PISUM: Summing a Slowly Convergent Series

Prof. W. Kahan
27 Jan. 1998

For any integer $M \geq 0$ and constant $h > 0$, let $F(M)$ be the sum of the infinite series

$$F(M) := \sum_{\text{odd } k > 2M} h/(k^2 - \tfrac{1}{4}) = \sum_{\text{odd } k > 2M} \left( h/(k - \tfrac{1}{2}) - h/(k + \tfrac{1}{2}) \right)$$

$$= 2h/(4M+1) - 2h/(4M+3) + 2h/(4M+5) - 2h/(4M+7) + \ldots$$

$$= h\pi/2 - 2h( 1/1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \ldots + 1/(4M-7) - 1/(4M-5) + 1/(4M-3) - 1/(4M-1) ).$$

( Although $F(0) = h\pi/2$, this series is a slow way to compute $\pi$ ; for a series that converges faster see V. Adamchik and S. Wagon " A Simple Formula for $\pi$ " in Amer. Math. Monthly v. 104 #9, Nov. 1997, pp. 852-855.)

$F(M)$ can only be approximated by a sum $F(M) - F(M+N)$ of some finite number $N$ of terms:

$$F(M) - F(M+N) = \sum_{1 \leq n \leq N} h/( (2(M+n)-1)^2 - \tfrac{1}{4} ) .$$

Then the remainder

$$F(M+N) = \sum_{\text{odd } k > 2M+2N} h/(k^2 - \tfrac{1}{4})$$

can be estimated with the aid of the following elementary observations:

$$2F(M) > \sum_{k > 2M} h/(k^2 - \tfrac{1}{4}) = h/( 2M+1 - \tfrac{1}{2} ) , \quad \text{and}$$

$$2F(M) < \sum_{k \geq 2M} h/(k^2 - \tfrac{1}{4}) = h/( 2M - \tfrac{1}{2}) .$$

In short,

$$h/(4M+1) < F(M) < h/(4M-1) .$$

The approximation

$$F(M) \approx r(M) := h/(4M)$$

is in error by less ( actually far less ) than

$$h/(4M-1) - h/(4M) = h/((4M-1)(4M)) = h/((4M-\tfrac{1}{2})^2 - \tfrac{1}{4}) .$$

Consequently the error in approximating $F(M+N)$ by $r(M+N)$ is rather tinier than the last term $h/((2M+2N-1)^2 - \tfrac{1}{4})$ added to $F(M)-F(M+N)$. Therefore we could stop the summation when that last term contributed negligibly to the sum, and then obtain $F(M) \approx F(M)-F(M+N) + r(M+N)$ with negligibly more additional error.

How big must $N$ be before the last term $h/((2M+2N-1)^2 - \tfrac{1}{4})$ adds a negligible amount to $F(M)$ ? That depends a little upon $h$. We shall use $h := (4M+1)(4M+3)$, so that $F(M) \approx h/(4M) = 4M+4 + 3/(4M)$, and choose $M := 2\hat{}n - 2$ to get $F(M) \approx 2\hat{}(n+2) - 4$ slightly smaller than a power of 2. Then half an ULP of $F(M)$ must be about $\tfrac{1}{4}\varepsilon F(M)$ where $\varepsilon := |3\cdot(4.0/3 - 1) - 1|$ is an ULP of numbers between 1 and 2. The last term added falls below $F(M)\varepsilon/4$ when $h/(2M+2N-1)^2 < \tfrac{1}{4}\varepsilon h/(4M)$, i.e. $N > 2\sqrt{(M/\varepsilon)} - M$, roughly.

Actually the approximation $F(M) \approx r(M) = h/(4M)$ is much closer than was deduced above. To get a better estimate of the error we consider the following easily proved inequalities between integrals:

If $f(k) > 0$, $f'(k) < 0$ and $f''(k) > 0$ for all $k \geq 0$, then

$$\tfrac{1}{2}f(M) + \int_M^\infty f(k)\,dk < F(M) := \sum_{k=M}^{\infty} f(k) < \int_{M-\frac{1}{2}}^\infty f(k)\,dk .$$

The bounds differ by

$$\int_{M-\frac{1}{2}}^{M} f(k)\,dk - \tfrac{1}{2}f(M) \approx -f'(M)/8 \text{ roughly.}$$

In our case we have $f(k) := h/((2k+1)^2 - \frac{1}{4}) = h/(2k+\frac{1}{2}) - h/(2k+3/2)$ and $-f'(M)/8 = \frac{1}{2}h(2M+1)/((2M+1)^2 - \frac{1}{4})^2 \approx \frac{1}{2}h/(2M+1)^3$ and

$$\int_{M-\frac{1}{2}}^\infty f(k)\,dk = \tfrac{1}{2}h \ln((4M+1)/(4M-1)) = h/(4M) + (1/3)h/(4M)^3 + \ldots$$

so $\quad r(M) - 11h/(192\,M^3) < F(M) < r(M) + h/(192\,M^3)$ .

. . . . . .

Euler's summation formula ( 23.1.32, p. 806 of "Handbook of Math. Functions ..." ed. by M. Abramowitz & Irene Stegun (1964) ) tells us

$$\sum_{k=M}^{\infty} f(k) = \int_{M-\frac{1}{2}}^\infty f(k)\,dk + f'(M-\tfrac{1}{2})/24 - 7\,f'''(M-\tfrac{1}{2})/5760 + \ldots .$$

$$= \tfrac{1}{2}h \ln((4M+1)/(4M-1)) - (hM/3)/(4M^2 - \tfrac{1}{4})^2 - O(1/M^5)$$

$$= h/(4M) - h/(4M)^3 + 5h/(4M)^5 - 61h/(4M)^7 + O(1/M^9)$$

so actually $F(M) = r(M) - h/(4M)^3 + O(1/M^5)$ , which means for big $M$ that the error in $r(M)$ is about 3/11 of the error bound.

By these means or others it is possible to show that asymptotically

$$F(M) \text{ --> } h \sum_{n>0} g[n]/(4M)^{(2n-1)} \quad \text{as} \quad M \text{ --> } \infty \text{ , where}$$

$$g[n] := \Big( 1 - \sum_{0<j<n} 4^{(n-j)}\, \text{Comb}\{2n-1,\, 2j-2\}\, g[j] \Big)/(2n-1) \ ;$$

| n : | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| g[n]: | 1 | -1 | 5 | -61 | 1385 | -50521 | 2702765 | -199360981 | 19391512145 |

g[10] = -2404879675441 and g[n] grows super-exponentially.

Because $h = (4M+1)(4M+3)$ , we then find that

$$F(M) \text{ --> } 4M + 4 + \tfrac{1}{2}/M - \tfrac{1}{4}/M^2 - 3/(4M)^3 + h \sum_{n>2} g[n]/(4M)^{(2n-1)} .$$

To keep the error committed by neglecting $h\,g[n]/(4M)^{(2n-1)}$ and subsequent terms below $\varepsilon/(4M)$ , which is about half an ULP of the fractional part of $F(M)$ , we must keep $M$ bigger than roughly

$$m(n) := \tfrac{1}{4}\,( g[n]/\varepsilon )^{(1/(2n-4))} .$$

. . . . . .

To keep the error bound for $F(M)-F(M+N) + r(M+N)$ below about $\frac{1}{4}\varepsilon F(M)$ we must choose $N$ to satisfy $11h/(192\ (M+N)^3) < \frac{1}{4}\varepsilon h/(4M)$ , which means roughly $N > (\frac{1}{2}M/\varepsilon)^{(1/3)} - M$ . This is less onerous than the previous estimate $N > 2\sqrt{(M/\varepsilon)} - M$ , but still pessimistic.

However, the errors due to rounded summation also accumulate significantly if not attenuated by Compensated Summation, or else by summing the series backwards (small terms first instead of big terms first). Compensated is more accurate than backward summation, and requires no advance knowledge of the number of terms to be added.

Compensated summation can interact with the criterion for terminating the summation process in such a way as to diminish substantially the number of terms summed. The simplest stopping criterion is

" Stop when $F(M)-F(M+N) = F(M)-F(M+N-1)$ ."

With ordinary summation this takes effect as soon as the last term added falls below half an ULP of $F(M)$ . With compensated summation, this criterion can take effect shortly after the last term added falls below one ULP and $F(M)-F(M+N-1)$ gets rounded up by almost half an ULP, because then $F(M)-F(M+N)$ gets rounded down by almost half an ULP to match $F(M)-F(M+N-1)$ . In short, with that simplest stopping criterion, compensated summation tends to stop with about 29% fewer terms than ordinary summation. But that criterion is too simple; it doesn't stop until after vastly too many terms have been added.

A criterion that stops after fewer terms could be something like

" Stop when $F(M)-F(M+N) + r(M+N) \geq F(M)-F(M+N-1) + r(M+N-1)$ ."

This comes to mind because $F(M)-F(M+N) + r(M+N)$ should be a monotone decreasing function of $N$ in the absence of roundoff. But when $N$ increases by 1 the decrease is a tiny fraction ( $\approx (18/11)/(M+N)$ ) of the error bound. To see why, consider

$$\begin{aligned} \text{LHS} - \text{RHS} &= \\ &= ( F(M)-F(M+N) + r(M+N) ) - ( F(M)-F(M+N-1) + r(M+N-1) ) \\ &= 2h/(4M+4N-3) - 2h/(4M+4N-1) + h/(4M+4N) - h/(4M+4N-4) \\ &= -24h/( (4M+4N)(4M+4N-1)(4M+4N-3)(4M+4N-4) ) \\ &\approx -(3h/32)/(M+N-\tfrac{1}{2})^4 < 0 \end{aligned}$$

and compare this with $0 < r(M+N) - F(M+N) < (11h/192)/(M+N)^3$ . Therefore the last criterion would stop much too soon. Moreover it costs too much because it needs $r(M+N)$ computed too often.

Let us apply a similar criterion after several terms have been added:

$$\begin{aligned} &( F(M)-F(M+N) + r(M+N) ) - ( F(M)-F(M+K) + r(M+K) ) \\ &\quad\approx -(3/32)( h/(M+K+\tfrac{1}{2})^4 + h/(M+K+3/2)^4 + \ldots + h/(M+N-\tfrac{1}{2})^4 ) \\ &\quad\approx -(3/32) \int_{M+K}^{M+N} h/x^4\ dx = -(1/32)( h/(M+K)^3 - h/(M+N)^3 ) , \\ &\quad\approx -(11/192)h/(M+N)^3 \quad \text{when} \quad N-K \approx 0.415\ (M+K) . \end{aligned}$$

In other words, if $F(M)-F(M+K) + r(M+K)$ is not accurate enough yet, add $N-K = 1+\text{INT}( 0.415\ (M+K) )$ many more terms to get an improved estimate $F(M)-F(M+N) + r(M+N)$ and stop when it is negligibly smaller than the previous estimate $F(M)-F(M+K) + r(M+K)$ . We might expect this to occur when $(11h/192)/(M+N)^3 \approx \frac{1}{4}\varepsilon h/(4M)$ , which makes $N$ of the same order of magnitude as $(\frac{1}{2}M/\varepsilon)^{(1/3)} - M$ . Unfortunately, we cannot know this until it has happened twice, which adds something of at least order $0.415\ M$ to $N$ . When $M$ is so big that $M > \sqrt{(\frac{1}{2}/\varepsilon)}$ the approximation $F(M) \approx r(M)$ is already accurate enough to preclude the need for summation.

## Compensated Summation:

When we try to compute the sum $s = y + x$ we actually get a rounded sum $S = y + x - \delta$ in which $\delta$ is the rounding error committed when $y + x$ was rounded to fit into the register that holds $S$. Here is a picture of the digits, assuming $|y| \geq |x|$ and eight sig. digits:

```
          yyyyyyyy
        +   xxxxxxxx
          ----------
          sssssssssδδ   =  true sum  s
          SSSSSSSS~~    =  rounded sum  S
```

There are ways to recover the digits of $\delta$ lost to roundoff. The simplest way, valid only for reasonably well rounded binary ( and IBM's /360-/370-/390 chopped hexadecimal arithmetics ) goes thus:

Ensure $|y| \geq |x|$ by swapping if necessary. Then compute in turn

$S := y + x$ rounded ; actually $S = y + x - \delta$ ;
$c := (y - S) + x$ rounded; actually $c = \delta$ .
DO NOT DISREGARD PARENTHESES !

No rounding error occurs during the computation of $c$. This can be proved by applying the following ...

Theorem: If $p$ and $q$ are floating-point numbers in the same floating-point format, and if $\frac{1}{2} \leq p/q \leq 2$, then $p-q$ is representable exactly in that floating-point format unless it underflows ( which cannot happen if arithmetic conforms to IEEE Standards 754 and 854 ).

The proof of this theorem is not difficult, and then its application to the computation of $c = \delta$ is just a matter of case study, the cases being $-1 \leq x/y \leq -\frac{1}{2}$, $-\frac{1}{2} < x/y < 0$, and $0 \leq y/x \leq 1$. The foregoing computation of $c = \delta$ malfunctions for correctly rounded decimal arithmetic, as on H-P calculators, which requires a more complicated method; the simplest method fails on examples like $y = x = 0.999...996$ . We shall stick with correctly rounded binary.

The recovery of rounding errors of summation becomes important when we have to compute a sum $sN := x0 + x1 + x2 + x3 + \ldots + xN$ for a huge number $N$ of terms $xj$. When $x1 > x2 > \ldots > xN > 0$, which is the simplest case and applicable to the infinite series studies above, the $N$ rounding errors can obscure almost $\frac{1}{2}N$ ULPs of $sN$ if nothing is done to compensate for them. Summing in reverse order, smaller terms first, accumulates less roundoff but ( in extremely unlikely worst cases) it can still accumulate to almost $\frac{1}{4}N$ ULPs. Compensated Summation gets rid of most of those $N$ rounding errors thus:

Replace the simple program

```
s := x0 ;  for j = 1, 2, ... N in turn do  s := s + xj ;
```

by

```
s := x0 ;  c := 0 ;
for  j = 1, 2, 3, ..., N  in turn do
   {  t := c + xj ;  olds := s ;
      s := t + olds ;  c := (olds - s) + t } ;
```

The rounding error that occurs in $t := c + xj$ can be ignored because it typically amounts to less than half an ULP of $xj$. The error in $s := t + olds$ is recovered in $c$ to be added the next time around.