

REIST Tree: A Conditional Reduction Tree for Centered Remainder Sums

R. Stepan
Independent Researcher
Schwechat, Austria
rstepan@outlook.at

Abstract—This corrected conceptual note defines a parallel reduction for a finite list of independently available integer contributions with known centered quotient–remainder pairs. A bounded pairwise merge preserves both the integer sum and the canonical remainder interval. For n inputs, a balanced tree uses $O(n)$ merges and $O(\log n)$ parallel stages under a fixed-word-cost model with sufficient resources and no overflow. These bounds do not include discovering dependent corrections or dividing arbitrary unnormalized inputs. No measured speedup or new general division algorithm is claimed.

Index Terms—Centered remainder, modular addition, parallel reduction, exact arithmetic.

I. SCOPE OF THE CORRECTION

Earlier versions asserted that potentially active quotient corrections could be evaluated independently without giving a rule to discover them. Preservation of $T = qB + r$ alone is insufficient: for $T = 1$ and $B = 10$, the pair $(q, r) = (1, -9)$ satisfies that identity but violates the centered interval. This revision therefore states a narrower, explicit input contract and algorithm. It uses the centered representation discussed in [1]; expressing arithmetic through signed additions [2] does not remove control or dependency requirements.

II. INPUT CONTRACT

Let B be a positive integer and define

$$l = -\lfloor B/2 \rfloor, \quad h = \lfloor (B-1)/2 \rfloor. \quad (1)$$

There are $n \geq 1$ independently available integers x_i , supplied as exact pairs (q_i, r_i) with $x_i = q_i B + r_i$ and $l \leq r_i \leq h$. The pairs may encode independent increments or already normalized values. Producing them is a separate preprocessing cost. If later inputs depend on earlier states, the reduction cannot be applied as stated.

III. PAIRWISE MERGE

For two pairs (q_a, r_a) and (q_b, r_b) , set $s = r_a + r_b$ and

$$c = \begin{cases} 1, & s > h, \\ -1, & s < l, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The parent pair is

$$q = q_a + q_b + c, \quad r = s - cB. \quad (3)$$

At each stage, merge adjacent pairs; carry an unmatched pair unchanged to the next stage. Stop at one pair. The modulus and tie convention are identical at every node.

IV. CORRECTNESS

First, $qB + r = (q_a + q_b)B + s = x_a + x_b$. Second, $2l \leq s \leq 2h$. If $s > h$, then $s \geq h+1$ and $h+1-B = l$, while $2h-B \leq h$; hence $l \leq s-B \leq h$. If $s < l$, then $s \leq l-1$ and $l-1+B = h$, while $2l+B \geq l$; hence $l \leq s+B \leq h$. Otherwise s is already in range. One correction thus suffices, including $B = 1$ and even-modulus negative endpoints.

Induction over the tree gives $\sum_i x_i = qB + r$ with $r \in [l, h]$. This interval contains exactly B consecutive integers, so its representative and the corresponding quotient are unique. Therefore the root agrees with sequential canonical merging regardless of the grouping. Both the invariant and the range condition are needed for this argument.

V. WORK AND PARALLEL DEPTH

A balanced reduction has $n - 1$ merges and at most $\lceil \log_2 n \rceil$ stages. Thus its merge work is $O(n)$ and its parallel depth is $O(\log n)$ if each merge has constant cost and sufficient processors and communication are available. These are operation-count bounds, not elapsed-time guarantees. A serial implementation still performs linear work.

For machine integers, all sums, quotients and corrections must fit the chosen type. If only the modular sum is needed, quotient tracking can be omitted. With arbitrary-precision arithmetic, operand growth and bit-operation costs must be included; constant-cost merges cannot be assumed. Preprocessing, synchronization, memory traffic and SIMD lane reductions are outside the stated merge count.

VI. EXAMPLE AND LIMITATIONS

For $B = 10$, pairs $(0, -5)$ and $(0, -5)$ merge to $(-1, 0)$, representing -10 . The valid lower endpoint is retained and the carry is explicit. For $B = 5$, $(0, 2)$ and $(0, 2)$ merge to $(1, -1)$, representing 4 .

This construction is a standard parallel-reduction pattern applied to centered representatives. It does not establish novelty over existing modular reduction trees. In particular, it cannot replace a sequential loop whose next correction is determined only by the preceding state. A comparison against both sequential centered addition and non-negative bounded modular addition is still required; no empirical evaluation is supplied here.

VII. CONCLUSION

Independent, already normalized contributions admit an exact centered-remainder reduction tree. The explicit merge establishes the claimed arithmetic result under stated hypotheses. A general logarithmic-depth algorithm for discovering division corrections does not follow.

REFERENCES

- [1] R. Stepan, "REIST Division: An Implementation-Oriented Framing of Centered Remainder Arithmetic," 2025.
- [2] R. Stepan, "On the Reduction of Arithmetic Operations to Signed Addition," 2025.