

Ошибка аппроксимации функции \sqrt{x} .

Математика была ограничена 32-битными числами, (включая умножение $[R32L, R32H] \leq R32 * R32$). Изначально, как и все решил использовать кусочно-линейную аппроксимацию при вычислении корня квадратного из X.

Затем дополнил кусочно-линейную аппроксимацию, разложением ошибки в ряд подобный ряду Фурье, до 8-мого члена ряда. Это быстро, но на 32-битом числе дало максимальную ошибку результата порядка 4-х. Кусочно-линейная аппроксимация, давала ошибку порядка 800.

Аппроксимировать ошибку при помощи разложения в ряд подобный Фурье, достаточно дорого с точки зрения вычислительных ресурсов. По этому, использовал квадратичную аппроксимацию. Квадратичная аппроксимация, тоже обладает ошибкой, поскольку максимумы ошибки линейной аппроксимации не совпадают с серединой интервала. Я добавил коррекцию линейного коэффициента аргумента (x). Это,- фактически привело к увеличению степени аппроксимирующего полинома до третьей. По хорошему, следует смотреть на коррекцию с двух сторон, это может еще увеличить точность. Но я ограничился упрощенной версией. Расчеты проведены в SMathStudio, написан код на Assembler для ARM Cortex M3. Это не итеративное вычисление начального значения корня. Далее, можно провести уточнение используя стандартные методы.

<i>Количество бит (аргумента)</i>	<i>Значение ошибки (результата)</i>	<i>Количество бит (аргумента)</i>	<i>Значение ошибки (результата)</i>
1	0.000000	17	0.058273
2	0.000000	18	0.082397
3	0.000000	19	0.116546
4	0.000000	20	0.164901
5	0.011535	21	0.233276
6	0.012161	22	0.329895
7	0.012161	23	0.466552
8	0.012161	24	0.659805
9	0.012161	25	0.933120
10	0.012161	26	1.319641
11	0.012161	27	1.866271
12	0.012161	28	2.639312
13	0.014556	29	3.732498
14	0.020584	30	5.278549
15	0.029144	31	7.465072
16	0.041198	32	10.55725

Таблица 1: Абсолютная максимальная ошибка результата.

Соболев Е.В.

Тел.: +79003030374

e-mail: hwsdevsev@gmail.com

Добавил быструю итеративную коррекцию результата. Поскольку погрешность аппроксимации заранее известна для каждого диапазона, то использовал это значение для уточнения результата методом деления отрезка пополам. При этом используются операции умножения и вычитания (сравнения), сложение, операция сдвига. **Операция деления в алгоритме по прежнему не используется.**

Не стоит забывать, что результат ограничен 16-ю битами целой части и 16-ю битами дробной части. Поскольку, 16 битное значение дробной части ограничивает диапазон значений, это приводит к некоторой ошибке.

Данный квадратный корень можно рассматривать как квадратный корень из старших 32-бит 64-битного числа, младшая часть которого равна нулю. Или как квадратный корень из 16 битного числа, при этом младшие 16-бит результата представляют из себя дробное значение. Таким образом, работая с 32-битными числами, фактически, мы работаем с 64-битными с некоторой точностью.

Интересно следующее.

sqrtFixed(1) = 0x1000, что эквивалентно 1
sqrtFixed(2) = 0x16a09, что эквивалентно 1.4141998.
sqrtFixed(4) = 0x20000, что эквивалентно 2.
sqrtFixed(10000) = 0x640000, что эквивалентно 100.

Т.е. для получения целой части достаточно сдвинуть результат на 16 бит вправо. Какова цель? При работе с ПО станка CNC, без использования Floating-point можно получить квадратный корень числа с точностью порядка 1LSB.

Соболев Е.В.

Тел.: +79003030374

e-mail: hwsdevsev@gmail.com