Вычислительные технологии **Том 5, № 2, 2000**

УНИВЕРСАЛЬНЫЕ АЛГОРИТМЫ И ВЫЧИСЛЕНИЯ С ПРОИЗВОЛЬНОЙ ТОЧНОСТЬЮ*

Г. Л. Литвинов, Е. В. Маслова
Международный центр "Софус Ли", Москва, Россия
e-mail: litvinov@islc.msk.su

The notion of a universal algorithm is introduced and specific examples of such algorithms are considered which enable one to perform calculations with the selected accuracy as well as the specific features of their software realization.

1. Математические объекты и их машинные представления

Вычислительные алгоритмы представляют собой организованные наборы некоторых основных операций. Основные операции манипулируют с данными, описывающими "числа". Эти "числа" являются элементами некоторого "числового" домена, т. е. математического объекта типа поля действительных чисел, кольца целых чисел, идемпотентного числового полукольца (об идемпотентных полукольцах и их роли в рамках идемпотентной математики см., например, в [1–3]). При каждом конкретном компьютерном вычислении элементы указанных числовых доменов заменяются на их машинные представления, т. е. элементы некоторых конечных моделей этих доменов. Примерами моделей, удобных для компьютерного представления действительных чисел, являются различные варианты арифметики с плавающей запятой, приближенная арифметика рациональных чисел (см., например, [4]), версии интервальной арифметики и др. Различие между математическими объектами ("идеальными" числами) и их конечными моделями (машинными представлениями) приводит к вычислительным ошибкам типа ошибок округления.

2. Универсальные алгоритмы

Назовем алгоритм *универсальным*, если этот алгоритм не зависит от конкретного числового домена и/или от его конкретного машинного представления. Типичным примером универсального алгоритма является вычисление скалярного произведения (x, y) векторов $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_n)$ по формуле $(x, y) = x_1y_1 + \ldots + x_ny_n$. Этот алгоритм

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант №99–01–01198.

(формула) не зависит от конкретного домена и его машинного представления, так как формула имеет смысл для любого полукольца. Разумеется, один алгоритм может быть более универсальным, чем другой. Например, самый универсальный алгоритм численного интегрирования задается примитивной формулой прямоугольников; этот алгоритм работает даже для идемпотентного интегрирования (над любым идемпотентным полукольцом, см. [2]). Другие квадратурные формулы (например, составные формулы трапеций и Симпсона) не зависят от машинной арифметики и могут (например, в итерационной форме) использоваться для вычислений с произвольной точностью. Напротив, алгоритмы, основанные на квадратурных формулах Гаусса — Якоби, ориентированы на вычисления с фиксированной точностью и содержат константы (коэффициенты и узлы этих формул), задаваемые с фиксированной точностью. Разумеется, алгоритмы этого типа можно сделать более универсальными, включив в них процедуры вычисления констант, но это приведет к неоправданному усложнению алгоритмов.

Высокой степенью универсальности обладают алгоритмы компьютерной алгебры, используемые в таких системах аналитических вычислений, как МАТЕМАТИКА, Maple, REDUCE и др. Стандартные алгоритмы линейной алгебры можно переписать таким образом, чтобы они работали над любым полем и любым полным идемпотентным полукольцом (включая полукольца интервалов, см. [3], где рассматриваются интервальная версия идемпотентной линейной алгебры и соответствующие универсальные алгоритмы).

От машинного представления чисел, как правило, не зависят итерационные алгоритмы (начиная с обычного метода последовательных приближений), простые методы решений дифференциальных уравнений (например, методы Эйлера, Эйлера — Коши, Рунге — Кутта, Адамса, ряд вариантов метода разностных аппроксимаций и т. п.), методы вычисления элементарных и ряда специальных функций, основанные на разложениях в ряд Тейлора и в цепные дроби (аппроксимации Паде), и др.

3. Универсальные алгоритмы и точность вычислений

Для компьютерных вычислений, как правило, используют машинные арифметики с плавающей запятой и фиксированной длиной мантиссы, т. е. вычисления ведутся с фиксированной точностью. При этом фиксируется (грубо говоря) лишь относительная ошибка округления, что может приводить к катастрофической потере точности и получению заведомо ложных результатов (например, при суммировании рядов и вычитании близких чисел). Однако в этом случае удается добиться сравнительно высокой скорости вычислений. Ряд важных вычислительных алгоритмов ориентирован на вычисления в арифметике с плавающей запятой (фиксированной точности) и максимальную скорость вычислений. Эти алгоритмы не универсальны. Примерами являются квадратурные формулы Гаусса — Якоби (см. выше), алгоритмы вычисления элементарных и специальных функций с помощью наилучших полиномиальных и рациональных приближений или аппроксимаций Паде — Чебышева и т. п. Особенность этих алгоритмов — использование нетривиальных констант, заданных с фиксированной точностью.

Вместе с тем в последнее время вопросы точности, надежности и достоверности компьютерных вычислений (включая влияние ошибок округления) выдвигаются на передний план; отчасти это связано с постоянным ростом производительности средств вычислительной техники. В тех случаях, когда ошибки в исходных данных и ошибки округления оказывают сильное влияние на результат вычислений (некорректные задачи, исследование устойчивости решений и т. п.), часто целесообразно вести расчеты с повышенной и переменной точностью. Для этой цели можно, в частности, использовать арифметику рациональных чисел, в которой погрешность округления задается пользователем [4]. Такая арифметика является естественным дополнением к средствам интервального анализа, см. [5]. Соответствующие вычислительные алгоритмы должны быть универсальными (в том смысле, что не должны зависеть от машинного представления чисел).

4. Особенности программной реализации универсальных алгоритмов

Удобным средством программной реализации универсальных алгоритмов являются объектно ориентированные языки программирования (например, C^{++} и Java) и системы программирования, использующие технику абстрактных типов данных. В этом случае программные модули могут оперировать с абстрактными (и переменными) операциями и типами данных. При этом конкретные значения операций определяются типом входных данных; эти конкретные операции (и типы данных) генерируются дополнительными программными модулями.

С помощью такой техники на языке C^{++} написана система для решения задач линейной алгебры над полями и полукольцами (при различной машинной реализации соответствующих доменов) и оптимизационных задач на графах. В частности, можно использовать разные версии рациональной арифметики [4] и вести вычисления с произвольной заданной точностью; решая системы линейных уравнений Беллмана над идемпотентными полукольцами (см., например, [1–3]), можно решать стандартные оптимизационные задачи на графах (динамического программирования, о кратчайшем пути, о пути наибольшей ширины и т. п.), включая интервальные версии этих задач [3]. Подробное описание указанной системы будет дано в отдельной публикации.

Авторы выражают искреннюю благодарность А. Я. Родионову за ценные указания.

Список литературы

[1] Маслов В. П., Колокольцов В. Н. *Идемпотентный анализ и его применение в оптимальном управлении.* Наука, М., 1994.

[2] Litvinov G. L., Maslov V. P. The correspondence principle for idempotent calculus and some computer applications. In *"Idempotency. Publ. of the Newton Institute"*. Ed. J. Gunawardena. Cambridge University Press, 1998, 420–443.

[3] Литвинов Г. Л., Маслов В. П., Соболевский А. Н. Идемпотентная математика и интервальный анализ. *Вычисл. технологии*, 2000 (в печати).

[4] Литвинов Г. Л., Родионов А. Я., Чуркин А. В. Приближенная рациональная арифметика с контролируемыми ошибками округления. *Там же*, 2000 (в печати).

[5] Матиясевич Ю. В. Вещественные числа и ЭВМ. *Кибернетика и вычисл. техника*, вып. 2, 1986, 104–133.

Поступила в редакцию 14 марта 2000 г.