# Mixed Precision Blocked Summation: Solutions and Variations

## 1 Exercise 7: Mixed Precision Blocked Summation (3 points)

We consider the summation algorithm below, which is parameterized by three precisions $u_1$, $u_2$, and $u_3$, and by two block sizes $b_1$ and $b_2$ (we assume for simplicity that $n$ is a multiple of $b_1 b_2$).

**Algorithm 1** Mixed Precision Blocked Summation

**Input:** $x_1, \ldots, x_n$
**Output:** $\tilde{s} = \sum_{i=1}^{n} x_i$
**for** $i = 1$ to $\frac{n}{b_1}$ **do**
    $y_i = \sum_{j=(i-1)b_1+1}^{ib_1} x_j$ in precision $u_1$
**end for**
**for** $i = 1$ to $\frac{n}{b_1 b_2}$ **do**
    $z_i = \sum_{j=(i-1)b_2+1}^{ib_2} y_j$ in precision $u_2$
**end for**
$\tilde{s} = \sum_{i=1}^{n/b_1 b_2} z_i$ in precision $u_3$

### Questions

1. Draw the summation tree corresponding to this algorithm for $n = 12$, $b_1 = 3$, and $b_2 = 2$ (as a reminder, in a summation tree the leaves correspond to the summands $x_i$, the root corresponds to the sum $s$, and the rest of the nodes correspond to partial sums). You will assign to non-leaf nodes the labels 1, 2, or 3 to indicate which precision $u_1$, $u_2$, or $u_3$ is used.

2. The computed sum $\tilde{s}$ satisfies a backward error bound of the form

$$\tilde{s} = \sum_{i=1}^{n} x_i(1+\theta_i), \quad |\theta_i| \leq f(n, b_1, b_2, u_1, u_2, u_3).$$

   What is the expression of $f(n, b_1, b_2, u_1, u_2, u_3)$?

3. If $u_1 = u_2 = u_3$, which choice of $b_1$ and $b_2$ minimizes the bound?

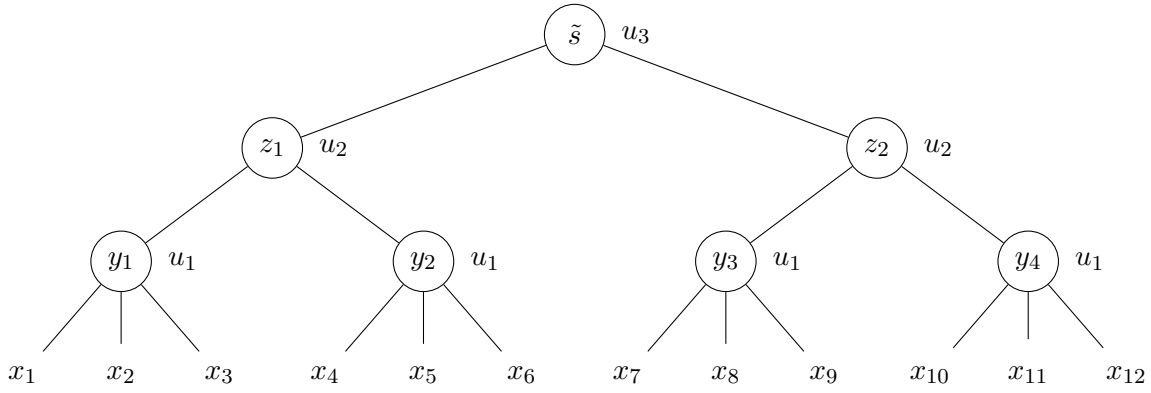
4. If we want to use three different arithmetics (such as fp64, fp32, and fp16), which arithmetic do you propose to assign to which parameter $u_1$, $u_2$, $u_3$? Why?

## 2 Solutions

### Solution 1: Summation Tree

For $n = 12$, $b_1 = 3$, and $b_2 = 2$:

- Number of blocks at first level: $\frac{n}{b_1} = \frac{12}{3} = 4$ (so we get $y_1, y_2, y_3, y_4$)
- Number of blocks at second level: $\frac{n}{b_1 b_2} = \frac{12}{6} = 2$ (so we get $z_1, z_2$)
- Final sum: $\tilde{s} = z_1 + z_2$




The tree shows:

- **Leaves:** $x_1, \ldots, x_{12}$ (input data)
- **Level 1 (precision $u_1$):** $y_1, y_2, y_3, y_4$ (each sums 3 elements)
- **Level 2 (precision $u_2$):** $z_1, z_2$ (each sums 2 $y$ values)
- **Root (precision $u_3$):** $\tilde{s}$ (sums the 2 $z$ values)

### Solution 2: Backward Error Bound

We analyze the error accumulation at each level:

**Level 1 (computing $y_i$ in precision $u_1$):**

Each $y_i$ is the sum of $b_1$ elements computed in precision $u_1$. Using the standard result for sequential summation:

$$y_i = \sum_{j=(i-1)b_1+1}^{ib_1} x_j(1+\eta_{i,j}), \quad |\eta_{i,j}| \leq b_1 u_1$$

**Level 2 (computing $z_i$ in precision $u_2$):**

Each $z_i$ is the sum of $b_2$ elements $y_j$ computed in precision $u_2$:

$$z_i = \sum_{j=(i-1)b_2+1}^{ib_2} y_j(1+\xi_{i,j}), \quad |\xi_{i,j}| \leq b_2 u_2$$

Substituting the expression for $y_j$:

$$z_i = \sum_{j=(i-1)b_2+1}^{ib_2} \sum_{k=(j-1)b_1+1}^{jb_1} x_k(1+\eta_{j,k})(1+\xi_{i,j})$$

**Level 3 (computing $\tilde{s}$ in precision $u_3$):**

$$\tilde{s} = \sum_{i=1}^{n/(b_1b_2)} z_i(1+\zeta_i), \quad |\zeta_i| \leq \frac{n}{b_1b_2}u_3$$

Combining all levels:

$$\tilde{s} = \sum_{i=1}^{n/(b_1b_2)} \sum_{j=(i-1)b_2+1}^{ib_2} \sum_{k=(j-1)b_1+1}^{jb_1} x_k(1+\eta_{j,k})(1+\xi_{i,j})(1+\zeta_i)$$

For small errors, $(1+\eta_{j,k})(1+\xi_{i,j})(1+\zeta_i) \approx 1+\eta_{j,k}+\xi_{i,j}+\zeta_i$.
Therefore:

$$\tilde{s} = \sum_{\ell=1}^{n} x_\ell(1+\theta_\ell)$$

where each $\theta_\ell$ is bounded by the sum of errors from all three levels:

$$|\theta_\ell| \leq b_1u_1 + b_2u_2 + \frac{n}{b_1b_2}u_3$$

Thus:

$$\boxed{f(n, b_1, b_2, u_1, u_2, u_3) = b_1u_1 + b_2u_2 + \frac{n}{b_1b_2}u_3}$$

## Solution 3: Optimal Block Sizes

If $u_1 = u_2 = u_3 = u$, then:

$$f(n, b_1, b_2, u, u, u) = u\left(b_1 + b_2 + \frac{n}{b_1b_2}\right)$$

To minimize this, we need to minimize:

$$g(b_1, b_2) = b_1 + b_2 + \frac{n}{b_1b_2}$$

Taking partial derivatives:

$$\frac{\partial g}{\partial b_1} = 1 - \frac{n}{b_1^2b_2} = 0 \implies b_1^2b_2 = n$$

$$\frac{\partial g}{\partial b_2} = 1 - \frac{n}{b_1b_2^2} = 0 \implies b_1b_2^2 = n$$

From these equations:

$$b_1^2b_2 = b_1b_2^2 \implies b_1 = b_2$$

Substituting back:

$$b_1^3 = n \implies b_1 = b_2 = n^{1/3}$$

Therefore, the optimal choice is:

$$\boxed{b_1 = b_2 = n^{1/3}}$$

With this choice, the error bound becomes:

$$f(n, n^{1/3}, n^{1/3}, u, u, u) = u(n^{1/3} + n^{1/3} + n^{1/3}) = 3n^{1/3}u$$

This is much better than the standard sequential summation bound of $nu$.

### Solution 4: Assigning Precisions

We want to assign fp64, fp32, and fp16 to $u_1$, $u_2$, $u_3$ to minimize the error bound:

$$f(n, b_1, b_2, u_1, u_2, u_3) = b_1u_1 + b_2u_2 + \frac{n}{b_1b_2}u_3$$

Let's denote:

- fp64: $u_{\text{high}} \approx 2^{-53}$
- fp32: $u_{\text{mid}} \approx 2^{-24}$
- fp16: $u_{\text{low}} \approx 2^{-11}$

The key observation is that the coefficients multiply the precisions:

- $u_1$ is multiplied by $b_1$ (small for reasonable block sizes)
- $u_2$ is multiplied by $b_2$ (small for reasonable block sizes)
- $u_3$ is multiplied by $\frac{n}{b_1b_2}$ (potentially large)

**Strategy:** Assign the highest precision (smallest unit roundoff) to the parameter with the largest coefficient.

Since typically $n \gg b_1, b_2$, we have $\frac{n}{b_1b_2} \gg b_1, b_2$.

Therefore, the optimal assignment is:

$$\boxed{u_3 = u_{\text{high}} \text{ (fp64)}, \quad u_2 = u_{\text{mid}} \text{ (fp32)}, \quad u_1 = u_{\text{low}} \text{ (fp16)}}$$

**Reasoning:**

- Use **fp16** for the many small block sums (level 1) since $b_1$ is small
- Use **fp32** for the intermediate sums (level 2) since $b_2$ is moderate
- Use **fp64** for the final sum (level 3) since $\frac{n}{b_1b_2}$ is large

This minimizes the overall error while using low precision where it's safe (many local operations) and high precision where it matters most (the final accumulation).

# 3 Variations

## Variation 1: Four-Level Hierarchy

**Problem:** Extend the algorithm to a four-level hierarchy with block sizes $b_1, b_2, b_3$ and precisions $u_1, u_2, u_3, u_4$. What is the error bound?

**Solution:**

The algorithm becomes:

1. Level 1: Compute $y_i = \sum_{j=1}^{b_1} x_{(i-1)b_1+j}$ in precision $u_1$
2. Level 2: Compute $z_i = \sum_{j=1}^{b_2} y_{(i-1)b_2+j}$ in precision $u_2$
3. Level 3: Compute $w_i = \sum_{j=1}^{b_3} z_{(i-1)b_3+j}$ in precision $u_3$
4. Level 4: Compute $\tilde{s} = \sum_{i=1}^{n/(b_1 b_2 b_3)} w_i$ in precision $u_4$

The error bound is:

$$\boxed{f(n, b_1, b_2, b_3, u_1, u_2, u_3, u_4) = b_1 u_1 + b_2 u_2 + b_3 u_3 + \frac{n}{b_1 b_2 b_3} u_4}$$

If all precisions are equal ($u_1 = u_2 = u_3 = u_4 = u$), the optimal block sizes are:

$$b_1 = b_2 = b_3 = n^{1/4}$$

giving an error bound of $4n^{1/4}u$.

## Variation 2: Arbitrary Number of Levels

**Problem:** Generalize to $L$ levels with block sizes $b_1, \ldots, b_{L-1}$ and precisions $u_1, \ldots, u_L$. What is the error bound?

**Solution:**

For an $L$-level hierarchy, the error bound is:

$$\boxed{f(n, b_1, \ldots, b_{L-1}, u_1, \ldots, u_L) = \sum_{i=1}^{L-1} b_i u_i + \frac{n}{\prod_{i=1}^{L-1} b_i} u_L}$$

If all precisions are equal ($u_1 = \cdots = u_L = u$), the optimal block sizes are:

$$b_1 = b_2 = \cdots = b_{L-1} = n^{1/L}$$

giving an error bound of:

$$f = u\left((L-1)n^{1/L} + n^{1/L}\right) = Ln^{1/L}u$$

As $L \to \infty$, we have $n^{1/L} \to 1$ and $Ln^{1/L} \to \infty$, so there's an optimal number of levels.

### Variation 3: Non-Uniform Block Sizes

**Problem:** Consider the case where we use different block sizes for different groups. For example, at level 1, blocks 1 to $k$ have size $b_1'$ and blocks $k+1$ to $n/b_1$ have size $b_1''$. How does this affect the error bound?

**Solution:**

The error bound becomes more complex. For an element $x_i$ in a block of size $b_1'$ at level 1:

$$|\theta_i| \leq b_1' u_1 + b_2 u_2 + \frac{n}{b_1' b_2} u_3$$

For an element $x_j$ in a block of size $b_1''$ at level 1:

$$|\theta_j| \leq b_1'' u_1 + b_2 u_2 + \frac{n}{b_1'' b_2} u_3$$

The overall error bound is:

$$\max\{b_1' u_1 + b_2 u_2 + \frac{n}{b_1' b_2} u_3, b_1'' u_1 + b_2 u_2 + \frac{n}{b_1'' b_2} u_3\}$$

### Variation 4: Comparison with Pairwise Summation

**Problem:** How does the three-level blocked summation compare with standard pairwise summation (binary tree)?

**Solution:**

**Pairwise summation:** Uses a binary tree with depth $\log_2 n$. If all operations use precision $u$, the error bound is:

$$f_{\text{pairwise}}(n, u) = \log_2(n) \cdot u$$

**Three-level blocked summation:** With optimal $b_1 = b_2 = n^{1/3}$ and uniform precision $u$:

$$f_{\text{blocked}}(n, n^{1/3}, n^{1/3}, u) = 3n^{1/3} u$$

**Comparison:**

- For $n = 10^9$:
  - Pairwise: $f \approx 30u$
  - Blocked: $f = 3 \cdot 10^3 u = 3000u$
- For $n = 10^6$:
  - Pairwise: $f \approx 20u$
  - Blocked: $f = 3 \cdot 10^2 u = 300u$

Pairwise summation has a much better error bound ($O(\log n)$ vs $O(n^{1/3})$).

However, the blocked approach has advantages:

- Better cache locality (processes data in blocks)
- Easier to parallelize
- Can use mixed precision effectively
- Simpler implementation

With mixed precision (fp16/fp32/fp64), the blocked approach becomes competitive with pure fp64 pairwise summation while being more efficient computationally.

### Variation 5: Optimal Level Count

**Problem:** For a given $n$ and uniform precision $u$, what is the optimal number of levels $L$ that minimizes the error bound?

**Solution:**

With $L$ levels and optimal block sizes $b_i = n^{1/L}$, the error is:

$$f(L) = Ln^{1/L}u$$

Taking the derivative with respect to $L$ (treating $L$ as continuous):

$$\frac{df}{dL} = u\left(n^{1/L} - \frac{\ln(n)}{L}n^{1/L}\right) = un^{1/L}\left(1 - \frac{\ln(n)}{L}\right)$$

Setting this to zero:

$$1 - \frac{\ln(n)}{L} = 0 \implies L = \ln(n)$$

Therefore, the optimal number of levels is approximately $L^* = \ln(n)$, which matches the depth of a binary tree (pairwise summation)!

This shows that with optimal parameters, a balanced tree structure (like pairwise summation) is optimal.