Sums of Many Terms

The City College of New York
CSc 30100 – Scientific Programming
Fall 2018 – Professor Erik K. Grimmelmann, Ph.D.

Sums with many terms

Often we are faced with computing sums with many terms such as

$$S = \sum_{i=0}^{n-1} x_i .$$

Two questions naturally arise:

- What errors should we exptect in the sum?
- What is the best algorithm to use in calculating the sum?

Assumptions and notation

We assume that the standard model of floating-point arithmetic applies.

$$fl\left(x \text{ op } y\right) = \left(x \text{ op } y\right)\left(1+\delta\right), \text{ where } \left|\delta\right| \le \varepsilon \text{ and op } = +-*/.$$

Our task is to evaluate

$$S_n = \sum_{i=0}^{n-1} x_i, \text{ where } x_0, x_1, \ldots, x_{n-1} \text{ are real numbers.}$$

The result can depend on the ordering of the x_i; for now we make no assumptions about the ordering.

Naïve algorithm

We start with the naive algorithm

```
>>>
s = 0
for i in range(0,n):
     s += x[i]
>>>
```

We use $\hat{S}_n$ to denote the sum of the first $n+1$ terms computed using floating-point arithmetic and E_n to denote the error of $\hat{S}_n$,

$$E_n = \hat{S}_n - S_n.$$

First two terms

$$S_0 = x_0$$

$$\hat{S}_0 = fl\left(x_0\right) = x_0\left(1+\delta_0\right) = S_0 + S_0\delta_0$$

$$E_0 = \hat{S}_0 - S_0 = S_0\delta_0$$

$$S_1 = S_0 + x_1 = x_0 + x_1$$

$$\hat{S}_1 = fl\left(\hat{S}_0 + x_1\right) = \left(S_0 + S_0\delta_0 + x_1\right)\left(1+\delta_1\right)$$

$$= \left(S_1 + S_0\delta_0\right)_1\left(1+\delta_1\right) = S_1 + \left(S_0\delta_0 + S_1\delta_1\right) + S_0\delta_0\delta_1,$$

$$E_1 = \hat{S}_1 - S_1 = \left(S_0\delta_0 + S_1\delta_1\right) + S_0\delta_0\delta_1,$$

Third term

$$S_2 = S_1 + x_2 = x_0 + x_1 + x_2$$

$$\hat{S}_2 = fl\left(\hat{S}_1 + x_2\right) = \left(\hat{S}_1 + x_2\right)\left(1+\delta_2\right)$$

$$= \left[S_1 + \left(S_0\delta_0 + S_1\delta_1\right) + S_0\delta_0\delta_1 + x_2\right]\left(1+\delta_2\right)$$

$$= \left[S_2 + \left(S_0\delta_0 + S_1\delta_1\right) + S_0\delta_0\delta_1\right]\left(1+\delta_2\right)$$

$$= S_2 + \left(S_0\delta_0 + S_1\delta_1\right) + S_0\delta_0\delta_1$$

$$+ S_2\delta_2 + \left(S_0\delta_0\delta_2 + S_1\delta_1\delta_2\right) + S_0\delta_0\delta_1\delta_2$$

$$E_2 = \hat{S}_2 - S_2 = \left(S_0\delta_0 + S_1\delta_1 + S_2\delta_2\right)$$

$$+ \left(S_0\delta_0\delta_1 + S_0\delta_0\delta_2 + S_1\delta_1\delta_2\right) + S_0\delta_0\delta_1\delta_2$$

General recursion expression

$$\hat{S}_k = fl\left(\hat{S}_{k-1} + x_k\right) = \left(\hat{S}_{k-1} + x_k\right)\left(1+\delta_k\right),$$

$$\text{where } \left[\delta_k\right] \le \varepsilon,\ \hat{S}_{k-1} \equiv 0,\ 0 \le k \le n-1.$$

Applying this recursion relationship repeatedly, we obtain

$$\hat{S}_{n-1} = \sum_{i=0}^{n-1} x_i \prod_{k=i}^{n-1}\left(1+\delta_k\right).$$

Since $\left|\delta_k\right| \le \varepsilon \ll 1$ for $0 \le k \le n-1,$ we can simplify the product,

$$\prod_{k=i}^{n-1}\left(1+\delta_k\right) = 1+\theta_{n-1},\ \text{where } \theta_{n-1} \le \frac{(n-1)\varepsilon}{1-(n-1)\varepsilon} \equiv \gamma_{n-1}.$$

Order-dependent error

Thus,

$$\hat{S}_{n-1} = \sum_{i=0}^{n-1} x_i \left(1+\theta_{n-i}\right)$$

which leads to

$$\left|E_n\right| = \sum_{i=0}^{n-1} x_i \left(1+\theta_{n-i}\right) \leq \sum_{i=0}^{n-1} \left|x_i\right| \gamma_{n-1}.$$

As was noted, this expression depends on the ordering of the x_i.

Order-independent error

We can weaken this expression and remove the dependency on the ordering and obtain

$$\left|E_n\right| \le \gamma_{n-2} \sum_{i=0}^{n-1} \left|x_i\right| = (n-2)\,\varepsilon \sum_{i=0}^{n-1} \left|x_i\right| + O\left(\varepsilon^2\right).$$

Relative error

Since

$$\left|E_n\right| \le \gamma_{n-2} \sum_{i=0}^{n-1} \left|x_i\right| = (n-2)\,\varepsilon \sum_{i=0}^{n-1} \left|x_i\right| + O\left(\varepsilon^2\right),$$

the relative error can be written as $\dfrac{\left|\hat{S}_n - S_n\right|}{\left|S_n\right|} \le \gamma_{n-2} \dfrac{\sum_{i=0}^{n-1} \left|x_i\right|}{\left|S_n\right|} = \gamma_{n-2} R_n,$

where $R_n \equiv \dfrac{\sum_{i=0}^{n-1} \left|x_i\right|}{\left|\sum_{i=0}^{n-1} x_i\right|}.$

R_n is referred to as the **condition number of summation**.

Condition number of summation

Since the condition number of summation, R_n is defined as

$$R_n \equiv \frac{\sum_{i=0}^{n-1} |x_i|}{\left|\sum_{i=0}^{n-1} x_i\right|},$$

it follows that $R_n \geq 1$. In fact, $R_n = 1$ only if all of the x_i are of the same sign.

Since the relative error is $\dfrac{\left|\hat{S}_n - S_n\right|}{\left|S_n\right|} \leq \gamma_{n-2} R_n,$

the smaller the value of $R_n,$ the lower the bound on the relative error.

Condition number of summation …

Since the relative error is $\dfrac{\left|\hat{S}_n - S_n\right|}{\left|S_n\right|} \le \gamma_{n-2} R_n,$

if $R_n = 1,$ the bound on the relative error of the sum is $O\left(n\varepsilon\right).$

A tighter error bound

Since $\hat{S}_k = fl\left(\hat{S}_{k-1} + x_k\right) = \left(\hat{S}_{k-1} + x_k\right)\left(1+\delta_k\right)$, where

$\left[\delta_k\right] \le \varepsilon$, $\hat{S}_{k-1} \equiv 0$, $0 \le k \le n-1$, we have

$$\left(\hat{S}_{k-1} + x_k\right)\delta_k = \hat{S}_k \frac{\delta_k}{\left(1+\delta_k\right)}.$$

Summing these individual errors we have

$$E_n = \sum_{k=1}^{n-1} \hat{S}_k \frac{\delta_k}{\left(1+\delta_k\right)}.$$

Thus, to first order, the overall error is the sum of the $n-1$ relative rounding errors weighted by the partial sums.

Choosing an ordering of the x_i

Thus, $E_n = \left|\hat{S}_n - S_n\right| \le \left(\frac{\varepsilon}{1-\varepsilon}\right)\sum_{k=0}^{n-1}\left|\hat{S}_k\right|.$

This bound involves the computed partial sums, but not the individual x_i.

One strategy would be to order the x_i so as to minimize

$\sum_{k=0}^{n-1}\left|\hat{S}_k\right|$, but this is would be very computationally expensive.

An alternative would be to minimize, in turn, $\left|x_0\right|, \left|\hat{S}_2\right|, \ldots, \left|\hat{S}_{n-2}\right|.$

This strategy is referred to as **Psum** and requires $O(n \log n)$ comparisons.

Compensated summation

Compensated summation was introduced by Kahan in 1965.

His method extended the work of Gill from 1951.

Compensation summation is recurrsive summation with a clever correction term designed to diminish rounding errors.

To display this method graphically, we use

x_1	x_2

to denote the mantissa of a floating point number *x*. We divide the mantissa into two portions; exactly where we divide it will depend on the circumstances. We seek to calculate $\hat{s} = fl(a+b) = fl(s)$.

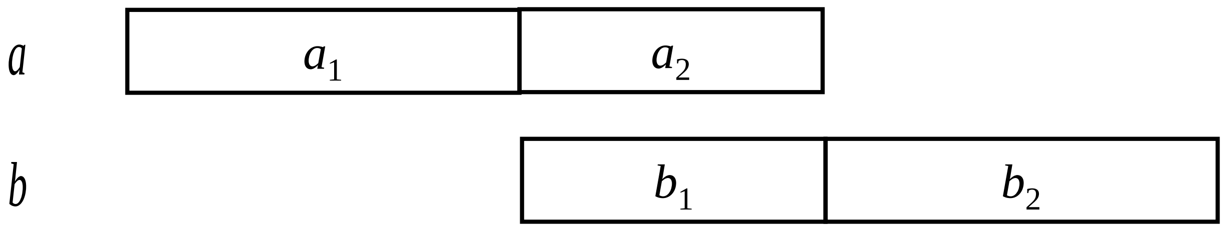
We assume for convenience that $a > b > 0$.

Compensated summation

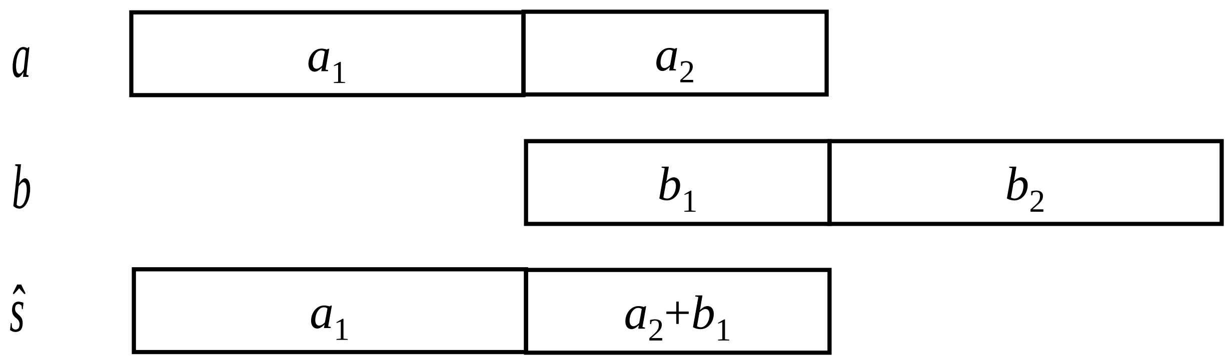
a $\boxed{\quad\quad\quad\quad a \quad\quad\quad\quad}$

b $\boxed{\quad\quad\quad\quad b \quad\quad\quad\quad}$

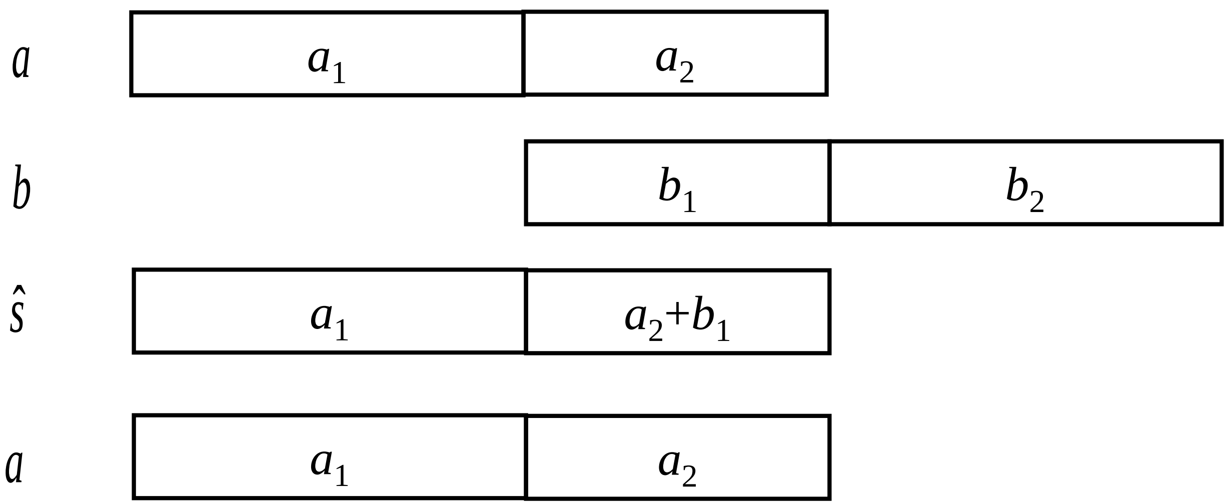
Compensated summation



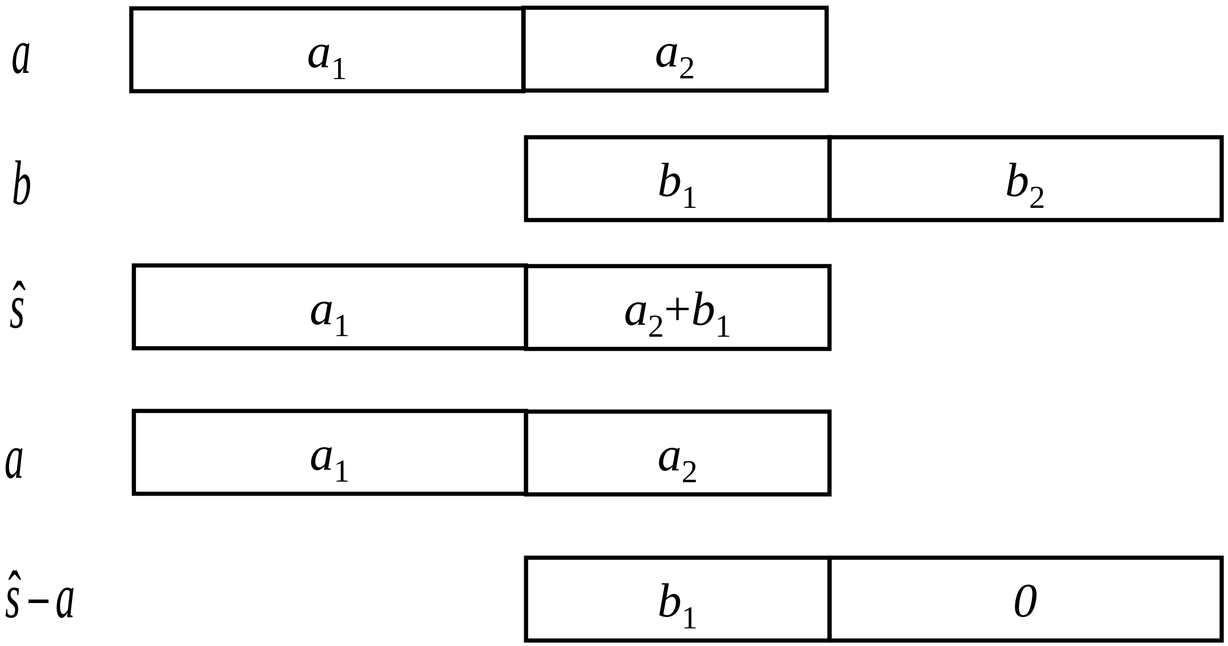
Compensated summation



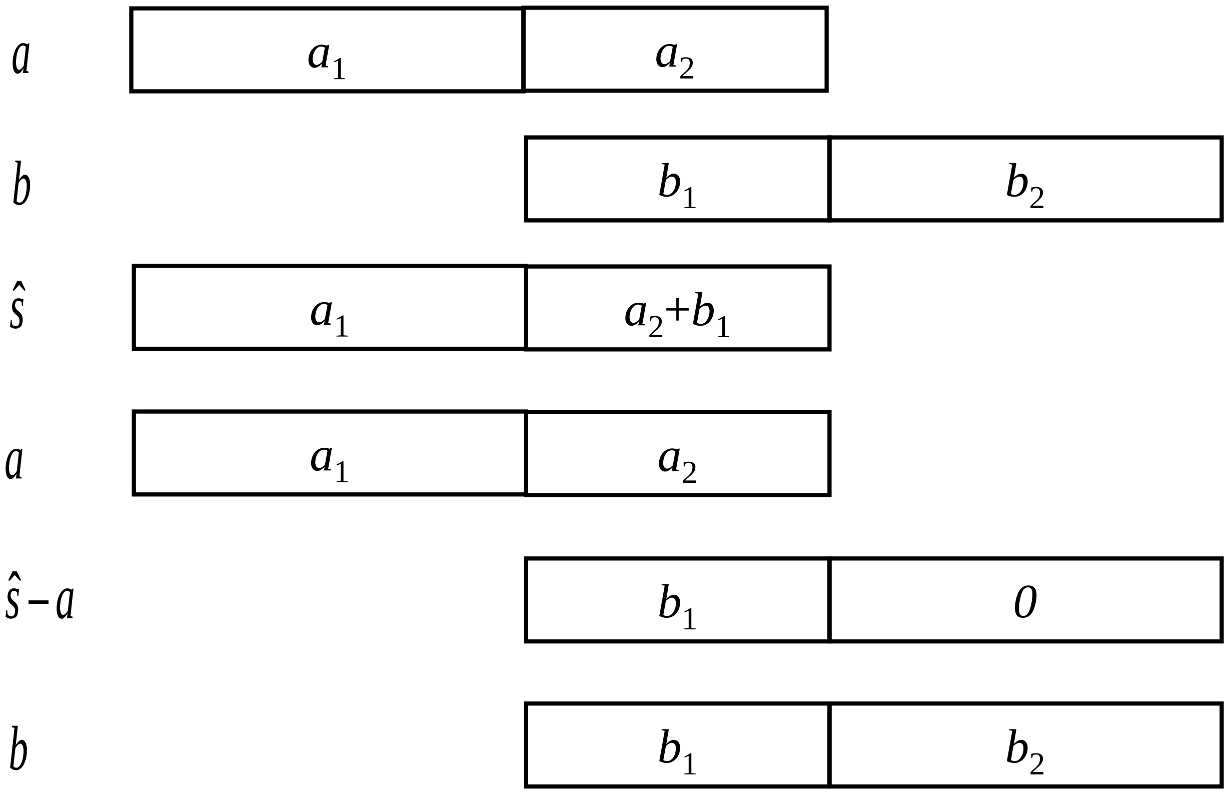
Compensated summation



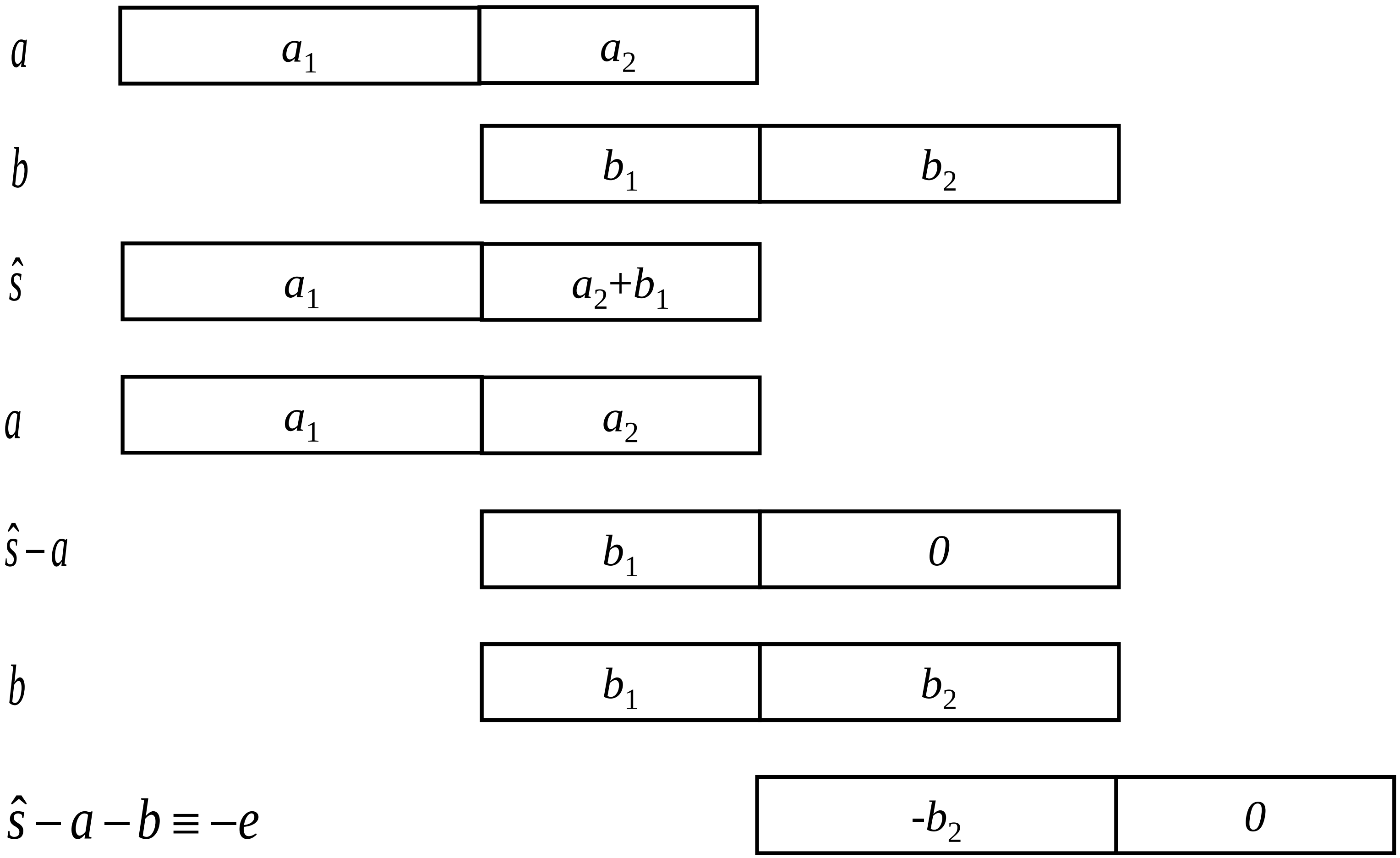
Compensated summation



Compensated summation



Compensated summation



Correction term

$$e = -\left\{\left[\left(a+b\right)-a\right]-b\right\} = \left(a-\hat{s}\right)+b$$

The correction term e is added to $\hat{s}$ before the next term in the sum is added.

Note that the rules of algebra would tell us that $e = 0$.

For rounded floating-point arithmetic in base 2,

$$a + b = \hat{s} + \hat{e}.$$

Thus, for base 2, $\hat{e}$ represents the error exactly.

Algorithm for compensated summation

```
>>>
s = 0
e=0

for i in range(0,n):
     temp =s
     y=x[i]+e
     s=temp+y
     e=(temp-s)+y
s+=e
>>>
```

Error bound for compensated addition

It can be shown that for compensated addition, the computed sum $\hat{S}_n$ satisfies

$$\hat{S}_n = \sum_{i=0}^{n-1} (1+\mu_i) x_i, \text{ where } |\mu_i| \le 2\varepsilon + O\left(n\varepsilon^2\right).$$

Error bounds and actual errors

Error bounds are just that, thorretical bounds on the errors that can occur.

Actual errors depend not only on the algorithm employed, but also on the data. Often, actual errors are significantly less than the theoretical bounds on those errors.