппаратной и программной поддержки IEEE-стандарта / И.И.Жульковская, О.А.Жульковский, А.Д.Журавский // Математичне моделювання.– Кам'янське: ДДТУ, 2017.– №1 (36).– С.3-6.