Some Toom-Cook Methods For Even Long Integers

Alberto Zanoni

Centro "Vito Volterra" – Università di Roma "Tor Vergata"
Via Columbia 2 – 00133 Roma, Italy
zanoni@volterra.uniroma2.it

Abstract. We present a new approach to evaluation and interpolation phases of some (balanced and unbalanced) Toom-Cook multiplication methods for long integers when at least one factor is even. Some other optimizations are also indicated.

AMS Subject Classification: 11A05, 11A25, 11K65, 11Y70

Keywords and phrases: Polynomial, long integers, multiplication, Karatsuba, Toom, interpolation

1 Introduction

We describe an *ad hoc* approach to evaluation and interpolation phases of some Toom-Cook multiplication methods for long integers, when at least one factor is even. In particular, the classical Toom-3 method (with balanced and unbalanced factors) and the Toom-3.5 method (with slightly and very unbalanced factors) are presented. General possible optimizations of Toom-4.5 and Toom-5 are also shown.

2 Toom-3 classical method

Long integer multiplications is standardly reduced to polynomial multiplications by considering as coefficients a_i, b_i the digits of a certain base B expansion of the factors (for computer applications, typically $B = 2^{32k}$ for some k). The classical Toom-Cook method [5], [3] – Toom-3 for short – applies when B is such that the obtained polynomials have both degree 2 (balanced case) or 3 and 1, respectively (unbalanced case). Coefficients multiplication is similarly treated by recursion, up to a certain threshold when Karatsuba or high school multiplication methods are more effective.

2.1 Complexity issues

We note that if a_i, b_i length (number of bits in base 2 expansion) in evaluation phase is n, in interpolation phase the coefficients length is about $2n$. In order to analyze complexity, we consider the possible availability of an *ad hoc* function: consider, for $e \in \mathbb{Z}$, the following two equivalent processes (I) and (II)

$$(\text{I}) \begin{array}{l} \mathbf{T} \leftarrow 2^e \mathbf{X}; \\ \mathbf{Z} \leftarrow \mathbf{Y} \pm \mathbf{T}; \end{array} \qquad ; \qquad (\text{II}) \;\; \mathbf{Z} \leftarrow \mathbf{Y} \pm 2^e \mathbf{X};$$

for which one could write a shift-add function $sa(\mathbf{X}, \mathbf{Y}, e, op) \mapsto (\mathbf{Y} + sign(op) 2^e \mathbf{X})$ (with $op \in \{-1, 1\}$) performing (II) process, which reads $\mathbf{X}$, $\mathbf{Y}$ just once and uses no temporary variable, taking benefit of code locality. As memory access is much more time consuming than computing, the less we read/write data, the better it is.

Operation		Time	Operation		Time
Sign change	unitary $-$	~ 0	Shift	$\ll$, $\gg$	S
Addition	$+$	A	Division by k	$/$	$D_{(k)}$
Subtraction	$-$	A	Shift-add	sa	$A +$ _1_2

A reasonable computational model to analyze the complexity of interpolation was proposed by Bodrato and Zanoni in [2] – where unbalanced Toom–$(n+1/2)$ methods are also introduced – considering operations costs represented by the constants reported in the aside table, referring to the execution time of additions/subtractions, shifts (multiplications/divisions by power of 2), exact divisions by small constants and sa, whose real values depend on the particular chosen architecture. The indicated costs are relative to the operands (maximum) length: as all operations are linear, costs are proportional to it. This means for example that there is a multiplicative factor 2 distinguishing evaluation and interpolation costs, as operands in interpolation phase have more or less double length with respect to the ones in the evaluation phase.

Obviously we have _1_2 $\leqslant S$. If sa is not available, process (I) must be used, so that _1_2 $= S$ and one more temporary variable is possibly needed. As the shift operation must read only one operand in memory instead of the two ones needed by additions/subtractions, we reasonably also suppose that $S \leqslant A$.

2.2 Method description

The balanced version of Toom-3 method considers two quadratic polynomials:

$$a(x) = a_2x^2 + a_1x + a_0 \qquad ; \qquad b(x) = b_2x^2 + b_1x + b_0$$

For the unbalanced version (when $a(x)$ and $b(x)$ have different degrees) we instead have

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \qquad ; \qquad b(x) = b_1x + b_0$$

To find the coefficients of their product $c(x) = a(x)b(x)$ by using the EMI scheme, we first consider the values w_i obtained by evaluating $a(x)b(x)$ in the five interpolation points $\{\infty, 2, -1, 1, 0\}$, as shown below:

$$\begin{array}{c|c|cccc}
 & \textbf{Balanced} & \textbf{Unbalanced} & & & \\
M_1 & a_2b_2 & a_3b_1 & = w_4 = & c_4 \\
M_2 & (4a_2+2a_1+a_0)(4b_2+2b_1+b_0) & (8a_3+4a_2+2a_1+a_0)(2b_1+b_0) & = w_3 = & 16c_4+8c_3+4c_2+2c_1+c_0 \\
M_3 & (a_2-a_1+a_0)(b_2-b_1+b_0) & (a_3-a_2+a_1-a_0)(b_1-b_0) & = w_2 = & c_4-c_3+c_2-c_1+c_0 \\
M_4 & (a_2+a_1+a_0)(b_2+b_1+b_0) & (a_3+a_2+a_1+a_0)(b_1+b_0) & = w_1 = & c_4+c_3+c_2+c_1+c_0 \\
M_5 & a_0b_0 & a_0b_0 & = w_0 = & c_0
\end{array}$$

In both cases we have

$$w = Mc \qquad \text{with} \qquad M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 16 & 8 & 4 & 2 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 12 \end{pmatrix} \; ; \; w = \begin{pmatrix} w_4 \\ w_3 \\ w_2 \\ w_1 \\ w_0 \end{pmatrix} \; ; \; c = \begin{pmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix}$$

so that $c = M^{-1}w$ (M_i corresponds to the i^{th} line of the matrix).

In [1] Bodrato found the aside shown evaluation procedure for balanced factors in the three more "delicate" points $\{1, -1, 2\}$ (we show the results u_i for a; values v_i for factor b are similarly obtained – we have $w_i = u_iv_i$): the total evaluation cost for both factors is $2(5A + \_1\_2) = 10A + 2(\_1\_2)$.

$$\begin{array}{lccccl}
 & & a_2 & a_1 & a_0 & \\
1)\ u_2 = a_2 + a_0 & [& 1 & 0 & 1 &] \quad A \\
2)\ u_1 = u_2 + a_1 & [& 1 & 1 & 1 &] \quad A \\
3)\ u_2 = u_2 - a_1 & [& 1 & -1 & 1 &] \quad A \\
4)\ u_3 = u_1 + a_2 & [& 2 & 1 & 1 &] \quad A \\
5)\ u_3 = (u_3 \ll 1) - a_0 & [& 4 & 2 & 1 &] \quad A + \_1\_2
\end{array} \tag{1}$$

The unbalanced version is instead realized as follows, with a slightly bigger cost: $[7A + 3(\_1\_2)] + 3A = 10A + 3(\_1\_2)$

$$\begin{array}{lcccccl}
 & & a_3 & a_2 & a_1 & a_0 & \\
1)\ u_3 = a_2 + a_0 & [& 0 & 1 & 0 & 1 &] \quad A \\
2)\ u_2 = a_3 + a_1 & [& 1 & 0 & 1 & 0 &] \quad A \\
3)\ u_1 = u_2 + u_3 & [& 1 & 1 & 1 & 1 &] \quad A \\
4)\ u_2 = u_2 - u_3 & [& 1 & -1 & 1 & -1 &] \quad A \\
5)\ u_3 = a_2 + (a_3 \ll 1) & [& 2 & 1 & 0 & 0 &] \quad A + \_1\_2 \\
6)\ u_3 = a_1 + (u_3 \ll 1) & [& 4 & 2 & 1 & 0 &] \quad A + \_1\_2 \\
7)\ u_3 = a_0 + (u_3 \ll 1) & [& 8 & 4 & 2 & 1 &] \quad A + \_1\_2
\end{array}
\qquad
\begin{array}{rcccl}
 & & b_1 & b_0 & \\
8)\ v_1 = b_1 + b_0 & [& 1 & 1 &] \quad A \\
9)\ v_2 = b_1 - b_0 & [& 1 & -1 &] \quad A \\
10)\ v_3 = v_1 + b_1 & [& 2 & 1 &] \quad A
\end{array} \tag{2}$$

Zimmermann, in GMP library [4] version 4.2.1, proposed the following sequence of operations (inversion sequence, or IS for short) to invert M.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 16 & 8 & 4 & 2 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{M_2+(2)M_3} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 18 & 6 & 6 & 0 & 3 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow[M_3+M_4]{M_2/(3)} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 6 & 2 & 2 & 0 & 1 \\ 2 & 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\xrightarrow[M_3\gg(1)]{M_2+M_5} \begin{pmatrix} 1&0&0&0&0 \\ 6&2&2&0&2 \\ 1&0&1&0&1 \\ 1&1&1&1&1 \\ 0&0&0&0&1 \end{pmatrix} \xrightarrow[M_2-(2)M_1]{M_2\gg(1)} \begin{pmatrix} 1&0&0&0&0 \\ 1&1&1&0&1 \\ 1&0&1&0&1 \\ 1&1&1&1&1 \\ 0&0&0&0&1 \end{pmatrix} \xrightarrow{M_4-M_2} \begin{pmatrix} 1&0&0&0&0 \\ 1&1&1&0&1 \\ 1&0&1&0&1 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{pmatrix} \xrightarrow{M_2-M_3} \begin{pmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 1&0&1&0&1 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{pmatrix} \xrightarrow[M_3-M_5]{M_3-M_1} I$$

with computational cost $$cost_{GMP} = 8A + D_{(3)} + 2S + 2(\_1\_2)$$

More efficient inversion sequences have already been found by exhaustive search in [2] with smaller cost:

$$cost_{BZ} = 8A + D_{(3)} + 2S + (\_1\_2)$$

3 Toom-3 with (at least) an even factor

We consider here the case in which at least one of a_0, b_0 is even, so that $c_0 = a_0 b_0$ is even, too. This happens in 75 % of all possible cases, and its detection is quite fast (just test the least meaningful bit of a_0 and possibly b_0). We will distinguish the balanced and unbalanced versions.

3.1 Evaluation phase: the balanced case

Factors have here the same degree. To fix ideas, without loss of generality let's suppose a_0 is even. Then, from

$$M_2 \;:\; (4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0) = 16c_4 + 8c_3 + 4c_2 + 2c_1 + c_0$$

dividing by 2 both sides we still obtain integer numbers

$$M_2' \;:\; \left(2a_2 + a_1 + \frac{a_0}{2}\right)(4b_2 + 2b_1 + b_0) = 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}$$

The new evaluation sequence (ES, for short) for factor a is shown aside. It has exactly the same computational complexity of (1): note infact that we do not have to *explicitly* divide by 2. This will help us in the interpolation phase, permitting to save one shift.

$$\begin{array}{lcccccl}
 & & a_2 & a_1 & a_0 & & \\
1)\ u_2 = a_2 + a_0 & [& 1 & 0 & 1 &] & A \\
2)\ u_1 = u_2 + a_1 & [& 1 & 1 & 1 &] & A \\
3)\ u_2 = u_2 - a_1 & [& 1 & -1 & 1 &] & A \\
4)\ u_3 = u_2 + a_2 & [& 2 & 1 & 1 &] & A \\
5)\ u_3 = u_3 - (a_0 \gg 1) & [& 2 & 1 & \frac{1}{2} &] & A + \_1\_2
\end{array}$$

3.2 Evaluation phase: the unbalanced case

In the unbalanced case we have $\deg(a) = 3$ and $\deg(b) = 1$. We have two asymmetrical subcases:

- When a_0 is even we have

$$M_2' \;:\; \left(4a_3 + 2a_2 + a_1 + \frac{a_0}{2}\right)(2b_1 + b_0) = 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}$$

  and the new evaluation of the first factor is slightly different but not worse than before: the whole evaluation cost does therefore not change with respect to the balanced case.

$$\begin{array}{lccccccl}
 & & a_3 & a_2 & a_1 & a_0 & & \\
1')\ u_3 = a_2 + a_0 & [& 0 & 1 & 0 & 1 &] & A \\
2')\ u_2 = a_3 + a_1 & [& 1 & 0 & 1 & 0 &] & A \\
3')\ u_1 = u_2 + u_3 & [& 1 & 1 & 1 & 1 &] & A \\
4')\ u_2 = u_2 - u_3 & [& 1 & -1 & 1 & -1 &] & A \\
5')\ u_3 = a_2 + (a_3 \ll 1) & [& 2 & 1 & 0 & 0 &] & A + \_1\_2 \\
6')\ u_3 = a_1 + (u_3 \ll 1) & [& 4 & 2 & 1 & 0 &] & A + \_1\_2 \\
7')\ u_3 = u_3 + (a_0 \gg 1) & [& 4 & 2 & 1 & \frac{1}{2} &] & A + \_1\_2
\end{array}$$

- When b_0 is even, the situation gets unfortunately a bit worse:

$$M_2 \;:\; (8a_3 + 4a_2 + 2a_1 + a_0)\left(b_1 + \frac{b_0}{2}\right) = 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}$$

  The ES cost for the second factor grows then to $3A + (\_1\_2)$, so that the whole evaluation cost is $10A + 4(\_1\_2)$.

$$\begin{array}{lccccl}
 & & b_1 & b_0 & & \\
9')\ v_1 = b_1 + b_0 & [& 1 & 1 &] & A \\
10')\ v_2 = b_1 - b_0 & [& 1 & -1 &] & A \\
11')\ v_3 = b_1 + (b_0 \gg 1) & [& 1 & \frac{1}{2} &] & A + \_1\_2
\end{array}$$

Obviously, if both a_0 and b_0 are even, it is therefore preferrable to aproach the first subcase.

3.3 Interpolation phase

We can therefore consider a different matrix M' to be inverted, with the second line divided by 2.

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 8 & 4 & 2 & 1 & \frac{1}{2} \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

We propose for it the following inversion sequence, someway inspired by Zimmermann's:

$$M' \overset{M_2'+M_3'}{\Longrightarrow} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 9 & 3 & 3 & 0 & \frac{3}{2} \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \underset{M_3'+M_4'}{\overset{M_2'/(3)}{\Longrightarrow}} \begin{pmatrix} 1&0&0&0&0 \\ 3&1&1&0&\frac{1}{2} \\ 2&0&2&0&2 \\ 1&1&1&1&1 \\ 0&0&0&0&1 \end{pmatrix} \underset{M_2'-(2)M_1'}{\overset{M_2'+(\frac{1}{2})M_5'}{\Longrightarrow}} \begin{pmatrix} 1&0&0&0&0 \\ 1&1&1&0&1 \\ 2&0&2&0&2 \\ 1&1&1&1&1 \\ 0&0&0&0&1 \end{pmatrix} \underset{M_3'\gg(1)}{\overset{M_4'-M_2'}{\Longrightarrow}} \begin{pmatrix} 1&0&0&0&0 \\ 1&1&1&0&1 \\ 1&0&1&0&1 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{pmatrix} \overset{M_2'-M_3'}{\Longrightarrow} \begin{pmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 1&0&1&0&1 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{pmatrix} \underset{M_3'-M_5'}{\overset{M_3'-M_1'}{\Longrightarrow}} I$$

Its computational cost is $8A + D_{(3)} + S + 2(\_1\_2)$, smaller than $cost_{GMP}$. We could here avoid a shift because we implicitly did it in the evaluation phase, but with no (small, when only b_0 is even in the unbalanced case) extra cost.

We report below an implementation in `gp-pari`: the three possible ES's and the common IS.

```
\\ Evaluation: Unbalanced case (a even).
a = a3*x^3 + a2*x^2 + a1*x + a0; b = b1*x + b0;

w0 = a2 + a0;            w4 = b1 - b0;
w1 = a3 + a1;
w3 = w1 - w0;
w2 = w4*w3; \\ Evaluation in (-1)
w3 = w1 + w0;            w4 = b1 + b0;
w1 = w3*w4; \\ Evaluation in (1)
w0 = a2 + (a3<<1);       w4 = w4 + b1;
w0 = a1 + (w0<<1);
w0 = w0 + (a0>>1);
w3 = w0*w4; \\ Evaluation in (2) divided by 2.
w0 = a0*b0; \\ Evaluation in (0)
w4 = a3*b1; \\ Evaluation in (1/0)

\\ Evaluation: Balanced case (a even)
a = a2*x^2 + a1*x + a0; b = b2*x^2 + b1*x + b0;

w0 = a2 + a0;            w4 = b2 + b0;
w1 = w0 - a1;            w3 = w4 - b1;
w2 = w1*w3; \\ Evaluation in (-1)
w0 = w0 + a1;            w4 = w4 + b1;
w1 = w0*w4; \\ Evaluation in (1)
w0 = w0 + a2;            w4 = w4 + b2;
w0 = w0 - (a0>>1);       w4 = (w4<<1) - b0;
w3 = w0*w4; \\ Evaluation in (2) divided by 2.
w0 = a0*b0; \\ Evaluation in (0)
w4 = a2*b2; \\ Evaluation in (1/0)
```

```
\\ Evaluation: Unbalanced case (b even).
a = a3*x^3 + a2*x^2 + a1*x + a0; b = b1*x + b0;

w0 = a2 + a0;            w4 = b1 - b0;
w1 = a3 + a1;
w3 = w1 - w0;
w2 = w4*w3; \\ Evaluation in (-1)
w3 = w1 + w0;            w4 = b1 + b0;
w1 = w3*w4; \\ Evaluation in (1)
w0 = a2 + (a3<<1);       w4 = b1 + (b0>>1);
w0 = a1 + (w0<<1);
w0 = a0 + (w0<<1);
w3 = w0*w4; \\ Evaluation in (2) divided by 2.
w0 = a0*b0; \\ Evaluation in (0)
w4 = a3*b1; \\ Evaluation in (1/0)

\\ Interpolation
w3 = w3 + w2;        \\ A        (9 3 3 0 3/2)
w3 = w3 / 3;         \\ D        (3 1 1 0 1/2)
w2 = w2 + w1;        \\ A        (2 0 2 0 2)
w3 = w3 + (w0>>1);   \\ A + _1_2 (3 1 1 0 1)
w3 = w3 - (w4<<1);   \\ A + _1_2 (1 1 1 0 1)
w1 = w1 - w3;        \\ A        (0 0 0 1 0)
w2 = w2 >> 1;        \\ S        (1 0 1 0 1)
w3 = w3 - w2;        \\ A        (0 1 0 0 0)
w2 = w2 - w0;        \\ A        (1 0 1 0 0)
w2 = w2 - w4;        \\ A        (0 0 1 0 0)

c = w4*x^4 + w3*x^3 + w2*x^2 + w1*x + w0;
```

We point out that one could equivalently use -2 as interpolation value instead of 2. This reduces the probability of carry in M_2 computation, but one has then to cope with more negative values.

4 Toom-3.5 with (at least) an even factor

When $\deg(c) = \deg(a) + \deg(b) = 5$ we may apply the Toom-3.5 method, which is intrinsically unbalanced. The used interpolation values are $\{\infty, -2, 2, 1, -1, 0\}$. There are two versions: the slightly and the very unbalanced one, and for each of them we have to consider two cases, depending on which among a_0, b_0 is even.

4.1 Evaluation phase: the slightly unbalanced case

When $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ and $b(x) = b_2x^2 + b_1x + b_0$, if sa is available, the ES is reported below: the cost for a is $8A + 4(\_1\_2)$, while the cost for b is $6A + 2(\_1\_2)$. The total cost is $14A + 6(\_1\_2)$.

$$
\begin{array}{rcrrrrcl}
 & & a_3 & a_2 & a_1 & a_0 & & \\
1)\ u_2 = a_3 + a_1 & [& 1 & 0 & 1 & 0 &] & A \\
2)\ u_3 = a_2 + a_0 & [& 0 & 1 & 0 & 1 &] & A \\
3)\ u_4 = u_2 - u_3 & [& -1 & 1 & -1 & 1 &] & A \\
4)\ u_5 = u_2 + u_3 & [& 1 & 1 & 1 & 1 &] & A \\
5)\ u_1 = a_0 + (a_2 \ll 2) & [& 0 & 4 & 0 & 1 &] & A + \_1\_2 \\
6)\ u_3 = a_1 + (a_3 \ll 2) & [& 4 & 0 & 1 & 0 &] & A + \_1\_2 \\
7)\ u_2 = u_1 - (u_3 \ll 1) & [& -8 & 4 & -2 & 1 &] & A + \_1\_2 \\
8)\ u_3 = u_1 + (u_3 \ll 1) & [& 8 & 4 & 2 & 1 &] & A + \_1\_2
\end{array}
\qquad
\begin{array}{rcrrrcl}
 & & b_2 & b_1 & b_0 & & \\
9)\ v_5 = b_2 + b_0 & [& 1 & 0 & 1 &] & A \\
10)\ v_4 = v_5 - b_1 & [& 1 & -1 & 1 &] & A \\
11)\ v_5 = v_5 + b_1 & [& 1 & 1 & 1 &] & A \\
12)\ v_2 = v_5 + b_2 & [& 2 & 1 & 1 &] & A \\
13)\ v_3 = (v_2 \ll 1) - b_0 & [& 4 & 2 & 1 &] & A + \_1\_2 \\
14)\ v_2 = v_3 - (b_1 \ll 2) & [& 4 & -2 & 1 &] & A + \_1\_2 \\
 & & & & & & \\
 & & & & & &
\end{array}
\tag{3}
$$

If sa is not available, the ES is reported below: the cost for a is $8A+3S$, while the cost for b is $6A+2S$ (the proposed ES for b is different from the one that could straightforwardly be obtained from the above one, in order to reduce carry presence probability). The total cost is $14A + 5S$.

$$
\begin{array}{rcrrrrcl}
 & & a_3 & a_2 & a_1 & a_0 & & \\
1)\text{ - }4)\text{ as in eq. }(3) & & & & & & & \\
5)\ u_1 = a_2 \ll 2 & [& 0 & 4 & 0 & 0 &] & S \\
6)\ u_1 = u_1 + a_0 & [& 0 & 4 & 0 & 1 &] & A \\
7)\ u_3 = a_3 \ll 2 & [& 4 & 0 & 0 & 0 &] & S \\
8)\ u_3 = u_3 + a_1 & [& 4 & 0 & 1 & 0 &] & A \\
9)\ u_3 = u_3 \ll 1 & [& 8 & 0 & 2 & 0 &] & S \\
10)\ u_2 = u_1 - u_3 & [& -8 & 4 & -2 & 1 &] & A \\
11)\ u_3 = u_1 + u_3 & [& 8 & 4 & 2 & 1 &] & A
\end{array}
\qquad
\begin{array}{rcrrrcl}
 & & b_2 & b_1 & b_0 & & \\
12)\ v_5 = b_2 + b_0 & [& 1 & 0 & 1 &] & A \\
13)\ v_4 = v_5 - b_1 & [& 1 & -1 & 1 &] & A \\
14)\ v_5 = v_5 + b_1 & [& 1 & 1 & 1 &] & A \\
15)\ v_2 = b_2 \ll 2 & [& 4 & 0 & 0 &] & S \\
16)\ v_2 = v_2 + b_0 & [& 4 & 0 & 1 &] & A \\
17)\ v_1 = b_1 \ll 1 & [& 0 & 2 & 0 &] & S \\
18)\ v_3 = v_2 + v_1 & [& 4 & 2 & 1 &] & A \\
19)\ v_2 = v_2 - v_1 & [& 4 & -2 & 1 &] & A
\end{array}
\tag{4}
$$

– When a_0 is even, dividing by 2 for the interpolation value $x = 2$ (and similarly for $x = -2$) we have

$$
M_2' \ :\ \left(4a_3 + 2a_2 + a_1 + \frac{a_0}{2}\right)(4b_2 + 2b_1 + b_0) = 16c_5 + 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}
$$

The ES for a changes, but the cost does not. We show it in both cases, when sa is and is not available, respectively.

$$
\begin{array}{rcrrrrcl}
 & & a_3 & a_2 & a_1 & a_0 & & \\
1')\text{ - }4')\text{ as }1)\text{ - }4)\text{ in eq. }(3) & & & & & & & \\
5')\ u_1 = a_0 + (a_2 \ll 2) & [& 0 & 4 & 0 & 1 &] & A + \_1\_2 \\
6')\ u_3 = a_1 + (a_3 \ll 2) & [& 4 & 0 & 1 & 0 &] & A + \_1\_2 \\
7')\ u_2 = (u_1 \gg 1) - u_3 & [& -4 & 2 & -1 & \frac{1}{2} &] & A + \_1\_2 \\
8')\ u_3 = (u_1 \gg 1) + u_3 & [& 4 & 2 & 1 & \frac{1}{2} &] & A + \_1\_2 \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & &
\end{array}
\qquad
\begin{array}{rcrrrrcl}
 & & a_3 & a_2 & a_1 & a_0 & & \\
1')\text{ - }4')\text{ as }1)\text{ - }4)\text{ in eq. }(4) & & & & & & & \\
5')\ u_1 = a_2 \ll 1 & [& 0 & 2 & 0 & 0 &] & S \\
6')\ u_2 = a_0 \gg 1 & [& 0 & 0 & 0 & \frac{1}{2} &] & S \\
7')\ u_1 = u_1 + u_2 & [& 0 & 2 & 0 & \frac{1}{2} &] & A \\
8')\ u_3 = a_3 \ll 2 & [& 4 & 0 & 0 & 0 &] & S \\
9')\ u_3 = u_3 + a_1 & [& 4 & 0 & 1 & 0 &] & A \\
10')\ u_2 = u_1 - u_3 & [& -4 & 2 & -1 & \frac{1}{2} &] & A \\
11')\ u_3 = u_1 + u_3 & [& 4 & 2 & 1 & \frac{1}{2} &] & A
\end{array}
\tag{5}
$$

– When b_0 is even, dividing by 2 for the interpolation value $x = 2$ (and similarly for $x = -2$) we instead have

$$
M_2' \ :\ (8a_3 + 4a_2 + 2a_1 + a_0)\left(2b_2 + b_1 + \frac{b_0}{2}\right) = 16c_5 + 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}
$$

The ES for b changes, but the cost does not. We show both cases, when sa is and is not available, respectively.

$$
\begin{array}{rcrrrcl}
 & & b_2 & b_1 & b_0 & & \\
9')\ v_5 = b_2 + b_0 & [& 1 & 0 & 1 &] & A \\
10')\ v_4 = v_2 - b_1 & [& 1 & -1 & 1 &] & A \\
11')\ v_5 = v_2 + b_1 & [& 1 & 1 & 1 &] & A \\
12')\ v_1 = v_5 + b_2 & [& 2 & 1 & 1 &] & A \\
13')\ v_1 = v_5 - (b_0 \gg 1) & [& 2 & 1 & \frac{1}{2} &] & A + \_1\_2 \\
14')\ v_2 = v_1 - (b_1 \ll 1) & [& 2 & -1 & \frac{1}{2} &] & A + \_1\_2 \\
 & & & & & & \\
 & & & & & &
\end{array}
\qquad
\begin{array}{rcrrrcl}
 & & b_2 & b_1 & b_0 & & \\
12')\ v_5 = b_2 + b_0 & [& 1 & 0 & 1 &] & A \\
13')\ v_4 = v_5 - b_1 & [& 1 & -1 & 1 &] & A \\
14')\ v_5 = v_5 + b_1 & [& 1 & 1 & 1 &] & A \\
15')\ v_1 = b_2 \ll 1 & [& 2 & 0 & 0 &] & S \\
16')\ v_2 = b_0 \gg 1 & [& 0 & 0 & \frac{1}{2} &] & S \\
17')\ v_1 = v_1 + v_2 & [& 2 & 0 & \frac{1}{2} &] & A \\
18')\ v_3 = v_2 + b_1 & [& 2 & 1 & \frac{1}{2} &] & A \\
19')\ v_2 = v_2 - b_1 & [& 2 & -1 & \frac{1}{2} &] & A
\end{array}
\tag{6}
$$

4.2 Evaluation phase: the very unbalanced case

When $a(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, $b(x) = b_1x + b_0$, if sa is available, the ES is reported below: the cost for a is $10A + 5(\_1\_2)$, for b is $4A$. The total cost is $14A + 5(\_1\_2)$.

$$\begin{array}{lcrrrrrcl}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & & \\
1)\ u_2 = a_3 + a_1 & [& 0 & 1 & 0 & 1 & 0 &] & A \\
2)\ u_1 = a_4 + a_0 & [& 1 & 0 & 0 & 0 & 1 &] & A \\
3)\ u_1 = u_1 + a_2 & [& 1 & 0 & 1 & 0 & 1 &] & A \\
4)\ u_4 = u_1 - u_2 & [& 1 & -1 & 1 & -1 & 1 &] & A \\
5)\ u_5 = u_1 + u_2 & [& 1 & 1 & 1 & 1 & 1 &] & A \\
6)\ u_1 = a_1 + (a_3 \ll 2) & [& 0 & 4 & 0 & 1 & 0 &] & A + \_1\_2 \\
7)\ u_2 = a_2 + (a_4 \ll 2) & [& 4 & 0 & 1 & 0 & 0 &] & A + \_1\_2 \\
8)\ u_3 = a_0 + (u_2 \ll 2) & [& 16 & 0 & 4 & 0 & 1 &] & A + \_1\_2 \\
9)\ u_2 = u_3 - (u_1 \ll 1) & [& 16 & -8 & 4 & -2 & 1 &] & A + \_1\_2 \\
10)\ u_3 = u_3 + (u_1 \ll 1) & [& 16 & 8 & 4 & 2 & 1 &] & A + \_1\_2
\end{array}
\qquad
\begin{array}{lcrrcl}
 & & b_1 & b_0 & & \\
11)\ v_4 = b_0 - b_1 & [& -1 & 1 &] & A \\
12)\ v_5 = b_0 + b_1 & [& 1 & 1 &] & A \\
13)\ v_2 = v_4 - b_1 & [& -2 & 1 &] & A \\
14)\ v_3 = v_5 + b_1 & [& 2 & 1 &] & A
\end{array}
\tag{7}$$

If sa is not available, the ES for a is reported below (the ES of b does not change): its cost is $10A + 4S$. The total cost is $14A + 4S$.

$$\begin{array}{lcrrrrrcl}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & & \\
1)\text{ - }5)\text{ as in eq. }(7) & & & & & & & & \\
6)\ u_1 = a_3 \ll 2 & [& 0 & 4 & 0 & 0 & 0 &] & S \\
7)\ u_1 = u_1 + a_1 & [& 0 & 4 & 0 & 1 & 0 &] & A \\
8)\ u_1 = u_1 \ll 1 & [& 0 & 8 & 0 & 2 & 0 &] & S \\
9)\ u_2 = a_4 \ll 2 & [& 4 & 0 & 0 & 0 & 0 &] & S
\end{array}
\qquad
\begin{array}{lcrrrrrcl}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & & \\
10)\ u_2 = u_2 + a_2 & [& 4 & 0 & 1 & 0 & 0 &] & A \\
11)\ u_3 = u_2 \ll 2 & [& 16 & 0 & 4 & 0 & 0 &] & S \\
12)\ u_3 = u_3 + a_0 & [& 16 & 0 & 4 & 0 & 1 &] & A \\
13)\ u_2 = u_3 - u_1 & [& 16 & -8 & 4 & -2 & 1 &] & A \\
14)\ u_3 = u_3 + u_1 & [& 16 & 8 & 4 & 2 & 1 &] & A
\end{array}
\tag{8}$$

- When a_0 is even, dividing by 2 for the interpolation value $x = 2$ (and similarly for $x = -2$) we have

$$M_2' \ : \ \left(8a_4 + 4a_3 + 2a_2 + a_1 + \frac{a_0}{2}\right)(2b_1 + b_0) = 16c_5 + 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}$$

The ES for a changes, but the cost does not. We show both cases, when sa is and is not available, respectively. The first 5 steps are as in equation (7).

$$\begin{array}{lcrrrrrcl}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & & \\
6')\ u_1 = a_1 + (a_3 \ll 2) & [& 0 & 4 & 0 & 1 & 0 &] & A + \_1\_2 \\
7')\ u_2 = a_2 + (a_4 \ll 2) & [& 4 & 0 & 1 & 0 & 0 &] & A + \_1\_2 \\
8')\ u_3 = a_0 + (u_2 \ll 2) & [& 16 & 0 & 4 & 0 & 1 &] & A + \_1\_2 \\
9')\ u_2 = (u_3 \gg 1) - u_1 & [& 8 & -4 & 2 & -1 & \frac{1}{2} &] & A + \_1\_2 \\
10')\ u_3 = (u_3 \gg 1) + u_1 & [& 8 & 4 & 2 & 1 & \frac{1}{2} &] & A + \_1\_2
\end{array}
\qquad
\begin{array}{lcrrrrrcl}
 & & a_4 & a_3 & a_2 & a_1 & a_0 & & \\
6')\ u_1 = a_3 \ll 2 & [& 0 & 4 & 0 & 0 & 0 &] & S \\
7')\ u_1 = u_1 + a_1 & [& 0 & 4 & 0 & 1 & 0 &] & A \\
8')\ u_2 = a_4 \ll 3 & [& 8 & 0 & 0 & 0 & 0 &] & S \\
9')\ u_3 = a_2 \ll 1 & [& 0 & 0 & 2 & 0 & 0 &] & S \\
10')\ u_2 = u_2 + u_3 & [& 8 & 0 & 2 & 0 & 0 &] & A \\
11')\ u_3 = a_0 \gg 1 & [& 0 & 0 & 0 & 0 & \frac{1}{2} &] & S \\
12')\ u_3 = u_2 + u_3 & [& 8 & 0 & 2 & 0 & \frac{1}{2} &] & A \\
13')\ u_2 = u_3 - u_1 & [& 8 & -4 & 2 & -1 & \frac{1}{2} &] & A \\
14')\ u_3 = u_3 + u_1 & [& 8 & 4 & 2 & 1 & \frac{1}{2} &] & A
\end{array}
\tag{9}$$

- When b_0 is even, dividing by 2 for the interpolation value $x = 2$ (and similarly for $x = -2$) we instead have

$$M_2' \ : \ (16a_4 + 8a_3 + 4a_2 + 2a_1 + a_0)\left(b_1 + \frac{b_0}{2}\right) = 16c_5 + 8c_4 + 4c_3 + 2c_2 + c_1 + \frac{c_0}{2}$$

In this case the situation gets unfortunately a bit worse: The new (partial) ES cost for the second factor grows to $4A + (\_1\_2)$ is sa is available, or $4A + S$ if it is not, respectively.

$$\begin{array}{lcrrcl}
 & & b_1 & b_0 & & \\
12')\ v_4 = b_0 - b_1 & [& -1 & 1 &] & A \\
13')\ v_5 = b_0 + b_1 & [& 1 & 1 &] & A \\
14')\ v_2 = (b_0 \gg 1) - b_1 & [& -1 & \frac{1}{2} &] & A + \_1\_2 \\
15')\ v_3 = b_0 - v_2 & [& 1 & \frac{1}{2} &] & A
\end{array}
\qquad
\begin{array}{lcrrcl}
 & & b_1 & b_0 & & \\
12')\ v_4 = b_0 \gg 1 & [& 0 & \frac{1}{2} &] & S \\
13')\ v_2 = v_4 - b_1 & [& -1 & \frac{1}{2} &] & A \\
14')\ v_3 = v_4 + b_1 & [& 1 & \frac{1}{2} &] & A \\
15')\ v_4 = b_0 - b_1 & [& -1 & 1 &] & A \\
16')\ v_5 = b_0 + b_1 & [& 1 & 1 &] & A
\end{array}
\tag{10}$$

The total ES cost amounts then to $14A + 6(\_1\_2)$ and $14A + 5S$, respectively.

4.3 Interpolation phase

The IS proposed in [1] for the general case has a cost of $12A + 2S + D_{(6)} + D_{(12)} + 2(\_1\_2)$. Strictly following the EMI scheme, we can manage the IS so that the cost becomes one of the following – when *sa* is available or not, respectively. Note that at least one division is now by a different constant.[1]

$$cost' = 12A + S + D_{(3)} + D_{(12)} + 3(\_1\_2) \qquad ; \qquad cost'' = 12A + 4S + 2D_{(3)}$$

The new M matrix resulting from the evaluation values $\{\infty, -2, 2, -1, 1, 0\}$ with second and third line divided by 2 is

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -16 & 8 & -4 & 2 & -1 & \frac{1}{2} \\ 16 & 8 & 4 & 2 & 1 & \frac{1}{2} \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

If *sa* is available, the proposed IS is:

$$M \xRightarrow[M_4+M_5]{M_2+M_3} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 1 \\ 16 & 8 & 4 & 2 & 1 & \frac{1}{2} \\ 0 & 2 & 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow[M_4 \gg 1]{M_3-(\frac{1}{2})M_2} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 1 \\ 16 & 0 & 4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow[M_5-M_4]{M_2-M_6} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 0 \\ 16 & 0 & 4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow[M_4-M_6]{M_3-M_5} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 0 \\ 15 & 0 & 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\xRightarrow[M_3/(3)]{M_2-(4)M_4} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow[M_3-(4)M_1]{M_2/(12)} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow[M_5-M_3]{M_4-M_2} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xRightarrow{M_3-M_1} I$$

If *sa* is not available, the IS is slightly different, and a small "trick" has to be used in order not to use any extra temporary variable: evaluation and interpolation phases have to be interlaced someway. We don't report explicitly this second IS: it can be easily deduced from the code reported in appendix A or B.

If we instead mix a bit IS and ES we can also work as follows: rewrite ES so that the interpolation matrix is the shown aside M: that is, simply reorganize the ES in order to put the evaluation in -2 (divided by 2) in M_6, where $c_0 = a_0 b_0$ – the evaluation in 0 – should be, and do not compute c_0. Note that M_2 line remains for now undefined. The IS (when *sa* is available) becomes then the following one, otherwise the above considerations apply:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \multicolumn{6}{c}{\langle\text{not computed value}\rangle} \\ 16 & 8 & 4 & 2 & 1 & \frac{1}{2} \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ -16 & 8 & -4 & 2 & -1 & \frac{1}{2} \end{pmatrix}$$

$$M \xRightarrow[M_5-M_4]{M_2=M_3+M_6} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 1 \\ 16 & 8 & 4 & 2 & 1 & \frac{1}{2} \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 2 & 0 & 2 & 0 & 2 & 0 \\ -16 & 8 & -4 & 2 & -1 & \frac{1}{2} \end{pmatrix} \xRightarrow[M_6=a_0b_0]{M_3-M_6} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 4 & 0 & 1 \\ 32 & 0 & 8 & 0 & 2 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 2 & 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Longrightarrow \cdots \langle\text{as in [2]}\rangle \cdots \Longrightarrow I$$

The cost becomes one of the following – when *sa* is available or not, respectively. Note that at least one division is now by a different constant.

$$cost' = 12A + S + D_{(6)} + D_{(12)} + 2(\_1\_2) \qquad ; \qquad cost'' = 12A + 3S + D_{(3)} + D_{(6)}$$

We don't provide the code in this case, as it can be easily deduced from the one relative to the precedent case.

[1] As GMP has an optimized function to divide a long integer by 3, it can be directly used, gaining efficiency.

5 Some savings in Toom-4.5 and Toom-5

Good IS's for Toom-4.5 and Toom-5 were introduced in [2]. They were not proven to be optimal (in the considered model): because of their too big dimension it was impossible to find the optimal IS by exhaustive search. By slightly changing the model (in particular, "interlacing" ES and IS), it is possible to avoid one _1_2 (or a shift) for Toom-4.5 and two for Toom-5, for whatever a, b.

5.1 Saving in Toom-4.5

It uses $\left\{\infty, -1, -2, \frac{1}{2}, 1, 2, -\frac{1}{2}, 0\right\}$ as interpolation points (with lines corresponding to $\pm\frac{1}{2}$ opportunely multiplied by 2^7). The matrix is

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -128 & 64 & -32 & 16 & -8 & 4 & -2 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & -2 & 4 & -8 & 16 & -32 & 64 & -128 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The proposed IS contains the "decoupling" instructions for lines 3 and 6, related to values -2 and 2: the partial cost of these two operations is $2A +$ _1_2.

$$\begin{array}{lcl} M_3 = M_3 - M_6 & ; & (\,-256 \;\; 0 \; -64 \;\; 0 \; -16 \;\; 0 \; -4 \;\; 0\,) \\ M_6 = (M_6 \ll 1) - M_3 & ; & (\;\; 0 \;\; 128 \;\; 0 \;\; 32 \;\; 0 \;\; 8 \;\; 0 \;\; 2\,) \end{array}$$

By slightly mixing evaluation and interpolation phases it is possible to reduce this cost to $2A$, as follows:

- First perform the ES for Toom-4.5 giving M, but modified such in a way that the obtained interpolation matrix is the aside shown M': that is, simply reorganize the ES in order to put the evaluation in -2 in M_8, where $c_0 = a_0 b_0$ – the evaluation in 0 – should be, and do not compute c_0. Note that M_3 line remains for now undefined.

$$M' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ \multicolumn{8}{c}{\langle \text{not computed value} \rangle} \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & -2 & 4 & -8 & 16 & -32 & 64 & -128 \\ -128 & 64 & -32 & 16 & -8 & 4 & -2 & 1 \end{pmatrix}$$

- Then decouple and complete the ES

$$\begin{array}{lcl} M_3 = M_6 - M_8 & ; & (\,256 \;\; 0 \;\; 64 \;\; 0 \;\; 16 \;\; 0 \;\; 4 \;\; 0\,) \\ M_6 = M_6 + M_8 & ; & (\;\; 0 \;\; 128 \;\; 0 \;\; 32 \;\; 0 \;\; 8 \;\; 0 \;\; 2\,) \\ M_8 = a_0 b_0 & ; & (\;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1\,) \end{array}$$

- Finally complete the IS following [2].

5.2 Saving in Toom-5

The corresponding matrix M is shown below: the IS proposed in [2] contained two decouplings: for values ± 2 (lines 2 and 5) and $\pm\frac{1}{2}$ (lines 3 and 8). In this case mixing ES and IS results in a saving of 2(_1_2).

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 256 & -128 & 64 & -32 & 16 & -8 & 4 & -2 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \\ 4^8 & 4^7 & 4^6 & 4^5 & 256 & 64 & 16 & 4 & 1 \\ 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 4 & -8 & 16 & -32 & 64 & -128 & 256 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- First perform the ES for Toom-5 giving M, but modified such in a way that the obtained interpolation matrix is the aside shown M': that is, simply reorganize the ES in order to put the evaluation in -2 in M_1, the evaluation in $-\frac{1}{2}$ (multiplied by 2^8) in M_9, and do not compute c_0 and c_9. Note that M_2 and M_8 lines remain for now undefined.

$$M' = \begin{pmatrix} 256 & -128 & 64 & -32 & 16 & -8 & 4 & -2 & 1 \\ \multicolumn{9}{c}{\langle\text{not computed value}\rangle} \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \\ 4^8 & 4^7 & 4^6 & 4^5 & 256 & 64 & 16 & 4 & 1 \\ 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \multicolumn{9}{c}{\langle\text{not computed value}\rangle} \\ 1 & -2 & 4 & -8 & 16 & -32 & 64 & -128 & 256 \end{pmatrix}$$

- Then decouple and adapt some IS steps to the new situation, as it must be slightly modified. The IS is a bit involved, we report `gp-pari` code for it:

```
\\ Evaluation of W0,...,W8 such that
W0 <-- Evaluation in (-1/2) times 2^8
W1 <-- Still not inizialized
W2 <-- Evaluation in (1)
W3 <-- Evaluation in (-1)
W4 <-- Evaluation in (2)
W5 <-- Evaluation in (1/2) times 2^8
W7 <-- Still not inizialized
W8 <-- Evaluation in (-2)

\\ Interpolation (and evaluation completion)
W7 = W4 - W8;
W4 = W4 + W8;
W1 = W6 - W0;
W6 = W6 + W0;
W8 = Evaluation in (1/0);
W0 = Evaluation in (0);
W2 = W2 + W3;
W6 = W6 - W2;
W2 = W2 >> 1;
W3 = W2 - W3;
W5 = W5 - W0;
W5 = W5 - (W8<<16);
W4 = W4 - (W2<<9);
W2 = W2 - W8;
W2 = W2 - W0;
```

```
W1 = W1 + W7;
W4 = W4 + W6;
W1 = W1 - (80*W3);
W6 = W6 - (510*W0);
W5 = W5 - W7;
W6 = 3*W6 + W4;
W1 = W1/180;
W4 = W4 + (378*W2);
W7 = W7>>2;
W3 = W3 - W7;
W4 = W4 /(-72);
W6 = W6 /(-360);
W7 = W7 - W1;
W2 = W2 - W6;
W5 = W5 - (W4<<8);
W6 = W6 - W4;
W5 = W5 - (W6<<12);
W5 = W5 - (W2<<4);
W5 = W5 + (W3<<8);
W3 = W3 + W7;
W7 = (W7*180) + W5;
W7 = W7/11340;
W5 = W5 + (720*W3);
W5 = W5/(-2160);
W3 = W3 - W5;
W1 = W1 - W7;
```

6 Conclusions

We showed how some Toom-Cook methods can be modified when one factor is even, permitting some savings. Lower-level implementations should be realized in order to understand the real gain over the classical methods. Possible optimizations of the interpolation phase of Toom-4.5 and Toom-5 were also suggested.

7 Acknowledgements

The author wishes to thank Marco Bodrato for many helpful discussions and fruitful suggestions and considerations. This work is dedicated to the memory of the author's beloved grandfather, Lino Broffoni.

References

1. Marco Bodrato. Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0. In Claude Carlet and Berk Sunar, editors, *WAIFI '07 proceedings*, volume 4547 of *Lecture Notes in Computer Science*. Springer, June 2007.
2. Marco Bodrato and Alberto Zanoni. Integer and polynomial multiplication: Towards optimal Toom-Cook matrices. In Christopher W. Brown, editor, *Proceedings of the ISSAC 2007 Conference*. ACM press, July 2007. URL: `http://bodrato.it/papers/#ISSAC2007`.
3. Stephen A. Cook. *On the minimum computation time of functions*. PhD thesis, Department of Mathematics, Harvard University, 1966.
4. The GNU multiple precision (GMP) library documentation. URL: `http://gmplib.org/#DOC`.
5. Andrei L. Toom. The complexity of a scheme of functional elements realizing the multiplication of integers. *Soviet Mathematics Doklady*, 3:714–716, 1963. URL: `http://www.de.ufpe.br/ toom/articles/engmat/MULT-E.PDF`.

A Toom-3.5 code when a_0 is even

1. **Slightly unbalanced case :** We propose two complete procedures: when *sa* is or is not available. To avoid the use of *any* extra temporaries, the version without *sa* is sensibly different: evaluation and interpolation sequence are mixed. The values for c_0 is computed during the IS, as soon it is needed, so that the variable that will contain it can meanwhile be used as a temporary one. In the code presented when *sa* is available we suppose that $2(\_1\_2) \leqslant S$. If it is not the case, one should have

```
W0 = W0>>1;
W4 = W0 + W3;
W0 = W0 - W3;
```

instead of

```
W4 = (W0>>1) + W3;
W0 = (W0>>1) - W3;
```

With *sa*

```
a = a3*x^3 + a2*x^2 + a1*x + a0;
b =          b2*x^2 + b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a2;        W1 = b0 + b2;
W4 = a1 + a3;        W5 = W1 - b1;
W3 = W0 - W4;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;        W5 = W1 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
W0 = a0 + (a2<<2);   W5 = W5 + b2;
W3 = a1 + (a3<<2);   W5 = (W5<<1) - b0;
W4 = (W0>>1) + W3;
W0 = (W0>>1) - W3;
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W5 = W5 - (b1<<2);
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2.
W0 = a0*b0;  \\ Evaluation in (0)
W5 = a3*b2;  \\ Evaluation in (1/0)

\\\\\\\\\\\\ Interpolation
W4 = W4 + W3;
W2 = W2 + W1;
W3 = W3 - (W4>>1);
W2 = W2>>1;
W4 = W4 - W0;
W1 = W1 - W2;
W3 = W3 - W1;
W2 = W2 - W0;
W4 = W4 - (W2<<2);
W3 = W3/3;
W4 = W4/12;
W3 = W3 - (W5<<2);
W1 = W1 - W3;
W2 = W2 - W4;
W3 = W3 - W5;

c = W5*x^5 + W4*x^4 + W3*x^3 + W2*x^2 + W1*x + W0;
```

Without *sa*

```
a = a3*x^3 + a2*x^2 + a1*x + a0;
b =          b2*x^2 + b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a2;        W1 = b0 + b2;
W4 = a1 + a3;        W5 = W1 - b1;
W3 = W0 - W4;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;        W5 = W1 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
W0 = a2<<1;          W5 = W5 + b2;
W3 = a0>>1;          W5 = W5<<1;
W0 = W0 + W3;        W5 = W5 - b0;
W3 = a3<<2;
W3 = W3 + a1;
W4 = W0 + W3;
W0 = W0 - W3;
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W4 = b1<<2;
                     W5 = W5 - W4;
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2.
W5 = a3*b2;  \\ Evaluation in (1/0)

\\\\\\\\\\\\ Interpolation
W4 = W4 + W3;
W2 = W2 + W1;
W0 = W4>>1;
W3 = W3 - W0;
W2 = W2>>1;
W1 = W1 - W2;
W3 = W3 - W1;
W3 = W3/3;
W0 = W5<<2;
W3 = W3 - W0;
W1 = W1 - W3;
W3 = W3 - W5;
W0 = a0*b0;  \\ Evaluation in 0.
W4 = W4 - W2;
W4 = W4/3;
W2 = W2 - W0;
W4 = W4 - W2;
W4 = W4>>2;
W2 = W2 - W4;

c = W5*x^5 + W4*x^4 + W3*x^3 + W2*x^2 + W1*x + W0;
```

2. **Very unbalanced case :** We propose just the evaluation phases when *sa* is (complete ES) or is not (not complete ES) available. The interpolation (pure or mixed) and the final reconstruction are the same as above, respectively.

With *sa*

```
a = a4*x^4 + a3*x^3 + a2*x^2 + a1*x + a0;
b =                            b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a4;
W0 = W0 + a2;
W4 = a1 + a3;
W3 = W0 - W4;        W5 = b0 - b1;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;
W1 = a2 + (a4<<2);
W1 = a0 + (W1<<2);
W4 = a1 + (a3<<2);
W0 = (W1>>1) + W4;
W1 = (W1>>1) - W4;   W5 = W5 - b1;
W4 = W1*W5;  \\ Evaluation in (-2) divided by 2.
                     W5 = b0 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
                     W5 = W5 + b1;
W3 = W0*W5;  \\ Evaluation in (2) divided by 2.
W0 = a0*b0;  \\ Evaluation in (0)
W5 = a4*b1;  \\ Evaluation in (1/0)
```

Without *sa*

```
a = a4*x^4 + a3*x^3 + a2*x^2 + a1*x + a0;
b =                            b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a4;
W0 = W0 + a2;
W4 = a1 + a3;
W3 = W0 - W4;        W5 = b0 - b1;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;
W1 = a4<<2;
W1 = W1 + a2;
W1 = W1<<2;
W1 = W1 + a0;
W4 = a3<<2;
W4 = W4 + a1;
W1 = W1>>1;
W0 = W1 + W4;
W1 = W1 - W4;        W5 = W5 - b1;
W4 = W1*W5;  \\ Evaluation in (-2) divided by 2.
                     W5 = b0 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
                     W5 = W5 + b1;
W3 = W0*W5;  \\ Evaluation in (2) divided by 2.
W5 = a4*b1;  \\ Evaluation in (1/0)
```

B Toom-3.5 code when b_0 is even

1. **Slightly unbalanced case :** We propose just the evaluation phases when *sa* is (complete ES) or is not (not complete ES) available. The interpolation (pure or mixed) and the final reconstruction are the same as in the case when a_0 is even, respectively.

With *sa*

```
a = a3*x^3 + a2*x^2 + a1*x + a0;
b =          b2*x^2 + b1*x + b0;
\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a2;        W1 = b0 + b2;
W4 = a1 + a3;        W5 = W1 - b1;
W3 = W0 - W4;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;        W5 = W1 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
W0 = a0 + (a2<<2);   W5 = W5 + b2;
W3 = a1 + (a3<<2);   W5 = W5 - (b0>>1);
W4 = W0 + (W3<<1);
W0 = W0 - (W3<<1);
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W5 = W5 - (b1<<1);
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2
W0 = a0*b0;  \\ Evaluation in (0)
W5 = a3*b2;  \\ Evaluation in (1/0)
```

Without *sa*

```
a = a3*x^3 + a2*x^2 + a1*x + a0;
b =          b2*x^2 + b1*x + b0;
\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a2;        W1 = b0 + b2;
W4 = a1 + a3;        W5 = W1 - b1;
W3 = W0 - W4;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;        W5 = W1 + b1;
W1 = W3*W5;  \\ Evaluation in (1)
W0 = a2<<2;          W5 = W5 + b2;
W0 = W0 + a0;        W4 = b0>>1;
W3 = a3<<2;          W5 = W5 - W4;
W3 = W3 + a1;
W3 = W3<<1;
W4 = W0 + W3;
W0 = W0 - W3;
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W4 = b1<<1;
                     W5 = W5 - W4;
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2.
W5 = a3*b2;  \\ Evaluation in (1/0)
```

2. **Very unbalanced case :** We propose just the evaluation phases when sa is (complete ES) or is not (not complete ES) available. The interpolation (pure or mixed) and the final reconstruction are the same as in the case when a_0 is, respectively.

With *sa*

```
a = a4*x^4 + a3*x^3 + a2*x^2 + a1*x + a0;
b =                            b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a4;
W0 = W0 + a2;
W5 = a1 + a3;
W3 = W0 - W5;        W4 = b0 - b1;
W2 = W3*W4;  \\ Evaluation in (-1)
W3 = W0 + W5;        W4 = b0 + b1;
W1 = W3*W4;  \\ Evaluation in (1)

W0 = a2 + (a4<<2);
W0 = a0 + (W0<<2);
W3 = a1 + (a3<<2);

W4 = W0 + (W3<<1);
W0 = W0 - (W3<<1);   W5 = (b0>>1) + b1;
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W5 = b0 - W5;
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2.

W0 = a0*b0;  \\ Evaluation in (0)
W5 = a4*b1;  \\ Evaluation in (1/0)
```

Without *sa*

```
a = a4*x^4 + a3*x^3 + a2*x^2 + a1*x + a0;
b =                            b1*x + b0;

\\\\\\\\\\\\\\ Evaluation
W0 = a0 + a4;
W0 = W0 + a2;
W4 = a1 + a3;
W3 = W0 - W4;        W5 = b0 - b1;
W2 = W3*W5;  \\ Evaluation in (-1)
W3 = W0 + W4;        W5 = b0 + b1;
W1 = W3*W5;  \\ Evaluation in (1)

W0 = a4<<2;
W0 = W0 + a2;
W0 = W0<<2;
W0 = W0 + a0;
W3 = a3<<2;
W3 = W3 + a1;
W3 = W3<<1;
W4 = W0 + W3;        W5 = b0>>1;
W0 = W0 - W3;        W5 = W5 + b1;
W3 = W4*W5;  \\ Evaluation in (2) divided by 2.
                     W5 = b0 - W5;
W4 = W0*W5;  \\ Evaluation in (-2) divided by 2.
W5 = a4*b1;  \\ Evaluation in (1/O).
```