Quadratic Short Division

Juraj Sukop, Paul Zimmermann

Quadratic Short Division

Juraj Sukop Paul Zimmermann*

April 30, 2024

Abstract

In Modern Computer Arithmetic [1], the authors describe a quadratic division with remainder, and mention a factor of two speedup can be obtained when only the quotient is needed. We give an explicit quadratic algorithm that computes an approximate quotient.

We first recall the quadratic division with remainder algorithm from [1]:

Algorithm 1 BasecaseDivRem

Input: $A = \sum_0^{n+m-1} a_i\beta^i$, $B = \sum_0^{n-1} b_j\beta^j$, B normalized, $m \geq 0$
Output: quotient Q and remainder R of A divided by B

if $A \geq \beta^m B$ **then** $q_m \leftarrow 1$, $A \leftarrow A - \beta^m B$ **else** $q_m \leftarrow 0$
for j **from** $m-1$ **downto** 0 **do**
  $q_j^* \leftarrow \lfloor (a_{n+j}\beta + a_{n+j-1})/b_{n-1} \rfloor$ ▷ quotient selection
  $q_j \leftarrow \min(q_j^*, \beta - 1)$
  $A \leftarrow A - q_j\beta^j B$
  while $A < 0$ **do**
    $q_j \leftarrow q_j - 1$
    $A \leftarrow A + \beta^j B$
return $Q = \sum_0^m q_j\beta^j$, $R = A$.

We propose the following algorithm to compute an approximate quotient:

Algorithm 2 BasecaseShortDiv

Input: $A = \sum_0^{n+m-1} a_i\beta^i$, $B = \sum_0^{n-1} b_j\beta^j$, B normalized, $m \geq 0$, $n \geq 1$
Output: approximate quotient Q of A divided by B

if $A \geq \beta^m B$ **then** $q_m \leftarrow 1$, $A \leftarrow A - \beta^m B$ **else** $q_m \leftarrow 0$
for j **from** $m-1$ **downto** 0 **do**
  write $\beta^j B = H_j\beta^{n-1} + L_j$ with $0 \leq L_j < \beta^{n-1}$
  $q_j^* \leftarrow \lfloor (a_{n+j}\beta + a_{n+j-1})/b_{n-1} \rfloor$ ▷ quotient selection
  $q_j \leftarrow \min(q_j^*, \beta - 1)$
  $A \leftarrow A - q_j H_j \beta^{n-1}$
  while $A < 0$ **do**
    $q_j \leftarrow q_j - 1$
    $A \leftarrow A + H_j\beta^{n-1}$
  if $A \geq \beta^j B$ **then** $q_j, \ldots, q_0 \leftarrow \beta - 1$; break
return $Q = \sum_0^m q_j\beta^j$

*Université de Lorraine, CNRS, Inria, LORIA

Theorem 1 *The approximate quotient Q returned by Algorithm BasecaseShortDiv satisfies:*

$$-1 < Q - A/B < 2\min(m, n-1).$$

Proof: We first prove that the value A at the beginning of the for-loop with index j satisfies $A < \beta^{j+1}B$. If we have $A \geq \beta^m B$ initially, then since $B \geq \beta^n/2$, $2\beta^m B \geq \beta^{n+m} > A$ thus the new value of A is $A - \beta^m B < \beta^m B$, and $A < \beta^{j+1}B$ holds for $j = m-1$. Since when $A \geq \beta^j B$, we exit the for-loop at step 10, when $A < \beta^j B$ we have $A < \beta^{j+1}B$ at the next iteration.

Let j_0 be the last index for which the for-loop is processed. Let A_{j+1} (resp. A_j) be the value of A at the beginning (resp. end) of the loop for index j, and q_j the value after the potential corrections at step 9. We thus have $A_j = A_{j+1} - q_j H_j \beta^{n-1}$. Since $\beta^j B = H_j \beta^{n-1} + L_j$, it follows:

$$A_j = A_{j+1} - q_j(\beta^j B - L_j),$$

and thus:

$$A_{j_0} = A - QB + \sum_{j=j_0}^{m-1} q_j L_j. \tag{1}$$

If $A \geq \beta^j B$ and we exit the for-loop early, since $\beta^j B \geq H_j \beta^{n-1}$, we could not have entered the while-loop, thus we already had $A \geq \beta^j B$ before the while-loop. Assume $q_j = q_j^*$, and write $a_{n+j}\beta + a_{n+j-1} = q_j b_{n-1} + r$ with $0 \leq r < b_{n-1}$. Since H_j has the same $j+1$ upper words as $\beta^j B$, we can write $H_j\beta^{n-1} = b_{n-1}\beta^{n+j-1} + S$ with $0 \leq S < \beta^{n+j-1}$. It follows:

$$\begin{aligned} A - q_j H_j \beta^{n-1} &= (q_j b_{n-1} + r)\beta^{n+j-1} + R - q_j(b_{n-1}\beta^{n+j-1} + S) \\ &\leq r\beta^{n+j-1} + R - q_j S \\ &< (b_{n-1} - 1)\beta^{n+j-1} + \beta^{n+j-1} = b_{n-1}\beta^{n+j-1} \leq \beta^j B. \end{aligned} \tag{2}$$

It would contradict $A \geq \beta^j B$ before the while-loop.

Thus if we exit the for-loop early, necessarily we had $A \geq \beta^j B$ before the while-loop, and $q_j^* \geq \beta$, thus $q_j = \beta - 1$. Then since $A - q_j H_j \beta^{n-1} \geq \beta^j B$:

$$\begin{aligned} A &\geq (\beta-1)H_j\beta^{n-1} + \beta^j B \\ &= (\beta-1)(\beta^j B - L_j) + \beta^j B \\ &= \beta^{j+1}B - (\beta-1)L_j > \beta^{j+1}B - \beta^n \geq \beta^{j+1}B - 2B. \end{aligned}$$

Since we had $A < \beta^{j+1}B$ at the beginning of the loop, we deduce $\beta^{j+1}B - 2B < A < \beta^{j+1}B$, which can be written $A_{j_0} = A_{j_0+1} - (\beta^{j_0+1} - 1)B + T$, where we define $A_{j_0} = B - 1$ and $-1 < T < 2B - 1$. This matches Eq. (1), with the term $q_{j_0}L_{j_0}$ replaced by T, where both terms share the same bounds (T is an integer): $0 \leq q_{j_0}L_{j_0}, T < 2B$.

Now assume we don't exit the for-loop early. Since for $j = 0$ the condition $A \geq \beta^j B$ is not fulfilled, we have $A_0 < B$. Now for $j \geq n-1$, we have $\beta^j B = (\beta^{j-(n-1)}B)\beta^{n-1}$, thus $H_j = \beta^{j-(n-1)}B$ and $L_j = 0$. The term $q_j L_j$ is thus zero for $j \geq n-1$. Since we proved $0 \leq A_0 < B$, we have:

$$-2\min(m, n-1) \leq -\min(m, n-1)\beta\beta^{n-1}/B < A/B - Q < 1.$$

■

REMARK 1: since H_j has $j+1$ words, the multiplication $q_j H_j \beta^{n-1}$ at step 6 reduces to a multiplication of $j+1$ words by 1 word, instead of n words by 1 word in BasecaseDivRem, which yields a potential factor of two speedup. Similary at step 9, the addition $A + H_j\beta^{n-1}$ only involves $j+1$ words.

REMARK 2: for the integer quotient $\lfloor A/B \rfloor$, Theorem 1 yields $A/B - 1 < Q < A/B + 2\min(m, n-1)$ thus:

$$\lfloor A/B \rfloor \leq Q \leq \lfloor A/B \rfloor + 2\min(m, n-1) - 1.$$

Algorithm BasecaseShortDiv can thus be used to replace Algorithm Div in [2], since Div has $m = n$, and expects an approximate quotient Q satisfying $\lfloor A/B \rfloor \le Q \le \lfloor A/B \rfloor + 2n$.

REMARK 3: if one wants to reduce the upper bound $2\min(m, n-1)$, one can take one more word in H_j, say $\beta^j B = H_j \beta^{n-2} + L_j$ with $0 \le L_j < \beta^{n-2}$. The upper bound will become $2\min(m, n-1)/\beta$, which in most usual cases will be less than 1. This will however induce more operations at steps 6 and 9.

REMARK 4: the number of iterations of the while-loop at steps 7-9 is at most 2. Indeed, from Eq. (2) we have

$$\begin{aligned} A - q_j H_j \beta^{n-1} + 2H_j\beta^{n-1} &\ge -q_j S + 2H_j\beta^{n-1} \ge -\beta S + 2(b_{n-1}\beta^{n+j-1} + S) \\ &\ge -(\beta-2)S + 2b_{n-1}\beta^{n+j-1} \ge -(\beta-2)\beta^{n+j-1} + \beta\cdot\beta^{n+j-1} \ge 0. \end{aligned}$$

REMARK 5: the average number of corrections in the while-loop (per value of j) is about 0.35 for $m = n = 100$ (same value for BasecaseDivRem). To decrease the number of corrections, one should compute a more accurate partial quotient q_j by dividing the upper 3 words from A by the upper 2 words from $H_j\beta^{n-1}$:

$$q_j^* \leftarrow \lfloor (a_{n+j}\beta^2 + a_{n+j-1}\beta + a_{n+j-2})/(b_{n-1}\beta + b'_{n-2}) \rfloor,$$

where b'_{n-2} is the second most significant word of H_j. This is what function `mpfr_divhigh_n_basecase` from GNU MPFR does, for which corrections are extremely rare.

REMARK 6: the condition $A \ge \beta^j B$ at step 10 can be replaced by $A \ge \beta^{n+j}$, which is easier to check. Indeed, if $\beta^j B \le A < \beta^{n+j}$, the next iteration will get $q_{j-1}^* \ge \beta$, thus $q_{j-1} = \beta - 1$, and

$$A - q_{j-1}H_{j-1}\beta^{n-1} \ge \beta^j B - (\beta-1)\beta^{j-1}B = \beta^{j-1}B,$$

thus the condition for early termination will hold at the next iteration, and so on until $j = 0$. Then all subsequent values of q_j will be $\beta - 1$, which is exactly what the early exit code does. (The invariant $A < \beta^{j+1}B$ is no longer satisfied, but this does not matter in this case.)

References

[1] BRENT, R. P., AND ZIMMERMANN, P. *Modern Computer Arithmetic*. No. 18 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2010. Electronic version at `http://www.loria.fr/~zimmerma/mca/pub226.html`.

[2] HARVEY, D., AND ZIMMERMANN, P. Short Division of Long Integers. In *20th IEEE Symposium on Computer Arithmetic (ARITH-20)* (Tuebingen, Germany, July 2011), E. Antelo, D. Hough, and P. Ienne, Eds., IEEE, pp. 7–14.