Разработка и исследование алгоритмов вычисления некоторых математических функций

Артамонов Сергей Максимович
9 класс, МАОУ Лицей № 36 Нижнего Новгорода, Научное объединение «Школа юного исследователя» АНО ДО «Академ клуб», ИПФ РАН
Научный руководитель О.А. Козина,
учитель математики МАОУ Лицей № 36

Бортовые вычислительные или связные приборы имеют жесткие требования к массе, габаритам, потребляемой мощности и условиям эксплуатации. Такая аппаратура реализуется на надежных, малопотребляющих микросхемах, набор математических операций которых часто включает в себя только сложение, вычитание и умножение. В работе синтезированы алгоритмы нахождения функций синус, корень и обратный корень, использующие только базовые арифметические операции и небольшие объемы памяти.

Данная работа относится к прикладной математике и методам вычисления функций с применением вычислительной техники. Вычислительные или связные приборы, используемые на борту летательного или космического аппарата, на борту геологоразведочного кунга, имеют жесткие требования к массе, габаритам, потребляемой мощности и условиям эксплуатации. В таких условиях элементная база ограничивается узким перечнем, состоящим из надежных, малопотребляющих микросхем. Часто набор математических операций этих микросхем включает в себя только сложение, вычитание и умножение. Кроме того, часто необходимо искать компромисс между быстродействием, занимаемой памятью, точностью вычислений.

В работе синтезированы алгоритмы нахождения функций синус, корень и обратный корень, использующие только базовые арифметические операции и небольшие объемы памяти.

Алгоритм нахождения функции $sin(x)$ на основе формул разложения тригонометрических функций в степенной ряд наиболее часто приводится в руководствах к сигнальным процессорам [1]. Описательная часть такого алгоритма содержит формулу разложения в степенной ряд:

$$sin(x) = x/1! - x^3/3! + x^5/5! - x^7/7! + \ldots \ (R = \infty) \qquad (1)$$

и код на языке программирования.

Несмотря на то, что ряд сходятся на интервале $(-\infty,\infty)$ [1], ввиду периодичности функции ее аргумент приводится в интервал $[-\pi,\pi)$. Разработанный же алгоритм приводит аргумент в область $[0,\pi/4)$. Это сделано с целью уменьшения количества суммируемых членов ряда при обеспечении заданной точности. Анализ степенного ряда $sin(x)$ показывает, что уменьшение x в 4 раза ведет к уменьшению первого члена ряда в 4 раза, второго члена ряда в 64 раза, третьего – в 1024 раза, четвертого – в 16384 раза и т.д. Можно показать и улучшение точности при суммировании заданного количества членов ряда. Например, при $x = 11\cdot\pi/12$ просуммировав первые три члена ряда получим приближение $sin(x)$ отличающееся от точного значения почти на 0.3, но приведя x в область $[0,\pi/4)$, т.е. к значению $\pi/12$, и просуммировав опять же первые три члена ряда мы получим приближение $sin(x)$ отличающееся от точного значения на $1.67\cdot10^{-8}$. Выигрыш в точности от приведения аргумента в данном примере – семь порядков. Но, приведение аргумента в область $[0,\pi/4)$ потребует организации нескольких условных переходов, выбора проведения вычислений по формуле разложения в ряд для $sin(x)$ или по формуле для ряда

$$cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + \ldots \ (R = \infty) \qquad (2)$$

и работу со знаком результата по окончании вычислений.

Для одинарной точности, возможно построение неитерационного табличного алгоритма вычисления $sin(x)$. Известны формулы синуса и косинуса суммы углов [2]:

$$sin(y+z) = sin(y)\cdot cos(z) + sin(z)\cdot cos(y) \qquad (3)$$

$$cos(y+z) = cos(y)\cdot cos(z) - sin(y)\cdot sin(z) \qquad (4)$$

Если представить градусную меру угла x в виде суммы $(y+z) \approx x$, то используя формулу синуса и косинуса суммы, мы можем интерполировать значения функции между отсчетами с "крупным" шагом с точностью, задаваемой таблицами с "мелким" шагом. Особенность задания таблиц такова, что таблицы с "мелким" шагом заданы на интервале одного "крупного" шага. Дальнейшее уточнение результата можно произвести по той же формуле синуса суммы, но в качестве одного из суммируемых углов взять приближение $(y+z)$, другого отклонение $x - (y+z)$, а синус и косинус отклонения оценивать первыми членами разложений в ряд.

Точность алгоритма, основанного на разложении в степенной ряд, можно установить на 12 порядков выше точности табличного алгоритма, но он требует в семь раз больше вычислительных операций (около 143 операций против 19). Объемы памяти, требуемые для заранее посчитанных данных, близки у алгоритмов.

Другие востребованные в бортовых устройствах функции – это корень и обратный корень. Для возведения числа в степень в большинстве математических библиотек языков программирования применяется формула вида $a^x = e^{x \cdot ln(a)}$, где правую часть выражают через степенной ряд. Однако, для поиска корня и обратного корня невысокой степени можно предложить другой способ. Для поиска нуля некоторой функции применяются методы оптимизации. Хорошо зарекомендовал себя итерационный метод Ньютона [3]:

$$x_{n+1} = x_n - F(x_n)/F'(x_n) \quad (5)$$

где x_n – значение аргумента функции на n-й итерации, $F(x_n)$ – значение функции на n-й итерации, $F'(x_n)$ – значение производной от функции на n-й итерации.

Для нахождения корня квадратного можно было бы предложить в качестве рабочей функции: $F(x) = x^2 - C$, где C – число, корень квадратный которого мы ищем. Очевидно, если $x = C^{1/2}$, то рабочая функция обращается в ноль. Метод Ньютона ведет к итерационной процедуре поиска x:

$$x_{n+1} = x_n - (x_n^2 - C)/(2x_n) = 0.5 \cdot (x_n + C/x_n) \quad (6)$$

содержащей операцию деления, а деление очень накладная операция и требует не один такт на себя. С другой стороны, если найден обратный квадратный корень, то сам квадратный корень находится как произведение обратного квадратного корня от числа на само это число:

$$x^{1/2} = x \cdot (x^{-1/2}) \quad (7)$$

Рабочая функция для нахождения обратного квадратного корня [4]:

$$F(x) = 1/(x^2) - C \quad (8)$$

приводит к итерационной формуле:

$$x_{n+1} = x_n - F(x_n)/F'(x_n) = 0.5\, x_n\, (3 - Cx_n^2) \quad (9)$$

не содержащей делений.

Несмотря на востребованность операции корня k-й степени, в литературе обобщения подобных итераций для высших степеней встречено не было. Рабочая функция для нахождения обратного корня k-й степени:

$$F(x) = 1/(x^k) - C \quad (10)$$

приводит к итерациям:

$$x_{n+1} = x_n - F(x_n)/F'(x_n) = (1/k)\, x_n\, (k + 1 - Cx_n^k) \quad (11)$$

в которых $(1/k)$ для разных k может быть заранее сохранено в массив.

Положительное число X, хранящееся в памяти процессора, представляется в виде [5]:

$$X = (1 + Mx/L) \cdot 2^{(Ex - B)} \quad (12)$$

где $L = 2^{52}$, $B = 1023$ для двойной точности; $L = 2^{23}$, $B = 127$ для одинарной точности.

Пусть $Y = X^{-1/k}$, с другой стороны:

$$Y = (1 + My/L) \cdot 2^{(Ey - B)} \quad (13)$$

Тогда приравняв обратный корень k-й степени от X и Y, а затем от левой и правой части равенства взяв двоичный логарифм, получим:

$$log_2(1+My/L) + Ey - B = (-1/k) \cdot (log_2(1+Mx/L) + Ex - B) \quad (14)$$

Воспользуемся аппроксимацией:

$$log_2(1+z) \approx z + q \quad (15)$$

Параметр q аппроксимирующей функции на интервале z э $[0;1]$

определим из критерия минимума модуля максимальной ошибки аппроксимации.

В результате приходим к формуле для первого приближения:

$$My + L \cdot Ey \approx ((k+1)/k) \cdot L \cdot (B - q) - (1/k) \cdot (Mx + L \cdot Ex) \quad (16)$$

Первое слагаемое левой части просчитывается заранее, а второе слагаемое содержит целочисленное деление. Но в данном случае оно может быт реализовано через произведение и сдвиг. Если значение обратного корня k-й степени найдено, то значение корня k-й степени находится следующим образом

$$x^{1/k} = x \cdot (x^{-1/k})^{k-1} \quad (17)$$

В ходе работы были изучены возможности численной реализации алгоритмов вычисления тригонометрических функций $sin(x)$ и $cos(x)$, функции вычисления корня и обратного корня. Разработаны пригодные для реализации в современных вычислительных системах алгоритмы вычисления $sin(x)$, корня k-й степени, обратного корня k-й степени. Для разработки алгоритма вычисления корня теоретически обоснован выбор константы в аппроксимирующем функцию $log_2(1+z)$ выражении $z+q$. Проведен сравнительный анализ двух подходов к разработке алгоритма вычисления функции $sin(x)$. Выполнена проверка точности алгоритмов. В зависимости от примененной в аппаратуре элементной базы алгоритмы могут быть адаптированы под соответствующие требования.

Разработанные алгоритмы вычисления синуса могут быть использованы в формирователях связных сигналов, преобразователях частоты, корреляторах и других устройствах цифровой обработки сигналов. Корень квадратный используется при амплитудной демодуляции. Обратный квадратный корень используется в устройствах нормировки мощности. Корень пятой, например, степени может быть использован для компрессии сигнала.

1. М.Я. Выгодский. *Справочник по высшей математике*, 2006, М.: АСТ: Астель.
2. В.В. Зайцев, В.В. Рыжков, М.И. Сканави. *Элементарная математика*, 1967, М.: Наука.
3. В.В. Лесин, Ю.П. Лисовец. *Основы методов оптимизации*, 1998, М.: Изд-во МАИ.
4. *ADSP-21000 Family Application Handbook*, 1994, Analog Devices Inc.
5. Д. Каханер, К. Моулер, С. Нэш. *Численные методы и математическое обеспечение*, 1998, М.: Мир.