МАТЕМАТИКА

Сергей Арнольдович ИНЮТИН — проректор по научной работе Сургутского государственного педагогического института, доктор технических наук, доцент

УДК 681.26

ВЫЧИСЛИТЕЛЬНЫЕ ЗАДАЧИ БОЛЬШОЙ АЛГОРИТМИЧЕСКОЙ СЛОЖНОСТИ И МОДУЛЯРНАЯ АРИФМЕТИКА

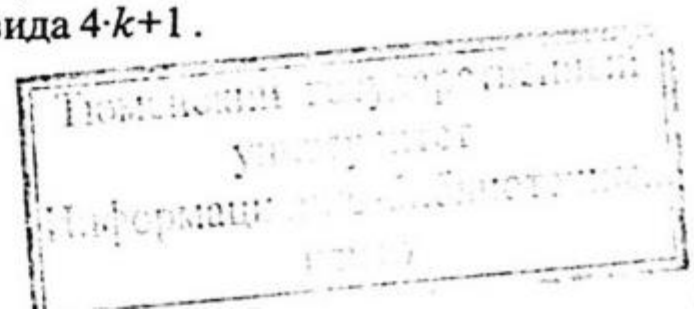
АННОТАЦИЯ. Автором введены математические понятия, базовые для вычислительных средств на основе модулярной машинной арифметики.

The author introduces basic mathematical notions for calculation means within modular computer arithmetic.

Вычисления с величинами, меняющимися в больших диапазонах, сводятся к вычислениям с многоразрядными числами и являются одной из областей, в которых модулярные вычислительные средства (на модулярной вычислительной базе) имеют преимущества перед иными [1]. Важные для теории и практики математические задачи, требующие для таких вычислений больших вычислительных ресурсов, лежат в областях прикладной и вычислительной теории чисел [2]. Большинство таких задач (или проблем) содержит целочисленные вычисления с числами или числовыми величинами, принимающими значения из больших и сверхбольших машинных диапазонов. В настоящее время интенсивно развивается прикладная теория чисел, отвечая потребности в разработке средств надежной передачи, хранения и обработки коммерческой и иной цифровой информации. При этом возникает широкий спектр вычислительных задач [1,2], приводящих к вычислениям, при которых значения целочисленных переменных значительно, в 10^3,...10^6 и более раз превышают максимум типового компьютерного диапазона серийной вычислительной техники, определяемого длиной аппаратно-поддерживаемого машинного слова. Назовем такой диапазон большим целочисленным компьютерным диапазоном. Наличие эффективных методов вычислений в больших диапазонах позволяет ставить задачи вычислений в сверхбольших диапазонах, максимум которых достигает константы Виноградова– Гольбаха $3^{3^{15}}$.

Перечислим ряд задач, часто называемых из-за их временной сложности вычислительными проблемами [3].

1. Тестирования на простоту чисел специального и произвольного вида.
2. Нахождение канонического мультипликативного разложения числа.
3. Поиск больших и сверхбольших простых чисел вида $4 \cdot k+1$.
4. Поиск псевдопростых чисел и чисел-близнецов.



5. Поиск нечетного совершенного числа.
6. Поиск цепочек простых чисел в арифметических прогрессиях.
7. Проверка гипотезы о местоположении нулей дзета-функции Римана.

Перечень не является исчерпывающим. Особенностью указанных задач является невозможность их решения в настоящее время только аналитическими или алгебраическими методами, и по этой причине находят широкое использование вычислительные методы при поиске полного или частичного решения, контрпримеров. На некоторые из этих проблем современная точка зрения такова, что найти их удовлетворительные решения возможно только вычислительными методами [3]. Решения перечисленных задач имеют большое теоретическое значение [2]. На данный момент даже частные решения, полученные вычислительными методами, отдельных из указанных вычислительных проблем находят, наряду с теоретическим, также широкое практическое приложение.

При классификации методов вычислений в больших и сверхбольших компьютерных диапазонах необходимо учитывать, наряду с постановкой исходной задачи, особенности модулярных вычислительных процессов [4].

Задачи вычислений в сверхбольших компьютерных диапазонах (СБД) и сами понятия больших и сверхбольших величин определены ранее [4,5]. Введем классификацию методов вычислений в СБД, взяв в качестве типовых методы и алгоритмы, возникающие при тестировании на простоту чисел специального вида: Ферма -F_n или Мерсенна – M_n.

Algorithm Pepen.

1. $m \leftarrow 0$, $A \leftarrow 3$
2. $A \leftarrow A^2 (mod\ F_n)$, $m \leftarrow m+1$
3. *if* $2^n - m = 0$ *then end else goto* 2
4. *end: if* $A = 2^{2^n}$ *then* F_n *- prime else* F_n *–not prime*

Algorithm Lucas-Lehmer.

1. $m \leftarrow 0$, $A \leftarrow 4$
2. $A \leftarrow A^2 - 2 (mod\ M_n)$, $m \leftarrow m+1$
3. *if* $m = n-2$ *then end else goto* 2
4. *end: if* $u \equiv 0 (mod\ M_n)$ *then* M_n *- prime* else M_n *–not prime*

Базовой операцией в алгоритмах является вычисление вычета от некоторой сверхбольшой величины по модулю, являющемуся большой величиной. Каноническое разложение большого модуля F_n или M_n неизвестно.

$$C = | A^{f(B)} |_B = A^{f(B)} - N \cdot B ,$$

где $A, B, f(B)$ — большие числовые величины; $A^{f(B)}$ — сверхбольшая величина.

Так как $A^{f(B)}$ — сверхбольшая числовая величина, то прямые алгоритмы, базирующиеся на вычислении соотношения $C = A^{f(B)} - [A^{f(B)} / B] \cdot B$, невозможно или нецелесообразно реализовать на вычислительной технике из-за большой алгоритмической и временной сложности.

Модулярная арифметика квадратичного диапазона МС(P^2) позволяет рассматривать это равенство в форме сравнения по модулю $P^2 > C$:

$$C(mod\ P^2) \equiv (A^{f(B)}(mod\ P^2) - [A^{f(B)} / B](mod\ P^2) \cdot B(mod\ P^2))(mod\ P^2)$$

Возможны два способа итерационного вычисления величины $C(mod\ P^2)$.

Определение. Вычетным итерационным алгоритмом 1-го рода называется алгоритм, который вычисляют при $B < P^2$, в частности при $B < P$, большую величину $C(mod\ P^2) : C(\text{mod}\ P^2) = | A^{f(B)} |_B\ (\text{mod}\ P^2)$ как последовательность:

$$C(\text{mod}\ P^2) = | \ldots \| A^{f_1(B)} |_B^{f_2(B)} |_B \ldots |_B\ (\text{mod}\ P^2) ,$$

при этом предполагается выполнение соотношения, характеризующего мультипликативную структуру показателя степени, например заданного каноническим мультипликативным разложением: $f(B) = f_1(B) \cdot f_2(B)$..... .

Определение. Вычетным итерационным алгоритмом 2-го рода называется алгоритм, который вычисляет при $B<P^2$ сверхбольшую величину $N(mod\ P^2)$, а затем большую величину $C(mod\ P^2)$:

$$N(mod\ P^2) = [\ A^{f(B)} / B\ ](mod\ P^2) = ((\ A^{f(B)} - X_i(mod\ P^2))\cdot |B|^{-1}\ )(mod\ P^2),$$

где последовательность больших величин X_i, генерируемая и оцениваемая некоторым образом, стремится к $C(mod\ P^2)$.

Отличие между алгоритмами 1, 2-го родов заключается в том, что для модулярного алгоритма 2-го рода в МС(P^2) целая величина $N(mod\ P^2)$, вычисляемая методом формального деления в МС(P^2), должна совпасть с целой величиной N, полученной фактическим делением на B .

Среди итерационных вычетных алгоритмов 1,2 -го родов целесообразно выделение подкласса алгоритмов, основанных на оценивании интервала, в котором может находиться сверхбольшая величина. Например, для $F_n n$ – числа Ферма выполняются неравенства: $2^{f(F_n)} < 3^{f(F_n)} < 2^{2\cdot f(F_n)}$.

Если при таком оценивании границы вычисляются с меньшей алгоритмической сложностью и их разность меньше значения большой величины — P^2, то возможно вычисление соответствующего вычета как алгебраической суммы вычета от значения границы и некоторой фиксированной разницы искомой сверхбольшой величины и границы интервала оценивания:

$$|3^{f(F_n)}|_B = ||2^{f(F_n)}|_B + |\Delta|_B|_B \ (\mathrm{mod}\ P^2)\ .$$

Определение. Алгоритм называется вычетным итерационным оценочным алгоритмом 1-го рода, если вычисляет большую величину $C(mod\ P^2)$ как алгебраическую сумму по модулю B в МС(P^2) некоторой сверхбольшой величины R и разности двух сверхбольших величин:

$$C = |R|_B + |R - A^{f(B)}|_B - 0|\ B(mod\ P^2\ )\ .$$

Определение. Алгоритм называется вычетным итерационным оценочным алгоритмом 2-го рода, если вычисляет сверхбольшую величину $N = [\ A^{f(B)} / B\ ]$ как алгебраическую сумму в МС(P^2) некоторой сверхбольшой величины M и разности искомой сверхбольшой величины N и M по модулю P^2 , что позволяет большую величину C вычислять в соответствии с соотношением:

$$C(\mathrm{mod}\ P^2) = (A^{f(B)}(\mathrm{mod}\ P^2) - (M(\mathrm{mod}\ P^2) - (N - M)(\mathrm{mod}\ P^2))\cdot B)(\mathrm{mod}\ P^2)\ .$$

Для вычислений с модулярными операндами в большом компьютерном диапазоне введем классификацию преобразований над модулярными величинами, применительно к модулярной системе квадратичного диапазона МС(P^2). Классификация вводится впервые для модулярных систем как одинарного, так и квадратичного диапазонов. Определим следующие понятия.

Определение. Модулярный функционал — функция от n -переменных, аргументами которой являются значения компонент-вычетов одной или более модулярных величин. Область определения функционала — подмножество модулярных величин $A\ (mod\ P^2\ )$ из МС(P^2) , область значений — подмножество натуральных, целых, рациональных, действительных, комплексных чисел.

Функционал может вычисляться по некоторому модулю, не совпадающему с основаниями МС(P^2): $f = f(A(\mathrm{mod}\ P^2))(\mathrm{mod}\ Q) = f(a_1,...,a_n)(\mathrm{mod}\ Q)$.

Нелинейный функционал, устанавливающий биективное отображение между множеством модулярных величин и их числовыми значениями, позволяет найти числовое значение модулярной величины.

$$A = f_1(a_1,...,a_n) - f_2(a_1,...,a_n)\cdot P^2 = f_1(a_1,...,a_n)(\mathrm{mod}\ P^2)\ .$$

Определение. Модулярная функция — отображение множества модулярных величин из МС(P^2) на (или в) множество модулярных величин из этой же МС(P^2), характеризующееся тем, что модулярная функция может быть представлена как совокупность n -функционалов, задающих отображение компонент модулярных величин-аргументов в компоненты модулярной величины — значения модулярной функции. Область определения и область значений суть некоторые подмножества МС(P^2).

Определение. Модулярной операцией называется модулярная функция, в которой совокупность функционалов зависит от компонент одной или двух модулярных величин — операндов модулярной операции.

Для вычислительной практики важны два класса модулярных операций.

Определение. Модульная операция унарная или бинарная — модулярная функция, в которой $\forall i = i \div n$ каждый i-й функционал зависит только от i-х компонент одной или двух модулярных величин из МС(P^2), являющихся операндами модульной операции. $\forall i = i \div n$

$$\begin{cases} s_1 = \varphi(a_1, b_1)(\text{mod } p_1) \\ s_i = \varphi(a_i, b_i)(\text{mod } p_i) \\ s_n = \varphi(a_n, b_n)(\text{mod } p_n). \end{cases}$$

Замечание. Если множество-носитель МС(P^2) рассматривать как алгебраическое кольцо, то все кольцевые операции — модульные.

Подмножество модульных величин из МС(P^2), имеющих обратные величины, является алгебраическим полем, в этом случае умножение на обратный элемент также является модульной операцией.

МС(P^2) есть декартово произведение классов вычетов по квадратам простых модулей, является алгебраическим кольцом, точнее телом. Множество наименьших неотрицательных или абсолютно наименьших вычетов (классов вычетов) по квадратам простых чисел является полем Галуа $GL(p_i^2)$, для которого рассматриваются, как правило, только кольцевые свойства для введенных модульных операций. Причина этого в использовании кольцевой структуры МС(P^2) и кольцевых модульных операциях в вычислительных системах [5,6].

Определение. Немодульная операция унарная или бинарная — модулярная функция, в которой $\forall i = i \div n$ каждый i-й функционал зависит от более чем одной, в общем случае — от n компонент-вычетов одной или двух модулярных величин из МС(P^2), являющихся операндами операции. $\forall i = i \div n$:

$$\begin{cases} s_1 = \varphi(a_1, ..., a_n, b_1, ..., b_n)(\text{mod } p_1) \\ s_i = \varphi(a_1, ..., a_n, b_1, ..., b_n)(\text{mod } p_i) \\ s_n = \varphi(a_1, ..., a_n, b_1, ..., b_n)(\text{mod } p_n) \end{cases}.$$

Определение. Модулярный оператор — отображение модулярных величин из одной модулярной системы МС(B^2) на (или в) множество модулярных величин другой модулярной системы МС(E^2).

Модулярные операторы позволяют строить помехозащитные коды — подмножества избыточных модулярных систем. Модулярный оператор может действовать из МС(P^2) на МС(P^2) , тогда он является модулярной функцией.

Определение. Вычислительный модулярный процесс — компьютерная реализация модулярных функций (или операторов) в виде суперпозиции модулярных функций, далее отображаемых суперпозицией модулярных операций.

Определение. Вычислительный модулярный процесс — немодульный, если суперпозиция содержит хотя бы одну немодульную операцию.

Определение. Преобразование кодонезависимого вычислительного процесса в вычетный вид или модулярный вычислительный процесс есть отображение функциональных преобразований (или операторов) исходного процесса в некоторую суперпозицию модулярных функций.

Определение. Модулярный вычислительный процесс — корректный, если получен изоморфным преобразованием в вычетный вид кодонезависимого вычислительного процесса, а также если при этом преобразовании сохраняется изоморфизм между областями определения и значений исходного вычислительного процесса и диапазонами модулярных систем, являющихся соответственно областями определения и значений модулярного вычислительного процесса.

Непосредственно из определений следуют.

Свойство 1. Модулярный вычислительный процесс является распараллеливаемым на n-вычислительных процессов, содержащих операции по i-модулю — основанию МС над i-компонентами операндов по p_i^2-модулю.

Свойство 2. Модулярный процесс имеет меньшую временную сложность, если содержит минимальное количество немодульных операций.

Свойство 3. Модулярный процесс имеет меньшую временную сложность, если немодульные операции содержат только линейные функционалы.

Свойство 4. В модулярной арифметике возможен выход за пределы диапазона на промежуточных этапах выполнения вычислительного процесса.

Этого свойства нет в позиционной и полиадической арифметиках [5,6].

Теорема. *О модулярных диапазонах.*

Для корректности модулярного вычислительного процесса необходимо, чтобы значение модулярного вычислительного диапазона было больше максимума объединения двух множеств значений выходных данных для кодонезависимого вычислительного процесса и значений модулярных диапазонов для областей определения немодульных операций.

Доказательство

Модулярный вычислительный процесс разделяется на этапы, содержащие модульные и немодульные операции.

Рассмотрим модульный этап процесса. Пусть для чисел компонент входных модулярных величин a_i, b_i для индексов выполняется $i=1\div n$, а для выходных компонент s_j $j=1\div m$, причем $m=n$.

Максимальное числовое значение модулярной величины из области значений модульного вычислительного процесса не превышает максимума модулярного диапазона для входных или выходных величин, следовательно:

$$s_i = f_1(\ldots f_k(a_i, b_i)\ldots)\ , \ i=j=1\div n\ .$$

Рассмотрим немодульный этап вычислительного процесса. Пусть немодульная операция определена для модулярных величин, имеющих вычеты: a_i, b_i $i=1\div e$, а для выходных величин: s_q, $q=1\div m$: $s_q = h(a_1,\ldots,a_e,b_1,\ldots,b_e)$.

Пусть далее выполняются только модульные операции, этот случай рассмотрен ранее. Следовательно, должно выполняться: $i, j, q = 1 \div \max\{m, e\}$.

Количество оснований МС при фиксированном наборе определяет значение модулярного диапазона [4]: $P^2 = \prod_{i=1}^{\max\{m,e\}} p_i^2$.

Следствие С- 1. При численных вычислениях максимальные значения из теоремы заменяются на меньшие, возникающие при таких вычислениях.

Следствие С- 2. На модульных промежуточных этапах вычислительного модулярного процесса максимумы модулярных диапазонов могут быть меньше значений, требуемых для данных кодонезависимого вычислительного процесса.

Следствие С- 3. Значение модулярного вычислительного диапазона можно выбрать равным максимальному из значений диапазонов входных и выходных модулярных величин, оно может меняться в динамике вычислительного процесса, в частности увеличиваться для входных операндов перед выполнением немодульной операции.

Функциональные преобразования для двух классов вычислительных модулярных процессов выполняются в основном вычислительном диапазоне МС.

Определение. Прямой вычислительный модулярный процесс есть суперпозиция некоторых модулярных функций (или операторов). Результат будет получен сразу после завершения выполнения конечного количества функций (или операторов) из суперпозиции.

Определение. Итерационный вычислительный модулярный процесс есть суперпозиция модулярных функций, в которой можно выделить такие повторяющиеся последовательности, что результат одной последовательности есть входные данные для другой, при этом количество повторений выделенных последовательностей для вычислительного процесса не фиксируется заранее, а определяется в процессе счета сравнением результатов при различных вычислениях в выделенных последовательностях в суперпозиции.

$$A_{i+1} = f(A_i, A_{i-1}, ..., A_{i-l}) .$$

Критерий останова задается вычислительной проверкой на истинность некоторого предиката-функционала: $R(A_{i+1}, A_i) = const$.

Введем ряд новых нетривиальных математических конструкций, в частности производные в модулярных системах, адаптированные к их специфике.

Определение. Первая производная от модулярной величины $A\,(mod\ P^m)$, заданной в МС(P^m), с каноническим представлением:

$$A\,(mod\ P^m) = "A + K_1 \cdot P + K_2 \cdot P^2 \ldots + K_{m-1} \cdot P^{m-1}\ (mod\ P^m)$$

есть модулярная величина в МС(mP^{m-1}) с каноническим представлением:

$$A^1\ (mod\ mP^{m-1}) = K_1 \cdot + 2K_2 \cdot P \ldots + (m-1)\ K_{m-1} \cdot P^{m-2}\ (mod\ mP^{m-1}) .$$

Определение. Обобщение, производная l- порядка есть модулярная величина с каноническим представлением:

$$A^{(l)} = (A^{(l-1)})'(\mathrm{mod}\ m(m-1)...(m-l+1)P^{m-l}) .$$

Замечание. Для m=3 получим по определению $A^{(m)} \equiv 0$ в $\mathbf{Z}$.

$A^1\ (mod\ P^2) = K_1 \cdot + 2K_2 \cdot P\ (mod\ 3P^2)$

Коэффициент 3 в модуле является необходимым, т. к.

$P^2 \leq \max A' < 2P^2$, $\max A' = (P-1)(2P+1)$,

$A'' = 2K_2 (\mathrm{mod}\ 6 \cdot P)$, $A''' = 0$.

Для m=2 коэффициент два в модуле можно не учитывать, получим:

$$A^1\ (mod\ P) \equiv K_1 \cdot (mod\ 2P) = |K_1|_P\ (\mathrm{mod}\ 2P) = K_1 (\mathrm{mod}\ P^2) .$$

Очевидна аналогия с конечными разностями. Для квадратичного случая m=2 получим:

$$A'(\mathrm{mod}\ P) = (A - |A|_P)(\mathrm{mod}\ P^2) / P = K_1 (\mathrm{mod}\ P) .$$

Для произвольного случая m получим:

$A^{(1)}\ (\mathrm{mod}\ mP^{m-1}) = (K_1 \cdot P + 2K_2 \cdot P^2 \ldots + (m-1)\ K_{m-1} \cdot P^{m-1})/P =$

$= K_1 \cdot + 2K_2 \cdot P \ldots + (m-1)\ K_{m-1} \cdot P^{m-2}\ (mod\ mP^{m-1})$

$A^{(2)}\ (\mathrm{mod}\ m(m-1)P^{m-2}) = 2K_2 \cdot + 6 \cdot K_3 \cdot P\ (\mathrm{mod}\ m(m-1)P^{m-2})$

.....

$A^{(m-1)}\ (\mathrm{mod}\ m!\ P) = (A^{(m-2)} - |A^{(m-2)}|_p)/P = (m!)\ K_{m-1} \cdot\ (\mathrm{mod}\ m!\ P)$

$A^{(m)} \equiv 0$.

ЛИТЕРАТУРА

1. Ноден П., Китте К. Алгебраическая алгоритмика. М., 1999. 720 с.
2. *Hans Riesel* Prime Numbers and Computer Methods for Factorization. Stuttgart-Boston, 1985. 452 p.
3. *Munro I.* The Computational Complexity of Algebraic and Numeric Problems // American Elsevier. New-York. 1986. № 7. P. 28-40.
4. Инютин С. А. Компьютерная модулярная алгебра квадратичного диапазона и область ее приложения // Вестник Тюменского госуниверситета. Тюмень. 2001. № 2. С. 141-148.
5. Инютин С. А. Арифметико-логические основы вычислительных систем. Сургут, 2001.117 с.
6. Инютин С. А. Модулярные вычисления в сверхбольших компьютерных диапазонах // Электроника. 2001. № 6. С. 54-61.

Владимир Николаевич КУТРУНОВ — декан факультета математики и компьютерных наук, доктор физико-математических наук, профессор;
Елена Борисовна ОРЛОВА — старший преподаватель кафедры математического моделирования

УДК 519.6

ЭКСПЕРИМЕНТЫ С ИТЕРАЦИОННЫМ МЕТОДОМ ОПЕРАТОРНОГО ПОЛИНОМА НАИЛУЧШЕГО ПРИБЛИЖЕНИЯ

АННОТАЦИЯ. В работе сопоставлены несколько вариантов нового итерационного двухточечного стационарного метода, построенного на основе алгебраического полинома наилучшего равномерного приближения (метод ПНП), пригодного для решения линейных операторных уравнений .

Several variants of new iteration two-point stationary method built on the base of the algebraic multinomial of the best uniform approximation (the method MBA) suitable to solve linear operator equations are matched.

Решаемое линейное операторное уравнение имеет вид:

$$(\mathbf{aI} - \mathbf{T})\mathbf{X} = \mathbf{B}, \qquad (1)$$

где **a** — некоторое число; **X, B** — искомый и заданный элементы банахова пространства **E**; **I** — тождественный, а **T** — линейный ограниченный операторы с действительным спектром. Особенности метода ПНП проще показать на примере решения простейшего алгебраического уравнения

$$(a - t)x = b, \qquad (2)$$

здесь **a, t, b, x** — действительные числа. Если в (2) запретить операцию деления на число t, тогда точное решение, использующее деление

$$x = (a - t)^{-1}b\,, \qquad (3)$$

не может быть получено. Это соответствует отсутствию явно найденного обратного оператора $(\mathbf{aI} - \mathbf{T})^{-1}$ в операторном уравнении (1). Необходимо построить метод решения уравнения (2) без использования операции деления, затем перенести эту технику на операторное уравнение (1). Простейший путь опирается на использование по-