Solutions for the "many digits" friendly competition

The MPFR team

12 October 2005

Note: in the original problems[1], it is requested to get 10^N digits according to the input parameter N. To simplify the analysis, we assume here that we ask N digits, so instead of taking values $N = 2, 3, 4, \ldots$, the parameter N will be $100, 1000, 10000, \ldots$

In the whole document, p denotes the working precision in bits, and θ represents a value such that $|\theta| \leq 2^{-p}$ (different occurrences of θ may represent different values).

Unless explicitely mentioned, we compute the correct output with rounding toward zero of the last digit.

Problem C01: Compute the first N decimal digits after the decimal point of $\sin(\tan(\cos 1))$. We have $\sin(\tan(\cos 1)) \approx 0.564$: the first N decimal digits after the decimal point match the first N mantissa digits.

We use a target decimal precision $M \geq N$, and a binary precision p. We compute $x = \circ(\cos 1)$, $y = \circ(\tan x)$, $z = \circ(\sin y)$, with all roundings to nearest. It is easy to see that for $p \geq 5$, we have $1/2 \leq x, y, z < 1$, thus all rounding errors are bounded by 2^{-p-1}. We can thus write $x = \cos 1 + \epsilon_x$ with $|\epsilon_x| \leq 2^{-p-1}$. It follows $y = \tan(\cos 1 + \epsilon_x) + \epsilon_y$ with $|\epsilon_y| \leq 2^{-p-1}$; we can write $\tan(\cos 1 + \epsilon_x) = \tan(\cos 1) + \frac{\epsilon_x}{1+\theta^2}$, thus the absolute error on y is bounded by $|\epsilon_x| + |\epsilon_y| \leq 2^{-p}$. Similarly, the error on z is bounded by $3 \cdot 2^{-p-1}$. With $p \geq 2 + M \frac{\log 10}{\log 2}$, we have $3 \cdot 2^{-p-1} < 1/2 \cdot 10^{-M}$.

Finally, we output the binary value z in decimal to M digits, with rounding to nearest. Since $1/2 \leq z < 1$, the last digit has weight 10^{-M}, thus the total error — including that on z and the output error — is bounded by 10^{-M}. Thus, unless the last $M - N$ digits of the output are all zero, we can decide the correct output to N digits, rounded toward zero.

Problem C02: Compute the first N decimal digits after the decimal point of $\sqrt{e/\pi}$. We have $\sqrt{\pi} \approx 0.930$.

Let $x = \circ(e)$, $y = \circ(\pi)$, $z = \circ(x/y)$, $t = \circ(\sqrt{z})$, with rounding to nearest. If we use a precision of p bits, we have $x = e(1+u)$, $y = \pi(1+v)$, $z = x/y(1+w)$, $t = \sqrt{z}(1+s)$, with $|u|, |v|, |w|, |s| \leq 2^{-p}$. Thus t can be written $\sqrt{e/\pi}(1+\theta)^{5/2}$ with $|\theta| \leq 2^{-p}$. For $p \geq 2$, $(1+\theta)^{5/2}$ can be written $1 + 3\varepsilon$ with $|\varepsilon| \leq 2^{-p}$. Thus $y = \sqrt{e/\pi}(1+3\varepsilon)$, and the absolute error is bounded by $3 \cdot 2^{-p}$.

[1] `http://www.cs.ru.nl/~milad/manydigits/`

Assume we output M digits of the approximation y, with $M \geq N$, with rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $3 \cdot 2^{-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 3 + M\frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem C03: Compute the first N decimal digits after the decimal point of $\sin((e+1)^3)$. We have $\sin((e+1)^3) \approx 0.909$. We use the following algorithm, with precision p and rounding to nearest:

```
$x = \circ(\exp 1)$
$y = \circ(x + 1)$
$z = \circ(y^2)$
$t = \circ(zy)$
$u = \circ(\sin t)$
```

We have with θ a generic value such that $|\theta| \leq 2^{-p}$: $x = e(1+\theta)$, $y = (x+1)(1+\theta) = (e(1+\theta)+1)(1+\theta) = (e+1)(1+\theta)^2$, $z = y^2(1+\theta) = (e+1)^2(1+\theta)^5$, $t = zy(1+\theta) = (e+1)^3(1+\theta)^8$.

For $p \geq 2$, we have $24 \leq t \leq 64$, and $(1+\theta)^8$ can be written $1 + 13\theta$ when $|\theta| \leq 2^{-p}$, so the absolute error on t is bounded by $13 \cdot 2^{6-p}$.

Thus $t = (e+1)^3 + r$ with $|r| \leq 13 \cdot 2^{6-p}$; $\sin t = \sin((e+1)^3) + r\cos a$ for some a, thus the final error, taking into account the rounding error on u, is bounded by $\frac{1}{2}\mathrm{ulp}(u) + 13 \cdot 2^{6-p} \leq 2^{-p-1} + 13 \cdot 2^{6-p} \leq 2^{10-p}$.

Thus for $2^{10-p} \leq \frac{1}{2}10^{-M}$, i.e. $p \geq 11 + M\frac{\log 10}{\log 2}$, the error on u is bounded by $\frac{1}{2}$ ulp of the output value.

Problem C04: Compute the first N decimal digits after the decimal point of $\exp(\pi\sqrt{2011})$. We have $\exp(\pi\sqrt{2011}) \approx 1.528 \cdot 10^{61}$, and $\exp(\pi\sqrt{2011}) \bmod 1 \approx 0.089$. Thus to get N digits after the decimal point, we must compute $N + 62$ mantissa digits.

We compute with precision $p \geq 11$ — so that 2011 can be represented exactly — and rounding to nearest:

```
$x = \circ(\pi)$
$y = \circ(\sqrt{2011})$
$z = \circ(xy)$
$t = \circ(\exp z)$
```

With θ a generic value such that $|\theta| \leq 2^{-p}$, we have $x = \pi(1+\theta)$, $y = \sqrt{2011}(1+\theta)$, thus $z = \pi\sqrt{2011}(1+\theta)^3$. For $p \geq 11$, $(1+\theta)^3$ can be written $1 + 3.002\theta$, thus the absolute error on z is bounded by $3.002\pi\sqrt{2011}2^{-p} \leq 423 \cdot 2^{-p}$.

We thus have $t = \exp(\pi\sqrt{2011})\exp(423\theta)(1+\theta)$; the expression $\exp(423\theta)(1+\theta)$ can be written $1 + 472\theta$, thus the total roundoff error is bounded by $472\exp(\pi\sqrt{2011})\theta \leq 2^{213-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 214 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: the program we used for the competition computes an interval enclosing the exact value using the MPFI library, and if that interval is small enough we can compute the output. This program is about 3.6 times slower than the MPFR-based variant we made after the competition. With the MPFR-based program, we would have got about 3.3 seconds instead of 11.6 for $N = 5$ on the competition machine, which would be faster than GINAC (4.4s) but still slower than Mathematica (2.05s).

Note: the program we used for the competition outputs in fact $N + 1$ digits when asked for N, so we should give $N - 1$ as argument to get N digits.

Problem C05: Compute the first N decimal digits after the decimal point of $\exp(\exp(\exp(1/2)))$. We have $\exp(\exp(\exp(1/2))) \approx 181.331$, thus we need to output $N+3$ digits. We use the following algorithm, with precision p and rounding to nearest:

```
$x = \circ(\exp \frac{1}{2})$
$y = \circ(\exp x)$
$z = \circ(\exp y)$
```

We have $x = e^{1/2}(1 + \theta)$, thus the absolute error on x is bounded by $e^{1/2}2^{-p}$; we write $x = e^{1/2} + e^{1/2}\theta$. We then have $y = \exp(e^{1/2}) \exp(e^{1/2}\theta)(1 + \theta)$, which can be written $y = \exp(e^{1/2})(1 + 2.8\theta)$ for $p \geq 5$. We then have

$$z = (\exp y)(1 + \theta) = \exp(\exp(e^{1/2})) \exp(2.8 \exp(e^{1/2})\theta)(1 + \theta),$$

which can be written $\exp(\exp(e^{1/2}))(1 + 21\theta)$. Thus the total roundoff error on z is bounded by $21\theta \exp(\exp(e^{1/2})) \leq 2^{12-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 13 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: the competition program used $p \geq 25 + M\frac{\log 10}{\log 2}$ instead.

Problem C06: Compute the first N decimal digits after the decimal point of $\operatorname{atanh}(1 - \operatorname{atanh}(1 - \operatorname{atanh}(1 - \operatorname{atanh}(1/\pi))))$. We have $\operatorname{atanh}(1 - \operatorname{atanh}(1 - \operatorname{atanh}(1 - \operatorname{atanh}(1/\pi)))) \approx 1.123$. We use the following algorithm:

```
$q = \circ(\pi)$
$r = \circ(1/q)$
$s = \circ(\operatorname{atanh} r)$
$t = \circ(1 - s)$
$u = \circ(\operatorname{atanh} t)$
$v = \circ(1 - u)$
$w = \circ(\operatorname{atanh} v)$
$x = \circ(1 - w)$
$y = \circ(\operatorname{atanh} x)$
```

We can check that for $p \geq 6$, we have $2 \leq q < 4$, $1/4 \leq r, s < 1/2$, $1/2 \leq t, u, x < 1$, $1/8 \leq v, w < 1/4$, $1 \leq y < 2$. More precisely, we have $t \leq 11/16$ and $x \leq 27/32$.

We have $q = \pi(1+\theta)$ with $|\theta| \leq 2^{-p}$, thus $r = 1/\pi(1+\theta)^2$, which can be written $1/\pi(1+3*\theta)$, thus the absolute error on r is at most 2^{-p}.

The error on s is at most $\frac{1}{2}\text{ulp}(s) + \text{err}(r)/(1-\alpha^2) \leq 2^{-p-2} + 2^{-p}/(1-(1/2)^2) \leq 2 \cdot 2^{-p}$.

The error on t is at most $\frac{1}{2}\text{ulp}(t) + \text{err}(s) \leq 2^{-p-1} + 2 \cdot 2^{-p} \leq 3 \cdot 2^{-p}$.

The error on u is at most $\frac{1}{2}\text{ulp}(u) + \text{err}(t)/(1-\alpha^2) \leq 2^{-p-1} + 3 \cdot 2^{-p}/(1-(11/16)^2) \leq 7 \cdot 2^{-p}$.

The error on v is at most $\frac{1}{2}\text{ulp}(v) + \text{err}(u) \leq 2^{-p-3} + 7 \cdot 2^{-p} \leq 8 \cdot 2^{-p}$.

The error on w is at most $\frac{1}{2}\text{ulp}(w) + \text{err}(v)/(1-\alpha^2) \leq 2^{-p-3} + 8 \cdot 2^{-p}/(1-(1/4)^2) \leq 9 \cdot 2^{-p}$.

The error on x is at most $\frac{1}{2}\text{ulp}(x) + \text{err}(w) \leq 10 \cdot 2^{-p}$.

Finally, the error on y is at most $\frac{1}{2}\text{ulp}(y) + \text{err}(x)/(1-\alpha^2) \leq 2^{-p} + 10 \cdot 2^{-p}/(1-(27/32)^2) \leq 36 \cdot 2^{-p} \leq 2(6-p)$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 7 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: the competition program used $p \geq 6 + M\frac{\log 10}{\log 2}$ instead, which may be not enough, but since it computed only $M-1$ digits after the decimal point, it was correct.

Problem C07: Compute the first N decimal digits after the decimal point of π^{1000}. We have $\pi^{1000} \approx 1.412 \cdot 10^{497}$, and $\pi^{1000} \bmod 1 \approx 0.967$, thus we need to compute $498 + N$ digits. We compute with precision p and rounding to nearest:

$x = \circ(\pi)$ $[\pi]$
$y = \circ(x^2)$ $[\pi^2]$
$y = \circ(y^2)$ $[\pi^4]$
$x = \circ(xy)$ $[\pi^5]$
$y = \circ(x^2)$ $[\pi^{10}]$
$y = \circ(y^2)$ $[\pi^{20}]$
$x = \circ(xy)$ $[\pi^{25}]$
$y = \circ(x^2)$ $[\pi^{50}]$
$y = \circ(y^2)$ $[\pi^{100}]$
$x = \circ(xy)$ $[\pi^{125}]$
$x = \circ(x^2)$ $[\pi^{250}]$
$x = \circ(x^2)$ $[\pi^{500}]$
$x = \circ(x^2)$ $[\pi^{1000}]$

where we've put in square brackets the corresponding approximations. (Is this an optimal addition chain for 1000?)

It can be seen by induction that each approximation of π^k can be written $\pi^k(1+\theta)^{2k-1}$ with $|\theta| \leq 2^{-p}$. Thus the final $x = \pi^{1000}(1+\theta)^{1999}$. For $p \geq 12$, $(1+\theta)^{1999}$ can be written $1 + 2577\theta$, thus the total roundoff error is bounded by $2577\pi^{1000}\theta \leq 2^{1663-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 1664 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: the competition program used $p \geq 1663 + M\frac{\log 10}{\log 2}$ only.

Problem C08: Compute the first N decimal digits after the decimal point of $\sin(6^{6^6})$. We have $\sin(6^{6^6}) \approx 0.953$. This is a difficult problem, indeed for small N. Indeed, the integer 6^{6^6} has 36306 digits, so to get N correct digits, we need to perform an argument reduction with accuracy $36306 + N$ digits! This explains why the system which could do that problem could do it for $N = 4$ at least.

We use the following method, with different precisions $q = 120606$ (this is one plus the number of bits of 6^{6^6}), $q + p$ and p:

```
$n \leftarrow 6^{6^6}$
$x \leftarrow 2 \circ_{q+p} (\pi)$
$y \leftarrow \circ_q(n)$ [exact]
$z \leftarrow \circ_q(y/x)$
$k \leftarrow \lfloor z \rceil$
$v \leftarrow \circ_{q+p}(kx)$
$s \leftarrow \circ_p(y - v)$
$t \leftarrow \circ_p(\sin s)$
```

We have $x = 2\pi(1+\theta)$ with $|\theta| \leq 2^{-q}$. Since $6^{6^6} < 2^q$, y is exactly n. Now $z = n/(2\pi)(1+\theta)^2$, and since $(1+\theta)^2$ can be written $1 + 3\theta$, the error on z is bounded by $3n/(2\pi)\theta \leq 1/8$. Thus k differs of at most $5/8$ from $n/(2\pi)$, i.e. $|n - 2k\pi| \leq 5\pi/4 < 4$.

Now $x = 2\pi(1 + \mu)$ with $|\mu| \leq 2^{-q-p}$, and $v = 2k\pi(1 + \mu)^2$, thus the error on v is at most $6k\pi\mu \leq 0.8 \cdot 2^{-p}$. Thus the error on s is at most $\frac{1}{2}\mathrm{ulp}(s) + 0.8 \cdot 2^{-p} \leq 2.8 \cdot 2^{-p}$ since $|s| < 4$. The total roundoff error on t is thus bounded by $\frac{1}{2}\mathrm{ulp}(t) + 2.8 \cdot 2^{-p} |\cos \alpha| \leq 3.3 \cdot 2^{-p} \leq 2^{2-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 3 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: the competition program used $q = 120640$ and $p \geq M\frac{\log 10}{\log 2}$, which is more than needed, but works.

Problem C09: Compute the first N decimal digits after the decimal point of $\sin(10 \operatorname{atan}(\tanh(\pi\sqrt{2011}/3)))$. We have $\sin(10 \operatorname{atan}(\tanh(\pi\sqrt{2011}/3))) \approx 0.999$, but the digits differ from 9 up from the 80th.

We compute with precision $p \geq 11$ — so that 2011 is exact — and rounding to nearest:

```
$q = \circ(\pi)$
$r = \circ(\sqrt{2011})$
$s = \circ(qr)$
$t = \circ(s/3)$
$u = \circ(\tanh t)$
$v = \circ(\operatorname{atan} u)$
$w = \circ(10v)$
$x = \circ(\sin w)$
```

For $p \geq 6$, we have $2 \leq q < 4$, $32 \leq r, t < 64$, $128 \leq s < 256$, $1/2 \leq u, v < 1$, $4 \leq w < 8$. More precisely, $45 \leq t \leq 48$, $63/64 \leq u \leq 1$, $61/8 \leq w \leq 8$, and $31/32 \leq \sin(61/8)$ thus $31/32 \leq x \leq 1$.

We have $q = \pi(1+\theta)$ with $|\theta| \leq 2^{-p}$, $r = \sqrt{2011}(1+\theta)$, $s = \pi\sqrt{2011}(1+\theta)^3$, $t = \pi sqrt 2011/3(1+\theta)^4$.

Since $(1+\theta)^4$ can be written $1+5\theta$, the absolute error on t is bounded by $\pi sqrt 2011/3(5\theta) \leq 235 \cdot 2^{-p}$. Thus $\mathrm{err}(u) \leq \frac{1}{2}\mathrm{ulp}(u) + \mathrm{err}(t)(1-\tanh(\alpha)^2) \leq 2^{-p-1} + 235 \cdot 2^{-p}(1-\tanh(45)^2) \leq 2^{-p-1} + 2^{-119-p} \leq 2^{-p}$.

Now $\mathrm{err}(v) \leq \frac{1}{2}\mathrm{ulp}(v) + \mathrm{err}(u)/(1+\alpha^2) \leq 2^{-p-1} + 2^{-p}/(1+(63/64)^2) \leq 2 \cdot 2^{-p}$.

And $\mathrm{err}(w) \leq \frac{1}{2}\mathrm{ulp}(w) + 10\mathrm{err}(v) \leq 2^{2-p} + 20 \cdot 2^{-p} \leq 24 \cdot 2^{-p}$.

Finally, $\mathrm{err}(x) \leq \frac{1}{2}\mathrm{ulp}(x) + \mathrm{err}(w)|\cos\alpha| \leq 2^{-p-1} + 24 \cdot 2^{-p} \leq 2^{5-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 6 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Problem C10: Compute the first N decimal digits after the decimal point of $(7+2^{1/5}-5\cdot 8^{1/5})^{1/3}+4^{1/5}-2^{1/5}$. This constant is equal to 1. If the question was to compute it with rounding to nearest, it would be impossible since if we output an approximation to M digits, with total error bounded by 1 ulp, this approximation will necessarily be $\underbrace{1.000\ldots000}_{M}$.

But the rules only ask for an absolute error of less than one ulp.

With $x = 2^{1/5}$, this expression is $(7+x-5*x^3)(1/3)+x^2-x$. We compute with precision p and rounding to nearest:

$$
\begin{aligned}
x &= \circ(2^{1/5})\\
y &= \circ(x^2)\\
z &= \circ(yx)\\
q &= \circ(7+x)\\
r &= \circ(5z)\\
s &= \circ(q-r)\\
t &= \circ(s^{1/3})\\
u &= \circ(t+y)\\
v &= \circ(u-x)
\end{aligned}
$$

(The values $\circ(2^{1/5})$ and $\circ(s^{1/3})$ are computed with the `mpfr_root` function.)

For $p \geq 10$, we have $1 \leq x, y, z < 2$, $8 \leq q < 16$, $4 \leq r < 8$, $1/2 \leq s, t < 1$, $2 \leq u < 4$, $1/2 \leq v \leq 2$. More precisely, $67/128 \leq s \leq 19/32$.

We have $x = 2^{1/5}(1+\theta)$, and the absolute error on x can be bounded by $2^{1/5}2^{-p} \leq 2(1-p)$.

Then $y = 2^{2/5}(1+\theta)^3$, thus $\mathrm{err}(y) \leq 2^{2/5}(4\theta) \leq 6 \cdot 2^{-p}$, and $z = 2^{3/5}(1+\theta)^5$.

Now $\mathrm{err}(q) \leq \frac{1}{2}\mathrm{ulp}(q) + \mathrm{err}(x) \leq 2^{3-p} + 2^{1-p} \leq 10 \cdot 2^{-p}$; $r = 5 \cdot 2^{3/5}(1+\theta)^6$. Since $(1+\theta)^6$ can be written $1+7\theta$, the absolute error on r is bounded by $5 \cdot 2^{3/5}(7\theta) \leq 54 \cdot 2^{-p}$.

Now $\mathrm{err}(s) \leq \frac{1}{2}\mathrm{ulp}(s) + \mathrm{err}(q) + \mathrm{err}(r) \leq 2^{-p-1} + 10 \cdot 2^{-p} + 54 \cdot 2^{-p} \leq 65 \cdot 2^{-p}$; $\mathrm{err}(t) \leq \frac{1}{2}\mathrm{ulp}(t) + \mathrm{err}(s)(1/3\alpha^{-2/3}) \leq 2^{-p-1} + 65 \cdot 2^{-p}(1/3(67/128)^{-2/3}) \leq 34 \cdot 2^{-p}$.

And $\mathrm{err}(u) \leq \frac{1}{2}\mathrm{ulp}(u) + \mathrm{err}(t) + \mathrm{err}(y) \leq 2^{1-p} + 34 \cdot 2^{-p} + 6 \cdot 2^{-p} \leq 42 \cdot 2^{-p}$.

Finally $\mathrm{err}(v) \leq \frac{1}{2}\mathrm{ulp}(v) + \mathrm{err}(u) + \mathrm{err}(x) \leq 2^{-p} + 42 \cdot 2^{-p} + 2^{1-p} \leq 45 \cdot 2^{-p} \leq 2^{6-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 7 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value.

Note: in March 2005, Torbjörn Granlund and Paul Zimmermann designed a new implementation of the `mpn_rootrem` function from GMP, on which `mpfr_root` is based. With this new implementation, $N = 5$ takes 768ms on an Athlon 1.7Ghz, and $N = 6$ takes 15.9s (among which 8.2s for the 5th root, and 2.6s for the 3rd root), which should correspond to 0.5s and 11.1s respectively on the competition machine.

Problem C11: Compute the first N decimal digits after the decimal point of $\tan(2^{1/2}) + \operatorname{atanh}(\sin 1)$. We have $\tan(2^{1/2}) + \operatorname{atanh}(\sin 1) \approx 7.560$, thus we need to output $N + 1$ digits.

We proceed as follows, with precision p and rounding to nearest:

$$\begin{aligned}
&x = \circ(\sqrt{2})\\
&y = \circ(\tan x)\\
&z = \circ(\sin 1)\\
&t = \circ(\operatorname{atanh} z)\\
&u = \circ(y + t)
\end{aligned}$$

For $p \geq 5$, we have $45/32 \leq x \leq 91/64$, $6 \leq y \leq 107/16$, $107/128 \leq z \leq 27/32$, $77/64 \leq t \leq 79/64$, and $115/16 \leq u \leq 127/16$.

We thus have $\operatorname{err}(x) \leq \frac{1}{2}\operatorname{ulp}(x) \leq 2^{-p}$, $\operatorname{err}(y) \leq \frac{1}{2}\operatorname{ulp}(y) + \operatorname{err}(x)(1 + \alpha^2)$ for $\alpha \in [45/32, 91/64]$, i.e. $\operatorname{err}(y) \leq 2^{2-p} + 2^{-p}(1 + (91/64)^2) \leq 7.1 \cdot 2^{-p}$. Then $\operatorname{err}(z) \leq \frac{1}{2}\operatorname{ulp}(z) = 2^{-p-1}$, $\operatorname{err}(t) \leq \frac{1}{2}\operatorname{ulp}(t) + \operatorname{err}(z)/(1-\beta^2)$ for $\beta \in [107/128, 27/32]$, i.e. $\operatorname{err}(t) \leq 2^{-p} + 2^{-p-1}/(1-(27/32)^2) \leq 2.8 \cdot 2^{-p}$.

Finally $\operatorname{err}(u) \leq \frac{1}{2}\operatorname{ulp}(u) + \operatorname{err}(y) + \operatorname{err}(t) \leq 2^{2-p} + 7.1 \cdot 2^{-p} + 2.8 \cdot 2^{-p} = 13.9 \cdot 2^{-p} \leq 2^{4-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 5 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value. (The competition program used $p \geq 6 + M\frac{\log 10}{\log 2}$.)

For $N = 5$, the tangent takes 74% of the time, the sine 13% and the hyperbolic arctangent 14%.

Problem C12: Compute the first N decimal digits after the decimal point of $\operatorname{asin}(1/e^2) + \operatorname{asinh}(e^2)$. We have $\operatorname{asin}(1/e^2) + \operatorname{asinh}(e^2) \approx 2.833$, so we need to output $N + 1$ digits.

We proceed as follows, using the identity $\operatorname{asinh} t = \log(t + \sqrt{t^2 + 1})$. We thus get $\operatorname{asinh}(e^2) = \log(e^2 + \sqrt{e^4 + 1}) = 2 + \log(1 + \sqrt{1 + e^{-4}})$. Therefore replacing $\operatorname{asinh}(e^2)$ by $\log(1 + \sqrt{1 + e^{-4}})$ will give the same digits after the decimal point, and moreover without the integer part "2":

$$\begin{aligned}
&x = \circ(\exp(-2))\\
&y = \circ(x^2)\\
&z = \circ(1 + y)\\
&t = \circ(\sqrt{z})\\
&u = \circ(\mathrm{log1p}\, t)
\end{aligned}$$

$v = \circ(\operatorname{asin} x)$
$w = \circ(u + v)$

All quantities are positive. For $p \geq 4$, we have $x, v < 2^{-2}$, $y < 2^{-5}$, $1 \leq z, t < 2$, $u, w \leq 1$. We deduce $x = e^{-2}(1+\theta)$, $y = e^{-4}(1+\theta)^3$, $z = (1+e^{-4})(1+\theta)^4$, $t = \sqrt{1+e^{-4}}(1+\theta)^3$, thus the absolute error on t is bounded by $\sqrt{1+e^{-4}}(3.2\theta) \leq 3.3 \cdot 2^{-p}$. Now $\operatorname{err}(u) \leq \frac{1}{2}\operatorname{ulp}(u) + \operatorname{err}(t)/(1+\alpha)$ for $\alpha \in [1, 2]$, i.e. $\operatorname{err}(u) \leq 2^{-p-1} + 3.3 \cdot 2^{-p}/2 \leq 2.2 \cdot 2^{-p}$.

Then $\operatorname{err}(v) \leq \frac{1}{2}\operatorname{ulp}(v) + \operatorname{err}(x)(1-\beta^2)^{-1/2}$ for $\beta \in [1/8, 1/4]$, i.e. $\operatorname{err}(v) \leq 2^{-p-3} + e^{-2}2^{-p}(1-(1/4)^2)^{-1/2} \leq 0.27 \cdot 2^{-p}$.

Finally $\operatorname{err}(w) \leq \frac{1}{2}\operatorname{ulp}(w) + \operatorname{err}(u) + \operatorname{err}(v) \leq 2^{-p-1} + 2.2 \cdot 2^{-p} + 0.27 \cdot 2^{-p} \leq 2^{2-p}$.

Thus if we output $M \geq N$ digits after the decimal point, and $p \geq 3 + M\frac{\log 10}{\log 2}$, the output value differs by at most one ulp from the exact value. (The competition program used $p \geq 2 + M\frac{\log 10}{\log 2}$.)

Problem C13: Compute the first N decimal digits after the decimal point of the Nth term of the logistic map. The logistic map is defined by $x_0 = 1/2$, and

$$x_{n+1} = 3.999 x_n(1 - x_n).$$

We compute it as follows:

$t_n = \circ(1 - x_n)$
$u_n = \circ(x_n t_n)$
$v_n = \circ(3999 u_n)$
$x_{n+1} = \circ(v_n/1000)$

For $p \geq 12$, $2047/2048 \leq x_1 \leq 4095/4096$ and $1023/1048576 \leq x_2 \leq 4095/2097152 \approx 0.00195$. For $1023/1048576 \leq x_n \leq 1/2$, it can be checked by interval analysis with $p = 12$ that $2045/1048576 \leq x_{n+1}$, thus we deduce $1023/1048576 \leq x_n$ for all n. For $x_n \geq 1/2$, then t_n is exact by Sterbenz theorem, thus $u_n = \circ(x_n(1-x_n))$; since $x(1-x) \leq 1/4$ for $0 \leq x \leq 1$, it follows $u_n \leq 1/4$, $v_n \leq 3999/4$, and $x_{n+1} \leq \operatorname{up}(3999/4000) \leq 4095/4096$. We deduce $1023/1048576 \leq x_n \leq 4095/4096$ for all n, $0 \leq t_n < 1$, $0 \leq u_n \leq 1/4$, and $0 \leq v_n \leq 3999/4$.

Let ϵ_n be the absolute error on x_n, and τ_n the rounding error on t_n, i.e. $t_n = 1 - x_n + \tau_n$. The absolute error on t_n is at most $\epsilon_n + \tau_n$, and that on u_n is at most $\frac{1}{2}\operatorname{ulp}(u_n) + \epsilon_n t_n + x_n(\epsilon_n + \tau_n)$; replacing t_n by $1 - x_n + \tau_n$, we get $2^{-p-3} + \epsilon_n + (x_n + \epsilon_n)\tau_n$. Since $\tau_n \leq \frac{1}{2}\operatorname{ulp}(t_n) \leq 2^{-p-1}$ and $x_n + \epsilon_n \leq 4095/4096$ — remember the exact value for x_n lies in the interval $[x_n - \epsilon_n, x_n + \epsilon_n]$ —, the error on u_n is bounded by $2^{-p-3} + \epsilon_n + \frac{4095}{4096}2^{-p-1} \leq \epsilon_n + \frac{5119}{8192}2^{-p}$.

The error on v_n is bounded by $\frac{1}{2}\operatorname{ulp}(v_n) + 3999(\epsilon_n + \frac{5119}{8192}2^{-p}) \leq 3999\epsilon_n + \frac{24665185}{8192}2^{-p}$. Finally, the error on x_{n+1} is bounded by

$$\epsilon_{n+1} \leq 2^{-p-1} + \frac{3999}{1000}\epsilon_n + \frac{24665185}{8192000}2^{-p} = \frac{3999}{1000}\epsilon_n + \frac{5752237}{1638400}2^{-p}.$$

Writing the recurrence $\epsilon_{n+1} = \alpha\epsilon_n + \beta$, and defining $\tau_n = \epsilon_n + c$ with $c = \alpha c - \beta$, we have $\tau_{n+1} = \alpha\tau_n$, thus $\tau_n = \alpha^n\tau_0$, or

$$\tau_n = 28761185/24567808 \cdot 2^{-p}[(3999/1000)^n - 1] \leq 2^{1-p}(3999/1000)^n.$$

Choose $M \geq N$. If $2^{1-p}(3999/1000)^N \leq \frac{1}{2}10^{-M}$, i.e. $p \geq M\frac{\log 39.99}{\log 2} + 2$, then the M-digit decimal output of x_N lies within one ulp of the corresponding exact value. (The competition solution used +3 instead of +2.) We have just to take care to add zeroes if needed before the floating-point mantissa of x_N if needed; indeed, x_N can be as small as $1023/1048576 \approx 0.000975$. For example, $x_{922} \approx 0.00258$.

For $N = 10^4$, and $M = N + 10$, we get $p = 53271$ and the result is $354 \ldots 324$; in fact, a precision of 42978 seems enough to get a correct result.

Remark: we now analyze a faster method, where we compute $x(1-x)$ by $x - x^2$ instead, which replaces one product by one square:

$$\begin{aligned}
&t_n = \circ(x_n^2)\\
&u_n = \circ(x_n - t_n)\\
&v_n = \circ(3999u_n)\\
&x_{n+1} = \circ(v_n/1000)
\end{aligned}$$

We first show by induction that for $p \geq 12$, $0 \leq t_n \leq 1$, $0 \leq u_n \leq 1/4$, $0 \leq v_n \leq 3999/4$, and $0 \leq x_{n+1} \leq 4095/4096$. Let τ_n be the error on t_n: $t_n = x_n^2 + \tau_n$. By induction, $|t_n| \leq 1$ since $|x_n| \leq 1$, thus $|\tau_n| \leq 2^{-p-1}$. We thus have $u_n = \circ(x_n - x_n^2 - \tau_n)$. If $x_n = 1/2$, then $\tau_n = 0$ and $u_n = 1/4$; otherwise either $x_n \leq 1/2 - 2^{-p-1}$ or $x_n \geq 1/2 + 2^{-p}$. In the former case, $x_n - x_n^2 \leq 1/4 - 2^{-2p-4}$, but since in that case $|t_n| \leq 1/4$, we have $|\tau_n| \leq 2^{-p-3}$, thus $x_n - x_n^2 - \tau_n \leq 1/4 - 2^{-2p-4} + 2^{-p-3}$, which implies that $u_n = \circ(x_n - x_n^2 - \tau_n) \leq 1/4$. In the later case, where $x_n \geq 1/2 + 2^{-p}$, we further distinguish two cases: either $x_n \geq \sqrt{2}/2$, in which case $x_n - x_n^2 \leq \sqrt{2}/2 - 1/2 \leq 0.21$, thus trivially $u_n \leq 1/4$; or $x_n < \sqrt{2}/2$, in which case the $|\tau_n| \leq 2^{-p-2}$, thus $x_n - x_n^2 - \tau_n \leq 1/4 - 2^{-2p} + 2^{-p-2}$, since the midpoint between $1/4$ and the next representable number is $1/4 + 2^{-p-2}$, again $u_n \leq 1/4$. The rest follows since $3999/4$ is exactly representable with $p \geq 12$ bits, and $3999/4000 \leq 4095/4096$, which is also representable.

Let $\hat{x}_n$ be the value of x_n we would get if computed with infinite precision, and ϵ_n be the corresponding error: $x_n = \hat{x}_n + \epsilon_n$. We have $t_n = x_n^2 + \tau_n$ and $u_n = x_n - t_n + \nu_n$, where $|\nu_n| \leq 2^{-p-3}$ is the rounding error on u_n. It follows $u_n = \hat{x}_n + \epsilon_n - [(\hat{x}_n + \epsilon_n)^2 + \tau_n] + \nu_n = \hat{x}_n - \hat{x}_n^2 + \epsilon_n(1 - 2\hat{x}_n - \epsilon_n) + \nu_n - \tau_n$. The expression $1 - 2\hat{x}_n - \epsilon_n$ can also be written $1 - \hat{x}_n - x_n$. Since both x_n and $\hat{x}_n$ are in $[0, 1]$, it follows $\mathrm{err}(u_n) \leq \epsilon_n + \frac{5}{8}2^{-p}$.

The rest of the analysis proceeds as above: $\mathrm{err}(v_n) \leq \frac{1}{2}\mathrm{ulp}(v_n) + 3999\mathrm{err}(u_n) \leq 2^{9-p} + 3999\epsilon_n + \frac{19995}{8}2^{-p} = 3999\epsilon_n + \frac{24091}{8}2^{-p}$. Now $\mathrm{err}(x_{n+1}) \leq \frac{1}{2}\mathrm{ulp}(x_{n+1}) + \mathrm{err}(v_n)/1000$, which yields

$$\epsilon_{n+1} \leq \frac{3999}{1000}\epsilon_n + \frac{28091}{8000}2^{-p}. \tag{1}$$

It follows $\epsilon_n \leq \frac{28091}{23992}(\frac{3999}{1000})^n 2^{-p} \leq 2^{1-p}(\frac{3999}{1000})^n$.

By looking at Equation (1), we see that when n increases, ϵ_n becomes much bigger than the term $\frac{28091}{8000}2^{-p}$. In other terms, the last bits of x_n are completely wrong, and there is no reason to round it to p bits. So the idea is to adjust the precision at step n, say p_n, so that the term 2^{-p} is of the same order of magnitude as ϵ_n. For example since we know that each iteration looses about $\alpha = \frac{\log 3.999}{\log 2}$ bits, we can take $p_n = p - n\alpha$. Replacing p by p_n in Equation (1), we get:

$$\epsilon_{n+1} \leq \frac{3999}{1000}\epsilon_n + \frac{28091}{8000}(\frac{3999}{1000})^n 2^{-p},$$

which admits as solution:

$$\epsilon_n = \frac{28091}{31992}n2^{-p}(\frac{3999}{1000})^n.$$

For $M \geq N$, it thus suffices to have $p \geq M\frac{\log 39.99}{\log 2} + \frac{logM}{\log 2} + 1$.

That new version takes 5.8s for $N = 4$ — and 1407 seconds for $N = 5$ — on the competition machine (instead of 9.1s for the original code we used): this is slightly better than the IRRAM time of 6.0s.

Problem C14: Compute the first N decimal digits after the decimal point of a_{100N}. The sequence a_n is defined by $a_0 = 14/3$, $a_1 = 5$, $a_2 = 184/35$,

$$a_{n+2} = 114 - \frac{1463 - \frac{6390 - \frac{9000}{a_{n-1}}}{a_n}}{a_{n+1}}.$$

We have in fact $a_n = \frac{6^{n+1}+5^{n+1}+3^{n+1}}{6^n+5^n+3^n}$. Thus $a_{100N} = 6 + O((5/6)^{100N})$. This means that since we print N digits, the answer should be either $999\ldots999$ or $000\ldots000$.

The program we used for the competition is heuristic, in the sense that we gave it the working precision: for $N = 10$ we gave precision of 3752 bits, and for $N = 100$ a precision of 37892 bits. It appears in fact that for $N = 10$ we need at least 4100 bits: for 3752 bits we get 0000000000 as output but the integer part is 100! Similarly for $N = 100$ we need a precision of at least 40933 bits.

We compute an approximation of a_n as follows:

$$\begin{aligned}
b_n &= \circ(9000/a_{n-1})\\
c_n &= \circ(6390 - b_n)\\
d_n &= \circ(c_n/a_n)\\
e_n &= \circ(1463 - d_n)\\
f_n &= \circ(e_n/a_{n+1})\\
a_{n+2} &= \circ(114 - f_n)
\end{aligned}$$

We assume for the rounded a_n: $a_0 < a_1 < a_2 < \cdots < a_n < 6$, all computed quantities are positive, and $b_n \leq 2^{11}$, $c_n \leq 2^{13}$, $d_n \leq 2^{10}$, $e_n \leq 2^{10}$, $f_n \leq 2^8$, and $14/3 \leq a_n \leq 6$.[2]

[2] Assuming $\circ_{\mathrm{down}}(14/3) \leq a_{n-1} \leq 6$, $5 \leq a_n \leq 6$ and $\circ_{\mathrm{down}}(184/35) \leq a_{n+1} \leq 6$, we can prove for $p \geq 10$ that $1500 \leq b_n \leq 1930$, $4456 \leq c_n \leq 4896$, $742 \leq d_n \leq 980$, $483 \leq e_n \leq 721$, $161/2 \leq f_n \leq 275/2$, which proves that $e_n = \circ(1463 - d_n)$ is exact, and $a_{n+2} = 114 - f_n$ is exact.

Let ϵ_n be the absolute error on a_n. We thus have $\mathrm{err}(b_n) \leq 2^{10-p} + \epsilon_{n-1}\frac{9000}{\theta^2}$ for $\theta \in [14/3, 6]$, thus $\mathrm{err}(b_n) \leq 2^{10-p} + \frac{20250}{49}\epsilon_{n-1}$; $\mathrm{err}(c_n) \leq 2^{12-p} + \mathrm{err}(b_n) \leq 5120 \cdot 2^{-p} + \frac{20250}{49}\epsilon_{n-1}$; $\mathrm{err}(d_n) \leq 2^{9-p} + \mathrm{err}(c_n)/a_n + c_n\mathrm{err}(a_n)/\theta'^2$ with $\theta' \in [5, 6]$, thus $\mathrm{err}(d_n) \leq 1536 \cdot 2^{-p} + \frac{4050}{49}\epsilon_{n-1} + \frac{4896}{25}\epsilon_n$; $\mathrm{err}(e_n) \leq \mathrm{err}(d_n)$ since e_n is exact by Sterbenz theorem; $\mathrm{err}(f_n) \leq 2^{7-p} + \mathrm{err}(e_n)/a_{n+1} + e_n\mathrm{err}(a_{n+1})/\theta''^2$ with $\theta'' \in [\circ_{\mathrm{down}}(184/35), 6]$, thus $\mathrm{err}(f_n) \leq \frac{2944}{7}2^{-p} + \frac{5400}{343}\epsilon_{n-1} + \frac{6528}{175}\epsilon_n + \frac{1648}{63}\epsilon_{n+1}$; and finally

$$\epsilon_{n+2} \leq \frac{2944}{7}2^{-p} + \frac{5400}{343}\epsilon_{n-1} + \frac{6528}{175}\epsilon_n + \frac{1648}{63}\epsilon_{n+1}.$$

It follows that

$$\epsilon_n \leq 0.0278\alpha^n 2^{-p},$$

where α is the real root of $77175x^3 - 2018800 * x^2 - 2878848 * x - 1215000$, i.e. $\alpha \approx 27.534$. This means that each iteration looses $\frac{\log \alpha}{\log 2} \approx 4.783$ bits, so to get a_{100N} with accuracy 10^{-N} we need to take $p \geq 478.3N + N\frac{\log 10}{\log 2} \approx 482N$.

Problem C15: Compute the first N decimal digits after the decimal point of $h(10N)$ where $h(n) = 1/n + \cdots + 1/n^2$. This is a more difficult problem than the corresponding practice problem which was $h(n) = 1 + 1/2 + \cdots + 1/n$, i.e. the usual harmonic number. Indeed, summing n^2 terms is not efficient enough, even with the help of binary splitting.

Let $n = 10N$. According to (6.3.2) in [1], we have:

$$\psi(n) = -\gamma + \sum_{k=1}^{n-1} 1/k,$$

thus $h(n) = \psi(n^2 + 1) - \psi(n)$ and we are reduced to the problem of computing $n/10$ digits of $\psi(n^2 + 1) - \psi(n)$.

Formula (6.3.18) from [1] gives:

$$\psi(z) \approx \log z - 1/(2z) - \sum_{k=1}^{\infty} \frac{B_{2k}}{2kz^{2k}}.$$

Since we have $|B_{2k}| \leq 2(2k)!/(2\pi)^{2k}/(1-2^{1-2k})$ [1, Eq. (23.1.15)], we deduce $|B_{2k}/(2kz^{2k})| \leq 2/k(2k)!/(2\pi z)^{2k}$ for $k \geq 1$. Using $n! \leq (n/e)^n\sqrt{8n}$ which is true for $n \geq 1$, we get

$$|B_{2k}/(2kz^{2k})| \leq 8(k/(e\pi z))^{2k}.$$

The minimum of $(k/(e\pi z))^{2k}$ is obtained for $k \approx \pi z$, with a value of $e^{-2k} \approx e^{-2\pi z}$. Thus to get a sufficient accuracy, we need that z is large enough. Here we have for $\psi(n^2 + 1)$: $z = n^2 + 1$ and we want $n/10$ digits, thus this is ok.

Let $R(z) = \sum_{k=1}^{\infty} \frac{B_{2k}}{2kz^{2k}}$. We have $\psi(n^2+1) \approx log(n^2+1) - 1/(2n^2+2) - R(n^2+1)$, thus

$$h(n) \approx log(n^2 + 1) - 1/(2n^2 + 2) - R(n^2 + 1) + \gamma - \sum_{k=1}^{n-1} 1/k.$$

The sum $\sum_{k=1}^{n-1} 1/k$ is computed by binary splitting, and $R(n^2 + 1)$ is an alternating series.

Problem C16: Compute the first N non-zero decimal digits of $f(i)$. The sequence $f(i)$ is defined by

$$f(i) = \frac{\pi^2}{6} - (13/8 + (1/(8 \cdot 27))(34/8 + (8/(8 \cdot 125))(\cdots((21i-8)/8 + ((i^3)/(8(2i+1)^3)))))).$$

The right part of $f(i)$ converges to $\frac{\pi^2}{6}$, thus we have a cancellation here.

More precisely, we can write $f(i) = \frac{\pi^2}{6} - S(i)$, where $S(i) = \sum_{k=0}^{i} f_k$, and

$$f_k = \frac{(k!)^3(21k+13)}{8^{k+1}[1 \cdot 3 \cdots (2k+1)]^3},$$

except the last term f_i which lacks the $21i + 13$ factor in the numerator, and a factor 8 in the denominator.

The program we used for the competition performed a classical summation. However, we can use binary splitting to compute $S(i)$. Define $P(k, k+1) = k^3$, except $P(0, 1) = 1$, $Q(k, k+1) = 8(2k+1)^3$, $T(k, k+1) = P(k, k+1)(21k+13)$, except $Q(i, i+1) = (2i+1)^3$ and $T(i, i+1) = P(i, i+1)$, and recursively for $c = \lfloor (a+b)/2 \rceil$,

$$P(a,b) = P(a,c)P(c,b), Q(a,b) = Q(a,c)Q(c,b), T(a,b) = Q(b,c)T(a,b) + P(a,b)T(b,c),$$

then $S(i) = T(0, i+1)/Q(0, i+1)$.

Problem C17: Compute the first N decimal digits after the decimal point of $S = -4\zeta(2) - 2\zeta3) + 4\zeta(2)\zeta(3) + 2\zeta(5)$. We have $S \approx 0.999222$. We use the identity $\zeta(2) = \pi^2/6$, and an implementation of $\zeta(n)$ for n integer with correct rounding:

$x = \circ(\pi)$
$y = \circ(x^2)$
$z = \circ(y/3)$
$t = \circ(z-1)$
$u = \circ(\zeta(3))$
$v = \circ(\zeta(5))$
$w = \circ(u-1)$
$a = \circ(v-1)$
$b = \circ(tw)$
$c = \circ(b+v)$
$d = 2c$

The obtained value d is exactly S.

For $p \geq 6$, all values are positive, and $x, z < 4$, $y < 11$, $t < 3$, $1 \leq u, v < 2$, $w < 1/4$, $a \leq 2^{-4}$, $b < 1$, and $c < 2$. We have $x = \pi(1+\theta)$, $y = \pi^2(1+\theta)^3$, $z = \pi^2(1+\theta)^4$, thus for $p \geq 6$ the absolute error on z is bounded by $13 \cdot 2^{-p}$. Now $\text{err}(t) \leq \frac{1}{2}\text{ulp}(t) + 13 \cdot 2^{-p} \leq 15 \cdot 2^{-p}$; $\text{err}(u), \text{err}(v) \leq 2^{-p}$, w and a are exact by Sterbenz theorem, thus $\text{err}(w), \text{err}(a) \leq 2^{-p}$, $\text{err}(b) \leq \frac{1}{2}\text{ulp}(b) + \text{err}(t)w + (2\zeta(2) - 1)\text{err}(w) \leq 7 \cdot 2^{-p}$, $\text{err}(c) \leq \text{err}(b) + \text{err}(v) \leq 8 \cdot 2^{-p}$, and $\text{err}(d) \leq 2^{4-p}$.

Problem C18: Compute the first N decimal digits after the decimal point of Catalan's constant G. Catalan's constant G is defined as

$$G = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)^2} \approx 0.915.$$

This is a native MPFR constant, thus we simply compute $x = \circ(G)$ with precision p and rounding to nearest. The largest possible error is $\frac{1}{2}\mathrm{ulp}(x) = 2^{-p-1}$.

Problem C19: Compute the first N decimal digits after the decimal point of $L = \sum_{n=1}^{\infty} 7^{-n^3-1}$. We have $L \approx 0.020$, thus we need only $N-1$ digits. This is simply a base-conversion problem. The base-7 representation of L is $(0.0100000010\ldots)$. We simply form a string corresponding to this base-7 representation, truncated to get enough accuracy, then convert this string to a binary floating-point value, which is then converted back to a decimal string.

Assume we truncate L to q base-7 digits, use a binary precision of p bits, and a final decimal output of $M \geq N-1$ digits, all with rounding to nearest. The error we make when truncating L to q digits is bounded by 7^{-q}, the input conversion error is bounded by $\frac{1}{2}2^{-p}$, and the output conversion error by $\frac{1}{2}10^{-M}$. If both $7^{-q} + \frac{1}{2}2^{-p} \leq \frac{1}{2}10^{-M}$, then the total error will be less than one ulp of the output. It thus suffices to have $q \geq 1 + M\frac{\log 10}{\log 7}$ and $p \geq 1 + M\frac{\log 10}{\log 2}$.

Problem C20: Compute the first Nth partial quotient of the continued fraction of $X = \sin(2\pi/17)$. We have $X \approx 0.361$. The first partial quotients are $2, 1, 3, 3, \ldots$ For $N = 10$ we get 8, for $N = 100$ we get 22, for $N = 10^3$ we get 70, 1 for $N = 10^4$, 1 for $N = 10^5$, and 3 for $N = 10^6$.

We use here the fact that X is algebraic, namely:

$$65536X^{16} - 278528X^{14} + 487424X^{12} - 452608X^{10} + 239360X^{8} - 71808X^{6} + 11424X^{4} - 816X^{2} + 17 = 0.$$

We can compute an approximation X using five square roots as follows (is it possible using only four?):

$x = \circ(\sqrt{17})$
$z = 2x$ [exact]
$y = \circ(34 - z)$
$z = \circ(34 + z)$
$y = \circ(\sqrt{y})$
$z = \circ(\sqrt{z})$
$z = 8z$ [exact]
$s = \circ(6x)$
$s = \circ(s + 34)$
$s = \circ(s - z)$
$z = \circ(x - 1)$

```
$z = \circ(zy)$
$s = \circ(s + z)$
$s = 2s$ [exact]
$s = \circ(\sqrt{s})$
$x = \circ(x + y)$
$s = \circ(s + x)$
$s = \circ(s - 1)$
$s = s/16$ [exact]
$s = \circ(s^2)$
$s = \circ(1 - s)$
$s = \circ(\sqrt{s})$
```

We claim the final error on s is less than $10.7 \cdot 2^{-p}$.

After having computed an approximation s of X, we use a subquadratic implementation of Lehmer's method — see Equation (3) page 4 from [2] — which guarantees the computed quotients to be correct.

Problem C21: Compute the first N digits after the decimal point of the root of equation $\exp(\cos x) = x$. This equation has a unique root $x \approx 1.302$, so we need to compute $N + 1$ digits and discard the initial "1".

We use the following Newton iteration, with a target precision of Q bits:

```
$x = \circ_2(1.3)$
$q = 2$
**while** true **do**
**if** $q < Q$
$q' = \lceil Q/2^k \rceil$ such that $q < q' \leq 2q$
**else** increase target precision $Q$
$p = q' + 4$
$y = \circ(\sin x)$
$z = \circ(\cos x)$
$z = \circ(\exp z)$
$u = \circ(x - z)$
$y = \circ(yz)$
$y = \circ(y + 1)$
$y = \circ(u/y)$
$x = \circ(x - y)$
$q = q'$     **if** $q \geq Q$ **then**
**if** we can round correctly to $N$ digits **then** exit
```

The error analysis is similar to that for practice problem P21.

Problem C22: Compute the first N digits after the decimal point of J. The integral J is defined by

$$J = \int_0^1 \sin(\sin(\sin x))dx.$$

We have $J \approx 0.407$. We use here an implementation by Laurent Fousse of Gauss-Legendre quadrature [3], with a rigorous bound on the total error, i.e. both the error due to the quadrature method and the rounoff error. The only assumption is the following conjecture:

Conjecture 1 *The maximum in absolute value of the nth derivative of* $\sin(\sin(\sin x))$ *on* $[0, 1]$ *for n even is attained in $x = 0$.*

For $N = 1000$, we used a composition in 26 intervals, with 218 points on each one, and a working precision of 3373 bits.

Problem C23: Compute the first 10 non-zero digits of the element $(N-1, N-3)$ of the matrix M_1. The matrix M_1 is defined as the inverse of the $N \times N$ Hankel matrix X such that $X_{ij} = 1/F_{i+j-1}$ where F_k denotes the kth Fibonacci number: $F_0 = 0$, $F_1 = 1$, $F_{k+2} = F_{k+1} + F_k$.

For $N = 4$ for example we get:

```
gp > n=4; X=matrix(n,n,i,j, 1/fibonacci(i+j-1));
gp > 1/X
%18 =
[-9 90 360 -780]

[90 -450 -2400 4680]

[360 -2400 -11520 23400]

[-780 4680 23400 -46800]
```

thus the result is the coefficient $(3, 1) = 360$. It can be shown that the inverse of M_1 has integer entries[3].

Theee different solutions were tried:

1. first; a generic solution, based on a straightforward MPFI implementation of the classical Gauss pivot. This required a colossal precision, which is not surprising given the size of the determinant of the matrix and of the coefficients of the inverse, see below. The computational complexity of this solution is of $O(N^3)$ operations on numbers which should have at least $O(N^2)$ digits of precision, so the total cost is $O(N^3 M(N^2))$.

[3] See http://arxiv.org/abs/math.LA/9905079.

2. second, Levinson-Zohar recursion for Toeplitz matrices was applied to the matrix deduced by a permutation of the rows. Unfortunately, this algorithm (which uses $O(N^2)$ operations only) proved to be highly unstable, at least in that case, and required a precision still larger than for Gauss pivot, thus requiring a still larger time than Gauss method.

3. finally, the matrix can be inverted in a formal way, and expliciting the required term, one finds that one has to evaluate

$$\frac{(-1)^N}{2}\left(\frac{\prod_{k=N-1}^{2N-6} F_k}{\prod_{k=1}^{N-4} F_k}\right)^2 F_{2N-5}F_{2N-4}^2F_{2N-3}F_{2N-2},$$

which can be evaluated in a naive way in roughly linear time, the precision required being quite small. When N grows one should be careful to divide out large powers of 10 periodically. Though this reduces the precision, it avoids exponent overflow.

A still more efficient solution would be to notice that for k greater than, say, 50, the approximation

$$F_k = \phi^k/\sqrt{5},$$

with $\phi = (1+\sqrt{5})/2$ allows one to get the approximate formula (for $N > 50$)

$$\approx \frac{(-1)^N}{2\sqrt{5}^{105}}\frac{\phi^{2N^2-2N+2548}}{\left(\prod_{k=1}^{50} F_k\right)^2}.$$

In this formula, one should take great care to perform an argument reduction on $2N^2 - 2N + 2548$ (which can be reduced modulo $\log_\phi 10$).

The whole computation has a $O(1)$ arithmetic operations cost, with precision O(1). Only $2N^2-2N+2548$ should be computed with enough precision to guarantee that $2N^2-2N+2548$ modulo $\log_\phi 10$ is known with slightly more than 10 digits, making the complexity of the whole computation roughly $O(M(\log N))$...

Problem C24: Compute the first 10 non-zero digits of the element $(N-1, N)$ of the matrix M_2. The matrix M_2 is defined as the inverse of the $N \times N$ Hankel matrix $I_N + X$ where I_N is the $N \times N$ identity matrix, and X is defined as in C23.

For $N = 4$ for example we get:

```
gp > n=4; X=matid(n)+matrix(n,n,i,j, 1/fibonacci(i+j-1));
gp > 1/X
%2 =
[29720421/36667252 -8791155/18333626 -1741410/9166813 -1281345/9166813]

[-8791155/18333626 9234675/9166813 -705300/9166813 -272610/9166813]
```

```
[-1741410/9166813 -705300/9166813 8594640/9166813 -327600/9166813]

[-1281345/9166813 -272610/9166813 -327600/9166813 8997300/9166813]
```

thus the result is the coefficient $(3, 4) = -327600/9166813 \approx -0.0357376113159$, so the correct answer is 3573761131.

The matrix M_2 has a much better conditioning than M_1 in C23, so we used the first approach mentioned in C23 (a straightforward implementation of the classical Gauss pivot with interval arithmetic). A working precision of 40 bits is enough for $N = 10$, and 41 bits for $N = 100$.

Acknowledgements. We thank Milad Niqui for comments on a previous version of this document.

References

[1] Abramowitz, M., and Stegun, I. A. *Handbook of Mathematical Functions.* Dover, 1973.

[2] Brent, R. P., van der Poorten, A. J., and te Riele, H. J. J. A comparative study of algorithms for computing continued fractions of algebraic numbers. In *Algorithmic Number Theory* (Berlin, 1996), H. Cohen, Ed., vol. 1122 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 35–47.

[3] Fousse, L. Multiple-precision correctly rounded Gauss-Legendre quadrature. Research Report 5705, Institut National de Recherche en Informatique et en Automatique, Sept. 2005. 17 pages.