Quotient approximation for schoolbook division

Niels Möller

August 5, 2018

Abstract

A new more efficient way to compute an approximative quotient suitable for school book division of multiprecision integers.

1 Background

This section gives a short overview of schoolbook division history, as it been applied to the GMP [1] library.

Knuth, 1969

The classic description of schoolbook division is Knuth's, see [3, Sec. 4.3.1, Alg. D]. It works as follows.

To compute the most significant quotient word, start by dividing the two most significant words of the numerator with the most significant word of the divisor (the later is assumed normalized, i.e., most significant bit set). Next, take one more word into account of both numerator and divisor, to check if the approximation is a correct quotient for dividing the three most significant numerator words with the two most significant. If it isn't, it's at most two units too large, and is adjusted accordingly.

After these preliminary adjustments, the quotient word is usually correct, with a small probability of it being one too large. So go ahead and compute the full multiprecision remainder; check for the unlikely underflow, to do a final adjustment of both the quotient word and the multiprecision remainder when this happens.

Granlund-Montgomery, 1994

When computing one quotient word at a time, to produce a multiprecision quotient, the numerator is updated incrementally by subtracting multiples of the divisor, but the divisor itself is unchanged; it is a loop invariant. One can therefore speed up the computation of the quotient approximation by precomputing an approximate reciprocal of the most significant divisor word. The initial divison is then replaced by a few multiplications and adjustments, which is a big win since division instructions are usually vastly slower than multiplication instructions. See [2] for one clever way to do that.

Möller-Granlund, 2011

The main idea of this paper, when applied to schoolbook division, is to use a reciprocal based on the two most sigificant divisor limbs. The reciprocal is still a single word, but we can simplify the adjustment steps needed for each quotient word, by using a slightly different reciprocal. In effect, moving some of the adjustment work out of the loop and doing it as part of the precomputation of the reciprocal. See [4].

This algorithm divides the three most significant words of the numerator with the two most significant words of the divisor, producing the same candidate quotient as used in Knuth, but with simpler adjustment steps than earlier methods.

Current work, 2018

The algorithms above all produce a correct quotient of three words by two. When this is ready, it is applied to compute the full multiprecision remainder. We need a final adjustment in the unlikely case that the computation of the full remainder underflows.

Further improvement is based on two observations. First, since we do have a final adjustment step, we don't need a three-by-two quotient that is correct in all cases. Second, the influence on the correct quotient from the third most significant word of the numerator is very small.

We therefore aim to compute a candidate quotient based on the two most significant words of both numerator and divisor. When applied to compute the multiprecision remainder, it must be either be correct or one too large, and the probability of error should be small. The resulting algorithm features simpler adjustment steps than the earlier methods.

2 Notation and requirements

Let ℓ denote the computer word size, and let $\beta = 2^\ell$ denote the base implied by the word size. Lower-case letters denote single-word numbers, and upper-case letters represent numbers of any size. We use the notation $X = \langle x_{n-1}, \ldots, x_1, x_0 \rangle = x_{n-1}\beta^{n-1} + \cdots + x_1\beta + x_0$, where the n-word integer X is represented by the words x_i, for $0 \le i < n$.

We consider only one iteration of the schoolbook division algorithm, computing a single quotient word; organizing the outer loop is out of scope for these notes. Let the divisor $D = \langle d_{n-1}, \ldots d_0 \rangle$ consist of $n > 2$ words, and the numerator $U = \langle u_n, \ldots n_0 \rangle$ consist of $n + 1$ words.

We assume that $U < \beta D$, so that the correct quotient $\lfloor U/D \rfloor$ is a single word, and that $d_{n-1} \ge \beta/2$ (normalization).

We need a function `divappr`, that lets us compute a candidate quotient $q \leftarrow \texttt{divappr}(u_n, u_{n-1}, d_{n-1}, d_{n-2})$. To ensure that a single adjustment step is sufficient for correctness, q must satisfy

$$-D \le U - qD < D$$

To get good performance, with the final adjustment step being rare, we want $U - qD < 0$ to be unlikely. If the remainder $U \mod D$ can be assumed uniformely

random, we can achieve this by improving the lower bound from $-D$ to $-\epsilon D$ for some small epsilon.

3 The `divappr` function

We define a function $q \leftarrow \texttt{divappr}\langle u_1, u_0\rangle, \langle d_1, d_0\rangle$. Inputs consists of four single-word numbers. We require that $d_1 \geq \beta/2$ and $\langle u_1, u_0\rangle \leq d_1, d_0$. The output Let R' denote the remainder

$$R' = \langle u_1, u_0, 0\rangle - q\langle d_1, d_0\rangle$$

The output q is also a single word. In the borderline case $\langle u_1, u_0\rangle = \langle d_1, d_0 0\rangle$, `divappr` must produce $q = \beta - 1$. This correesponds to $R' = \langle d_1, d_0\rangle$, and in the context of schoolbook divison, $q = \beta - 1$ is the correct quotient, thanks to the requirement that $U < \beta D$.

When $\langle u_1, u_0\rangle < \langle d_1, d_0\rangle$, we require R' to belong to the range

$$-2\beta \leq R' < \langle d_1, d_0\rangle$$

When q is applied to the full multiprecision numbers, the corresponding remainder

$$R = U - qD$$

satisfies $R < D$ and

$$R > -3\beta^{n-1} \geq -\frac{6}{\beta}D$$

ensuring that $R < 0$ is unlikely for random inputs.

To compute `divappr`, we will make use of the same approximate reciprocal as for three-by-two division, defined as

$$v = \lfloor(\beta^3 - 1)/\langle d_1, d_0\rangle\rfloor - \beta$$

4 The algorithm

$q \leftarrow$ DIVAPPR2$(\langle u_1, u_0\rangle, \langle d_1, d_0\rangle, v)$
    In: $\beta/2 \leq d_1 < \beta$, $\langle u_1, u_1\rangle \leq \langle d_1, d_0 0\rangle$,
        $v = \lfloor(\beta^3 - 1)/\langle d_1, d_0\rangle\rfloor - \beta$
1 **if** $\langle u_1, u_0\rangle \geq \langle d_1, d_0\rangle - d_1$
2   **return** $\beta - 1$
3 $\langle q_1, q_0\rangle \leftarrow vu_2 + \langle u_2, u_1\rangle$
4 $q \leftarrow q_1 + 1$
5 $\langle p_1, p_0\rangle \leftarrow qd_0$
6 $r \leftarrow (u_0 - qd_1 - p_1 - [p_0 > 0]) \bmod \beta$
7 **if** $r \geq q_0$
8      $q \leftarrow (q - 1) \bmod \beta$
9      $r \leftarrow (r + d_1 + 1) \bmod \beta$
10 **if** $r \geq d_1 - 1$
11     $q \leftarrow (q + 1) \bmod \beta$
12 **return** q

The expression $[p_0 > 0]$ on line 6 denotes a conditional expression, with the value one if $p_0 > 0$, otherwise zero. We need to prove that the algorithm produces the desired result in all cases.

Since

$$\langle d_1, d_0\rangle(\beta - 1) = \beta^2 d_1 + \beta(d_1 - d_0) - d_0$$

we have $\lfloor\langle u_1, u_0, 0\rangle/\langle d_1, d_0\rangle\rfloor \geq \beta - 1$ if and only if $\langle u_1, u_0\rangle \geq \langle d_1, d_0\rangle - d_1$. This is the condition on line 1, and it follows that we return $q = \beta - 1$ for all inputes where it's the correct quotient, and in the borderline case $\langle u_1, u_0\rangle = \langle d_1, d_0\rangle$.

So let us assume that $\langle u_1, u_0\rangle < \langle d_1, d_0\rangle - d_1$; then the correct quotient is at most $\beta - 2$. This ensures that in the cases that we return a quotient which is one too large, that incorrect quotient still fits in one word.

The value q_1 is always upper bounded by the correct quotient (since the reciprocal v is rounded down), hence the the initial quotient candidate quotient, computed on line 4, also fits in a single word.

Define

$$\begin{aligned} R' &= \langle u_1, u_0, 0\rangle - (q_1 + 1)\langle d_1, d_0\rangle \\ &= \beta\left[\langle u_1, u_0\rangle - (q_1 + 1)d_1 - p_1\right] - p_0 \end{aligned}$$

The computation on line 6 produces the middle word of R',

$$\begin{aligned} r_1 &= \lfloor(R' \bmod \beta^2)/\beta\rfloor \\ &= u_1 - (q_1 + 1)d_1 - p_1 - [p_0 > 0] \bmod \beta \end{aligned}$$

The last terms represents underflow from the low word, $0 - p_0$. Also let $r_0 = -p_0 \bmod \beta$ denote the low word, which intentionally isn't used by the algorithm.

To analyze the situation, turn to the analysis of three-by-two division in [4, Theorem 3]. We can tighten the upper bound slightly, since the proof in the paper adds β to bound the contribution to the remainder from the least significant numerator word, which in our setting is always zero. Hence, we have

$$\max(\beta^2 - \langle d_1, d_0\rangle, q_0\beta) - \beta^2 \leq R' < \max(\beta^2 - \langle d_1, d_0\rangle, q_0\beta) - \beta$$

We treat the two possible signs of R' separately.

Assume $R' < 0$

If $R' < 0$, then $R' = \langle r_1, r_0\rangle - \beta^2$, and the lower bound implies

$$\langle r_1, r_0\rangle = R' + \beta^2 \geq q_0\beta$$

Hence, $r_1 \geq q_0$, and so the first adjustment condition applies. The other half of the lower bound implies

$$\langle r_1, r_0\rangle = R' + \beta^2 \geq \beta^2 - \langle d_1, d_0\rangle$$

It follows that

$$r_1 + d_1 + 1 > \beta$$

Let r_1' denote the value after adjustment on line 8, it's

$$r_1' = r_1 + d_1 + 1 \bmod \beta = r_1 + d_1 + 1 - \beta$$

The corresponding two-word remainder is

$$\begin{aligned}\langle u_1, u_0, 0\rangle - q_1\langle d_1, d_0\rangle &= R' + \langle d_1, d_o\rangle \\ &= \langle r_1, r_0\rangle - \beta^2 + \langle d_1, d_o\rangle \\ &= \beta(r_1 + d_1 - \beta) + r_0 + d_0 \\ &= \beta(r_1 + d_1 + 1 - \beta) - \beta + r_0 + d_0 \\ &= \beta r_1' + (r_0 - \beta) + d_0 < \beta r_1 + d_0\end{aligned}$$

If $r_1' \leq d_1 - 2$, then this is the final remainder R, and it follows that $R < \beta(d_1 - 2) + d_0 < \langle d_1, d_0\rangle$. On the other hand, if $r_1' \geq d_1 - 1$, then the final remainder is $R = R'$, and

$$\begin{aligned}R = R' &= \beta r_1' + (r_0 - \beta) + d_0 - \langle d_1, d_0\rangle \\ &\geq \beta(d_1 - 1 - d_1) + (r_0 - \beta) \\ &\geq -2\beta\end{aligned}$$

Assume $R' \geq 0$

If $R' \geq 0$, then $R' = \langle r_1, r_0\rangle$. If the first adjustment step isn't done, then the second adjustment condition would produce a final remainder in the range

$$-2\beta < R < \langle d_1, 0\rangle \leq \langle d_1, d_0\rangle$$

But what happens if $r_1 \geq q_0$? We then have the upper bound

$$\langle r_1, r_0\rangle = R' < \beta^2 - \langle d_1, d_0\rangle - \beta = \beta(\beta - d_1 - 1) - d_0$$

It follows that $r_1 < \beta - d_1 - 1$, Hence, the value after the update is

$$r_1 + d_1 + 1 \bmod \beta = r_1 + d_1 + 1 \geq d_1 - 1$$

so we get two adjustments canceling out. Furthermore, since we require $d_1 \geq \beta/2$, we have

$$R = R' < \beta^2 - \langle d_1, d_0\rangle \leq \beta^2/2 \leq \langle d_1, d_0\rangle$$

and $R < \langle d_1, d_0\rangle$, as desired.

References

[1] Torbjörn Granlund. GNU multiple precision arithmetic library. `http://gmplib.org/`.

[2] Torbjörn Granlund and Peter L. Montgomery. Division by invariant integers using multiplication. In *Proceedings of the SIGPLAN PLDI'94 Conference*, June 1994.

[3] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, third edition, 1998.

[4] Niels Möller and Torbjörn Granlund. Improved division by invariant integers. *IEEE Transactions on Computers*, 60:165–175, 2011.