Sequential Summation

1 Error analysis

In this note, we derive an error bound for sequential summation of n numbers $x_1, \ldots, x_n$ using floating-point arithmetic. In exact arithmetic, the sequential summation algorithm can be expressed as a sequence

$$s_1 = x_1, \quad s_k = s_{k-1} + x_k, \quad k = 2, \ldots, n,$$

such that $s_n = x_1 + x_2 + \cdots + x_n$ is computed sequentially based on the recurrence relation $s_k = s_{k-1} + x_k$. Using our model for floating-point addition,

$$\mathrm{fl}(a+b) = (a+b)(1+\delta), \quad |\delta| \leq u,$$

where $u > 0$ is the unit round-off, the *computed* partial sums may be expressed as

$$\hat{s}_1 = x_1, \quad \hat{s}_k = \mathrm{fl}(\hat{s}_{k-1} + x_k) = (\hat{s}_{k-1} + x_k)(1+\delta_k), \quad k = 2, \ldots, n,$$

where $|\delta_k| \leq u$. Expanding the recursive definition of $\hat{s}_n$, we arrive at

$$\hat{s}_n = (x_1 + x_2) \prod_{k=2}^{n} (1+\delta_k) + \sum_{i=3}^{n} x_i \prod_{k=i}^{n} (1+\delta_k),$$

and if we define $\prod_{k=i}^{n}(1+\delta_k) = 1 + \theta_i$ for $i = 2, \ldots, n$, we have that

$$\begin{aligned} \hat{s}_n &= (x_1 + x_2)(1+\theta_2) + \sum_{i=3}^{n} x_i (1+\theta_i) \\ &= s_n + \theta_2 x_1 + \sum_{i=2}^{n} \theta_i x_i. \end{aligned}$$

We get an expression for the error $\hat{s}_n - s_n$ by subtracting s_n on both sides, i.e.,

$$\hat{s}_n - s_n = \theta_2 x_1 + \sum_{i=2}^{n} \theta_i x_i.$$

It follows that the absolute error satisfies the inequality

$$|\hat{s}_n - s_n| \leq |\theta_2||x_1| + \sum_{i=2}^{n} |\theta_i||x_i|.$$

Assuming that there exists a scalar $\hat{\theta}$ such that

$$|\theta_i| \leq \hat{\theta}, \quad i = 2, \ldots, n \tag{1}$$

we can obtain a simpler upper bound of the form

$$|\hat{s}_n - s_n| \leq \hat{\theta} \sum_{i=1}^{n} |x_i|.$$

Moreover, if $s_n \neq 0$, we may divide both sides of the inequality by $|s_n| = |\sum_{i=1}^{n} x_i|$, and this yields the following *relative* error bound

$$\frac{|\hat{s}_n - s_n|}{|s_n|} \leq \hat{\theta} \frac{\sum_{i=1}^{n} |x_i|}{|\sum_{i=1}^{n} x_i|}.$$

Notice that the right-hand side consists of two factors: the scalar $\hat{\theta}$ and the factor

$$\frac{\sum_{i=1}^{n} |x_i|}{|\sum_{i=1}^{n} x_i|}$$

which is the so-called *condition number* for summation. In the next section, we show that $\hat{\theta} = \frac{nu}{1-nu}$ satisfies (1) if $nu < 1$. Moreover, we will show that the somewhat simpler expression $\hat{\theta} = 1.06 \cdot nu$ satisfies (1) if $nu < 0.1$.

2 Bounding the relative error

We can derive an upper and a lower bound for $1 + \theta_i$ by assuming that each δ_k is equal to either u or $-u$ for all k, i.e.,

$$(1-u)^{n-i+1} \leq \underbrace{\prod_{k=i}^{n} (1+\delta_k)}_{1+\theta_i} \leq (1+u)^{n-i+1}. \tag{2}$$

It follows that

$$(1-u)^n < (1-u)^{n-i+1} \leq 1 + \theta_i \leq (1+u)^{n-i+1} < (1+u)^n, \quad i = 2, \ldots, n.$$

We now show that if $nu < 1$, these inequalities imply that

$$1 - nu < 1 + \theta_i < 1 + \frac{nu}{1-nu},$$

or equivalently,

$$-nu < \theta_i < \frac{nu}{1-nu}.$$

We begin by showing by induction that $1 - nu < (1-u)^n$ for $n \geq 2$. Indeed, for $n = 2$ we have

$$(1-u)^2 = 1 - 2u + u^2 > 1 - 2u.$$

Now suppose the inequality $1 - ku < (1-u)^k$ is true for some $k \geq 2$. It then follows that

$$\begin{aligned}
(1-u)^{k+1} &= (1-u)^k (1-u) \\
&> (1-ku)(1-u) \\
&= 1 - (k+1)u + ku^2 \\
&> 1 - (k+1)u
\end{aligned}$$

and hence $(1-u)^n > 1-nu$ for $n \geq 2$. Combining this result and (2) leads to the lower bound $-nu < \theta_i$.

The upper bound $\theta_i < \frac{nu}{1-nu}$ ($nu < 1$) follows from the two inequalities

$$e^t > 1+t,\ t \neq 0, \qquad e^{-t} > 1-t,\ t \neq 0.$$

The first inequality implies that

$$(1+u)^n < e^{nu},$$

and the second inequality implies that $e^t < 1/(1-t)$ if $0 < t < 1$, and hence

$$e^{nu} < \frac{1}{1-nu} = 1 + \frac{nu}{1-nu}, \quad nu < 1.$$

Combining these results and (2) leads to the upper bound $\theta_i < \frac{nu}{1-nu}$.

Fianlly, it follows from the above upper and lower bounds that

$$|\theta_i| < \max\left\{nu, \frac{nu}{1-nu}\right\} = \frac{nu}{1-nu}$$

for $nu < 1$.

3 Simplified bound

We now show that $|\theta_i| < 1.06 \cdot nu$ if $nu < 0.1$. Recall that the exponential function has the Maclaurin series

$$e^x = \sum_{k=1}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

and hence

$$1 + \theta_i < e^{nu} = 1 + nu + \frac{(nu)^2}{2!} + \frac{(nu)^3}{3!} + \cdots.$$

Subtracting 1 from both sides of the inequality and modifying the denominators in the series, we arrive at

$$\begin{aligned} \theta_i < e^{nu} - 1 &= nu\left(1 + \frac{nu}{2} + \frac{(nu)^2}{3!} + \cdots\right) \\ &< nu \underbrace{\left(1 + \frac{nu}{2} + \left(\frac{nu}{2}\right)^2 + \cdots\right)}_{\Sigma_{k=0}^{\infty}\left(\frac{nu}{2}\right)^k}. \end{aligned}$$

The right-hand side includes the sum of a geometric series which converges to

$$\sum_{k=0}^{\infty} \left(\frac{nu}{2}\right)^k = \frac{1}{1-\frac{nu}{2}}$$

provided that $nu < 2$. Assuming that $nu < 0.1$, we obtain the bound

$$\sum_{k=0}^{\infty} \left(\frac{nu}{2}\right)^k < \frac{1}{1-0.05} < 1.06,$$

which implies that $|\theta_i| < 1.06 \cdot nu$, i.e., the upper bound is a linear function of n.

4 Sequential summation in double precision

For double precision floating point numbers where $u = 2^{-53}$, the condition $nu < 0.1$ is satisfied whenever $n < 0.1 \cdot 2^{53} \approx 9 \cdot 10^{14}$. Storing this many double precision floating point numbers requires $8n$ bytes which is a whopping 7.2 petabytes (PB) of storage. Using a CPU that can do 300 billion floating-point operations per second (300 Gflops/s) in double precision, sequential summation of this many floating-point numbers will take approximately

$$\frac{9 \cdot 10^{14} \text{ flops}}{300 \cdot 10^{9} \text{ flops/s}} = 3{,}000 \text{ s}$$

or equivalently, 50 minutes. Of course, this requires that we can retrieve the data from where it is stored (RAM, hard drive, network, etc.) sufficiently fast. For example, if the memory bandwidth is 50 GB/s, it would take around 40 hours to retrieve the numbers from memory. In practice, current computers rarely have more than 10-100 GB of RAM, so we would typically have to retrieve the data from a hard drive or via a network interface. For a hard drive connected via SATA3, the upper limit on the bandwidth is around 600 MB/s (although current hard drives, including solid-state drives, are generally slower than that). Assuming that we can retrieve the numbers from a hard drive at a sustained rate of 300 MB/s, it would take more than around 40 weeks to do the aforementioned summation. Now, current hard drives are typically only a few terabytes big, so a more realistic scenario would be that the data is stored on disks across a network. Instead of retrieving the all the data from the network, one would most likely aim to distribute the computation of the sum across the network by computing partial sums locally and summing those at the end, effectively limiting the amount of data that we need to transfer. Note that this is no longer “sequential summation”, and in fact, an error analysis would yield a different worst-case bound than that of sequential summation. We will briefly return to this topic later in the course when we discuss parallelization.