Solutions to the practice problems

The MPFR team

September 20, 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 $100, 1000, 10000, \ldots$

Problem P01: Compute the first N decimal digits after the decimal point of $\sin(\sin(\sin 1))$, rounded toward zero. We have $\sin(\sin(\sin 1)) \approx 0.678$: the first N decimal digits after the decimal point match the first N mantissa digits.

We use a target decimal precision $N_1 > N$, and a binary precision p. We compute $x = \circ(\sin 1)$, $y = \circ(\sin x)$, $z = \circ(\sin y)$, with all roundings to nearest. It is easy to see that since $p \geq 3$, we have $1/2 \leq x, y, z < 1$, thus all rounding errors are bounded by 2^{-p-1}. We can thus write $x = \sin 1 + \epsilon_x$ with $|\epsilon_x| \leq 2^{-p-1}$. It follows $y = \sin(\sin 1 + \epsilon_x) + \epsilon_y$ with $|\epsilon_y| \leq 2^{-p-1}$; we can write $\sin(\sin 1 + \epsilon_x) = \sin(\sin 1) + \epsilon_x \cos\theta$, 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 $p \geq 2 + N_1 \frac{\log 10}{\log 2}$, we have $3 \cdot 2^{-p-1} < 1/2 \cdot 10^{-N_1}$.

Finally, we output the binary value z in decimal to N_1 digits, with rounding to nearest. Since $1/2 \leq z < 1$, the last digit has weight 10^{-N_1}, thus the total error — including that on z and the output error — is bounded by 10^{-N_1}. Thus, unless the last $N_1 - N$ digits of the output are all zero, we can decide the correct output to N digits, rounded toward zero.

Note: if the function $\sin(\sin(\sin x))$ was D-finite, i.e. if it would satisfy a linear differential equation with polynomial coefficients, then it would be possible to compute $\sin(\sin(\sin 1))$ to precision n in $O(M(n)\log n)$ using the "binary splitting" algorithm. Unfortunately, it does not seem that $\sin(\sin(\sin x))$ is D-finite.

Problem P02: Compute the first N decimal digits after the decimal point of $\sqrt{\pi}$. We have $\sqrt{\pi} \approx 1.772$, so we need to take the $N + 1$ first digits of the mantissa, and remove the first digit, namely "1".

Let $x = \circ(\pi)$ and $y = \sqrt{x}$, with rounding to nearest and a precision of p bits. If we use a precision of p bits, we have $x = \pi(1 + u)$ and $y = \sqrt{x}(1 + v)$ with $|u|, |v| \leq 2^{-p}$. Thus $y = \sqrt{\pi}\sqrt{1 + u}(1 + v)$. For $p \geq 2$, it is easy to see that $\sqrt{1 + u}(1 + v)$ can be written $1 + 2w$ with $|w| \leq 2^{-p}$. Thus $y = \sqrt{\pi}(1 + 2w)$, and the absolute error is bounded by 2^{2-p}.

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

Assume we output $M+1$ 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 $2^{2-p} \leq \frac{1}{2} \cdot 10^{-M}$, 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 P03: Compute the first N decimal digits after the decimal point of $\sin e$. We have $\sin e \approx 0.410$: the first N decimal digits after the decimal point match the first N mantissa digits.

Let $x = \circ(\exp 1)$ and $y = \circ(\sin x)$, with rounding to nearest and a precision of p bits. If we use a precision of p bits, we have $x = e(1+u)$ and $y = \sin(x)(1+v)$ with $|u|, |v| \leq 2^{-p}$. Since $\sin x = \sin(e + eu) = \sin e + eu \cos \theta$ for some $\theta \in (e, e+eu)$, the absolute error on y is bounded by $|v| + e|u| < 2^{2-p}$.

We find the same bound than for P02, thus the end of the analysis is identical.

Problem P04: Compute the first N decimal digits after the decimal point of $\exp(\pi\sqrt{163})$. We have $\exp(\pi\sqrt{163}) \approx 262537412640768743.999$: we thus have to compute $N+18$ digits, and disregard the first 18.

We compute $x = \circ(\pi)$, $y = \circ(\sqrt{163})$, $z = \circ(xy)$, and $t = \circ(e^z)$, with all computations to precision p and rounding to nearest.

We have $x = \pi(1+u)$, $y = \sqrt{163}(1+v)$, $z = xy(1+w)$, and $t = e^z(1+s)$, with $|u|, |v|, |w|, |s| \leq 2^{-p}$. We can thus write $z = \pi\sqrt{163}(1+\theta)^3$ with $|\theta| \leq 2^{-p}$. We have $|(1+\theta)^3 - 1| = |3\theta + 3\theta^2 + \theta^3| \leq 3|\theta| + 4\theta^2 \leq 4|\theta|$ for $p \geq 2$. The relative error on z is thus bounded by 2^{2-p}. We can write $z = \pi\sqrt{163} + h$ with $|h| \leq \pi\sqrt{163} 2^{2-p} \leq 41 \cdot 2^{2-p}$. Then $e^z = e^{\pi\sqrt{163}} \cdot e^h$. For $p \geq 8$, we have $|h| \leq 1$, thus $|e^h - 1| \leq 2|h|$. The relative error on e^z is thus bounded by $41 \cdot 2^{3-p}$, which since $e^z < 2^{58}$ corresponds to a maximal absolute error of $41 \cdot 2^{61-p}$. We must add the final rounding error, which is bounded by 2^{57-p}. This gives a final error less than 2^{66-p}.

Assume we output $M+18$ digits of the approximation t, with $M \geq N$, and rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $2^{66-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 67 + M \frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem P05: Compute the first N decimal digits after the decimal point of $\exp(\exp(\exp 1))$. We have $\exp(\exp(\exp 1)) \approx 3814279.104$, we thus have to compute $N+7$ digits, and disregard the first 7.

We compute $x = \circ(\exp 1)$, $y = \circ(\exp x)$, $z = \circ(\exp y)$, with all computations to precision p and rounding to nearest.

We have $x = e(1+u)$, $y = e^x(1+v)$, $z = e^y(1+w)$, with $|u|, |v|, |w| \leq 2^{-p}$. We use the following lemma: for $|h| \leq 1$, $|e^h - 1| \leq 2|h|$. For $p \geq 2$, we can use the lemma for $h = eu$: $e^x = e^e e^h$ can be written $e^e(1+2h')$ with $|h'| \leq 2^{-p}$; then $y = e^e(1+2h')(1+v)$ can be written $e^e(1+4v')$ with $|v'| \leq 2^{-p}$. We use again the lemma for $h' = 4e^e v'$, which is less than 1 for $p \geq 6$: $e^y = e^{e^e} e^{h'}$ can be written $e^{e^e}(1+2h'')$ with $|h''| \leq 2^{-p}$; then $z = e^{e^e}(1+2h'')(1+w)$

can be written $e^{e^e}(1+4w')$ with $|w'| \leq 2^{-p}$. Since $|e^{e^e}| < 2^{22}$, the absolute error on z is thus bounded by 2^{24-p}.

Assume we output $M+7$ digits of the approximation z, with $M \geq N$, and rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $2^{24-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 25 + M\frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem P06: Compute the first N decimal digits after the decimal point of $\log(1+\log(1+\log(1+\log(1+\pi))))$. We have $\log(1+\log(1+\log(1+\log(1+\pi)))) \approx 0.490$: the first N decimal digits after the decimal point match the first N mantissa digits.

We compute $s = \circ(\pi)$, $t = \circ(1+s)$, $u = \circ(\log t)$, $v = \circ(1+u)$, $w = \circ(\log v)$, $x = \circ(1+w)$, $y = \circ(\log x)$, $z = \circ(1+y)$, $r = \circ(\log z)$. It is easy to check that for $p \geq 9$, $2 \leq s, v < 4$, $4 \leq t < 8$, $1 \leq u, x, z < 2$, $1/2 \leq w, y < 1$, $1/4 \leq r < 1/2$.

The absolute error on s is bounded by $\frac{1}{2}\text{ulp}(s) = 2^{1-p}$, thus that on t is bounded by $2^{1-p} + \frac{1}{2}\text{ulp}(t) = 6 \cdot 2^{-p}$. We use the following lemma: if $q \geq a$ is an approximation of some unknown number $q' \geq a$ with error h bounded by ϵ, then the error on $\log q$ is at most ϵ/a. Using this lemma for $q = t$, $a = 4$, $\epsilon = 6 \cdot 2^{-p}$ yields an absolute error of at most $3/2 \cdot 2^{-p}$ for $\log t$. Together with the rounding error of at most $\frac{1}{2}\text{ulp}(u) = 2^{-p}$, this gives an absolute error $\leq 5/2 \cdot 2^{-p}$ for u. The same kind of analysis yields a bound of $9/2 \cdot 2^{-p}$ for v, $11/4 \cdot 2^{-p}$ for w, $15/4 \cdot 2^{-p}$ for x, $17/4 \cdot 2^{-p}$ for y, $21/4 \cdot 2^{-p}$ for z, and finally $11/2 \cdot 2^{-p} < 2^{3-p}$ for r.

Assume we output M digits of the approximation r, with $M \geq N$, with rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $2^{3-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 4 + M\frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem P07: Compute the first N decimal digits after the decimal point of e^{1000}. We have $e^{1000} \approx 0.197 \cdot 10^{435}$, thus we have to compute $N + 435$ digits, and disregard the first 435.

We compute $x = \circ(1000)$, $y = \circ(\exp x)$, with precision p and rounding to nearest. We choose $p \geq 7$, so that $x = 1000$ exactly. The error on y thus only consists of the final rounding error, which is bounded by $\frac{1}{2}\text{ulp}(y) \leq 2^{1442-p}$.

Assume we output $M + 435$ digits of the approximation r, with $M \geq N$, with rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $2^{1442-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 1443 + M\frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem P08: Compute the first N decimal digits after the decimal point of $\cos 10^{50}$. We have $\cos 10^{50} \approx -0.613$, the first N decimal digits after the decimal point match the first N mantissa digits (note that the sign is not requested).

We first compute $x = \circ(10^{50})$, then $y = \circ(\cos x)$.

If the precision is $p \geq 117$, then $x = 10^{50}$ exactly, thus as for P07, the only error is the final rounding error on y, which is at most $\frac{1}{2}\text{ulp}(y) = 2^{-p-1}$.

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

Problem P09: Compute the first N decimal digits after the decimal point of $\sin(3\log(640320)/\sqrt{163})$. We have $\sin(3\log(640320)/\sqrt{163}) \approx 0.221E{-15}$, thus the answer starts with 15 zeroes, followed by the first $N-15$ significant digits of the mantissa.

We compute $x = \circ(\log 640320)$, $y = \circ(\sqrt{163})$, $z = \circ(x/y)$, $s = \circ(3z)$, $t = \circ(\sin s)$. Taking the precision p large enough so that the constants 640320 and 163 are exact, e.g. $p \geq 14$, we can write $x = \log 640320(1+u)$ and $y = \sqrt{163}/(1+v)$ with $|u|, |v| \leq 2^{-p}$. Thus $x/y = \log(640320)/\sqrt{163}(1+u)(1+v)$ can be written $\log(640320)/\sqrt{163}(1+u')^2$ with $|u'| \leq 2^{-p}$, $z = \log(640320)/\sqrt{163}(1+u'')^3$ with $|u''| \leq 2^{-p}$, and $s = 3\log(640320)/\sqrt{163}(1+w)^4$ with $|w| \leq 2^{-p}$. For $p \geq 3$, we can write $(1+w)^4 = 1+5w'$ with $|w'| \leq 2^{-p}$; the absolute error on s is thus bounded by $15\log(640320)/\sqrt{163}2^{-p} \leq 15.8 \cdot 2^{-p}$. Since the sine function is contracting, the final absolute error on t is bounded by $15.8 \cdot 2^{-p} + \frac{1}{2}\mathrm{ulp}(s) = 15.8 \cdot 2^{-p} + 2^{-53-p} \leq 2^{4-p}$.

Assume we output $M-15$ digits of the approximation t, with $M \geq N$, with rounding to nearest. The output rounding error will be at most $\frac{1}{2} \cdot 10^{-M}$. If $2^{4-p} \leq \frac{1}{2} \cdot 10^{-M}$, which holds as soon as $p \geq 5 + M\frac{\log 10}{\log 2}$, the total error is bounded by 10^{-M}, i.e. one ulp of the output.

Problem P10: Compute the first N decimal digits after the decimal point of

$$z = [(32/5)^{1/5} - (27/5)^{1/5}]^{1/3} - (1 + 3^{1/5} - 9^{1/5})/25^{1/5}.$$

The constant z is identically zero. However, it is possible to output the first N decimal digits after the decimal point, since it suffices to show that $|z| < 10^{-N}$ to correctly output N zeroes.

Let $\alpha = 5^{-1/5}$ and $\beta = 3^{1/5}$. We have

$$z = [(2-\beta^3)\alpha]^{1/3} - (1+\beta-\beta^2)\alpha^2.$$

We compute successively $q = \circ(1/5)$, $r = \circ(q^{1/5})$, $s = \circ(3^{1/5})$, $u = \circ(s^2)$, $v = \circ(su)$, $w = \circ(2-v)$, $x = \circ(wr)$, $y = \circ(x^{1/3}$, $a = \circ(1+s)$, $b = \circ(a-u)$, $c = \circ(br)$, $d = \circ(cr)$, $e = \circ(y-d)$. (The powers $q^{1/5}$, $3^{1/5}$ and $x^{1/3}$ are computed with the `mpfr_root` function.) We use here the following simplified notation: $x = y(1+\theta)^k$ means that x is an approximation, which can be written $y(1+\theta)^k$ with $|\theta| \leq 2^{-p}$. We have $q = 1/5(1+\theta_1)$, $r = 5^{-1/5}(1+\theta_1)^{1/5}(1+\theta_2) = 5^{-1/5}(1+\theta_3)^2$, $s = 3^{1/5}(1+\theta_4)$, $u = 9^{1/5}(1+\theta_5)^3$, $v = 27^{1/5}(1+\theta_6)^5$. We can check that for $p \geq 9$, we have $1/16 \leq w < 1/8$, thus the rounding error on w is bounded by $\frac{1}{2}\mathrm{ulp}(w) = 2^{-p-4}$; for $p \geq 4$, we can write $(1+\theta_6)^5 = 1+6\theta_7$, thus the total error on w is at most $2^{-p-4} + 6\beta^3\theta_7 \leq 12 \cdot 2^{-p}$. We can thus write $w = W + 12\theta_8$ with $W = 2-\beta^3$. We want to be able to write $w = W(1+\theta_9)^k$ for some integer k; we thus need $W + 12\theta_8 = W(1+\theta_9)^k$, or $12\theta_8/W = (1+\theta_9)^k - 1$. A simple computation shows that $k = 241$ is enough: $w = (2-\beta^3)(1+\theta_9)^{241}$ for $p \geq 9$. We thus have $x = (2-\beta^3)\alpha(1+\theta_{10})^{244}$, $y = [(2-\beta^3)\alpha]^{1/3}(1+\theta_{11})^{83}$.

The absolute error on s being bounded by $\frac{1}{2}\mathrm{ulp}(s) = 2^{-p}$, that on a is at most $2^{-p} + \frac{1}{2}\mathrm{ulp}(a) = 3 \cdot 2^{-p}$; that on u is bounded by $9^{1/5}|(1+\theta_5)^3 - 1| \le 9^{1/5} \cdot (4\theta_5) \le 7 \cdot 2^{-p}$, thus that on b is bounded by $3 \cdot 2^{-p} + 7 \cdot 2^{-p} + \frac{1}{2}\mathrm{ulp}(b) \le 11 \cdot 2^{-p}$. We thus can write $b = B + 11 \cdot \theta_{12}$ with $B = 1 + \beta - \beta^2$; since $B \ge 1/2$, we can write similarly as above $b = B(1+\theta_{13})^{23}$.

Thus $c = (1+\beta-\beta^2)\alpha(1+\theta_{14})^{26}$, $d = (1+\beta-\beta^2)\alpha^2(1+\theta_{15})^{29}$, thus the absolute error on d is bounded by $(1+\beta-\beta^2)\alpha^2|(1+\theta_{15})^{29} - 1| \le (1+\beta-\beta^2)\alpha^2(30 \cdot 2^{-p}) \le 11 \cdot 2^{-p}$ for $p \ge 9$.

Similarly, the absolute error on y is bounded by $[(2-\beta^3)\alpha]^{1/3}|(1+\theta_{11})^{83} - 1| \le [(2-\beta^3)\alpha]^{1/3}(91 \cdot 2^{-p}) \le 34 \cdot 2^{-p}$, still for $p \ge 9$.

For $p \ge 9$, we can show that $|e| \le 5/128$, thus the rounding error on e is bounded by $\frac{1}{2}\mathrm{ulp}(e) \le 2^{-p-5}$. Therefore the total error on e is bounded by $11 \cdot 2^{-p} + 34 \cdot 2^{-p} + 2^{-p-5} \le 2^{6-p}$.

If $2^{6-p} < \frac{1}{2}10^{-N}$, i.e. $p \ge 7 + N\frac{\log 10}{\log 2}$, then since we know the exact answer is zero, we should have $|e| \le 2^{6-p}$, so we know the exact answer is less than 10^{-N} in absolute value, so the output should be N consecutive zeroes. Note that in this case no loop is needed: the first iteration should always be successful.

Problem P11: Compute the first N decimal digits after the decimal point of $\tan e + \arctan e + \tanh e + \mathrm{arctanh}(1/e)$. We have $\tan e + \arctan e + \tanh e + \mathrm{arctanh}(1/e) \approx 2.145$, thus we have to compute $N+1$ digits and discard the initial 2.

We compute $x = \circ(\exp 1)$, $y = \circ(\tan x)$, $z = \circ(\arctan x)$, $t = \circ(\tanh x)$, $u = \circ(1/x)$, $v = \circ(\mathrm{arctanh} u)$, $w = \circ(y+v)$, $a = \circ(w+t)$, $b = \circ(a+z)$.

For $p \ge 10$, we have $2 \le x, b < 3$, $-1/2 < y \le -1/4$, $1 \le z < 2$, $1/2 \le t, a < 1$, $1/4 \le u, v < 1/2$, $-1/8 < w \le -1/16$. The absolute error on x is at most $\frac{1}{2}\mathrm{ulp}(x) = 2^{1-p}$; since $x = e + h$ with $|h| \le 2^{1-p}$, we have $\tan x = \tan e + h(1 + \tan^2\theta)$ with $\theta \in (e, x)$, thus the error on y is at most $\frac{1}{2}\mathrm{ulp}(y) + 5.78 \cdot 2^{1-p} \le 11.9 \cdot 2^{-p}$. Similarly, we have $\arctan x = \arctan e + \frac{h}{1+\theta^2}$, thus the error on z is at most $\frac{1}{2}\mathrm{ulp}(z) + 1/52^{1-p} \le 1.4 \cdot 2^{-p}$. For t, we have $\tanh x = \tanh e + h(1 - \tanh^2\theta)$, thus the error on t is at most $\frac{1}{2}\mathrm{ulp}(t) + 0.071 \cdot 2^{1-p} \le 0.642 \cdot 2^{-p}$. The error on u is at most $\frac{1}{2}\mathrm{ulp}(u) + 2^{1-p}/\theta^2 \le 0.75 \cdot 2^{-p}$; then that on v is at most $\frac{1}{2}\mathrm{ulp}(v) + (0.75 \cdot 2^{-p}) \cdot 1/3 \le 0.5 \cdot 2^{-p}$. By Sterbenz theorem, $y + v$ is exact, thus the error on w is at most $11.9 \cdot 2^{-p} + 0.5 \cdot 2^{-p} \le 12.4 \cdot 2^{-p}$; that on a is at most $\frac{1}{2}\mathrm{ulp}(a) + 12.4 \cdot 2^{-p} + 0.642 \cdot 2^{-p} \le 13.6 \cdot 2^{-p}$; and finally that on b is at most $\frac{1}{2}\mathrm{ulp}(b) + 13.6 \cdot 2^{-p} + 1.4 \cdot 2^{-p} \le 17 \cdot 2^{-p} \le 2^{5-p}$.

Problem P12: Compute the first N decimal digits after the decimal point of $\arcsin(1/e) + \cosh e + \mathrm{arcsinh}\, e$. We have $\arcsin(1/e) + \cosh e + \mathrm{arcsinh}\, e \approx 9.712$, thus as in P11 we compute $N+1$ digits and discard the leading "9".

We proceed as follows: let $x = \circ(\exp 1)$, $y = \circ(1/x)$, $z = \circ(\arcsin y)$, $t = \circ(\mathrm{arcsinh}\, x)$, $u = \circ(\cosh x)$, $v = \circ(z+t)$, $w = \circ(v+u)$. For $p \ge 3$, we have $2 \le x, v < 3$, $1/4 \le y, z < 1/2$, $1 \le t < 2$, $4 \le u < 8$, $8 \le w < 16$. The same error analysis as for P11 yields a maximum error of at most 2^{1-p} for x, $0.75 \cdot 2^{-p}$ for y, $1.12 \cdot 2^{-p}$ for z, $2.79 \cdot 2^{-p}$ for t, $24.1 \cdot 2^{-p}$ for u, $5.91 \cdot 2^{-p}$ for v, and finally $38.1 \cdot 2^{-p} \le 2^{6-p}$ for w.

Problem P13: 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

$$x_{n+1} = \frac{15}{4} x_n (1 - x_n).$$

We compute it as follows:

$t_n = \circ(1 - x_n)$
$u_n = \circ(x_n t_n)$
$v_n = \circ(15 u_n)$
$x_{n+1} = v_n/4$ [exact]

For $p \geq 8$, $x_1 = \frac{15}{16} = 0.9375$ and $x_2 = \frac{225}{1024} = 0.2197265625$ are computed exactly. Since for $x_2 \leq x \leq x_1$, $x_2 \leq \frac{15}{4} x(1-x) \leq x_1$, we have $x_2 \leq x_n \leq x_1$ for all $n \geq 0$. We deduce from this that $0 \leq t_n < 1$, $0 \leq u_n \leq 1/4$, $0 \leq v_n \leq 15/4$.

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 $\frac{1}{2}\mathrm{ulp}(u_n) + \epsilon_n t_n + x_n(\epsilon_n + \tau_n)$; replacing t_n by $1 - x_n + \tau_n$, we get $2^{-p-3} + \epsilon_n + (x_n + \epsilon_n)\tau_n$. Since $\tau_n \leq \frac{1}{2}\mathrm{ulp}(t_n) \leq 2^{-p-1}$ and $x_n + \epsilon_n \leq 15/16$ — 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 $2^{-p-3} + \epsilon_n + \frac{15}{16} 2^{-p-1} \leq \epsilon_n + \frac{19}{32} 2^{-p}$.

The error on v_n is bounded by $\frac{1}{2}\mathrm{ulp}(v_n) + 15(\epsilon_n + \frac{19}{32} 2^{-p}) \leq 15\epsilon_n + \frac{83}{32} 2^{-p}$. Finally, the error on x_{n+1} is bounded by

$$\epsilon_{n+1} \leq \frac{15}{4}\epsilon_n + \frac{83}{128} 2^{-p}.$$

This recurrence admits as solution:

$$\epsilon_n = \frac{83}{352} 2^{-p}[(15/4)^n - 1] \leq 2^{-p-2}(15/4)^n.$$

Choose $M \geq N$. Since $0.2197265625 \leq x_N \leq 0.9375$, the first decimal digit of x_N has always weight $1/10$, so the Mth digit has weight 10^{-M}. If $2^{-p-2}(15/4)^n \leq \frac{1}{2} 10^{-M}$, i.e.$p \geq M \frac{\log 10}{\log 2} + n \frac{\log(15/4)}{\log 2} - 1$, then the M-digit decimal output of x_N lies within one ulp of the corresponding exact value.

Problem P14: Compute the first N decimal digits after the decimal point of a_{100N}. The sequence (a_n) is defined as follows: $a_0 = 11/2$, $a_1 = 61/11$,

$$a_{n+1} = 111 - \frac{1130 - 3000/a_{n-1}}{a_n},$$

and is due to Jean-Michel Muller. It is well known that $a_n = \frac{6^{n+1} + 5^{n+1}}{6^n + 5^n}$. So we could cheat and compute directly that closed form. However we believe this is not in the spirit of the competition.

We compute the sequence as follows, with precision p and rounding to nearest:

$b_n = \circ(3000/a_{n-1})$
$c_n = \circ(1130 - b_n)$
$d_n = \circ(c_n/a_n)$
$a_{n+1} = \circ(111 - d_n)$

Since $11/2 \le a_n \le 6$, we can show that $545 \le b_n \le 600$, $530 \le c_n \le 585$, $88 \le d_n \le 107$. Let ϵ_n be the absolute error on a_n. The error on b_n is bounded by $\frac{1}{2}\mathrm{ulp}(b_n) + \epsilon_n \frac{3000}{\theta^2}$ for some $\theta \in [a_{n-1} - \epsilon_{n-1}, a_{n-1} + \epsilon_{n-1}]$, which is at most $2^{9-p} + 100\epsilon_{n-1}$. The error on c_n is bounded by $\frac{1}{2}\mathrm{ulp}(c_n) + 2^{9-p} + 100\epsilon_{n-1} \le 15362^{-p} + 100\epsilon_{n-1}$; that on d_n is bounded by $\frac{1}{2}\mathrm{ulp}(d_n) + \mathrm{err}(c_n)/a_n + \epsilon_n \frac{585}{\theta^2} \le 3442^{-p} + 18\epsilon_{n-1} + 20\epsilon_n$. Finally a_{n+1} is exact by Sterbenz theorem, so we have

$$\epsilon_{n+1} \le 20\epsilon_n + 18\epsilon_{n-1} + 3442^{-p},$$

together with $\epsilon_0 = 0$ since $11/2$ is exact for $p \ge 4$, and $\epsilon_1 \le \frac{1}{2}\mathrm{ulp}(a_1) = 42^{-p}$. This Fibonacci-like recurrence admits an exact solution:

$$\epsilon_n 2^p \le (172/37 - 737/2183\sqrt{118})\alpha^n + (172/37 + 737/2183\sqrt{118})\beta^n - 344/37.$$

with $\alpha = 10 + \sqrt{118} \approx 20.863$, $\beta = 10 - \sqrt{118} \approx -0.863$. Since $|\beta| < 1$, it follows:

$$\epsilon_n 2^p \le (172/37 - 737/2183\sqrt{118})\alpha^n + (172/37 + 737/2183\sqrt{118}) - 344/37 \le \alpha^n.$$

Recall we want the first N digits after the decimal point of a_{100N}. Let $M \ge N$. If $\epsilon_{100N} \le \frac{1}{2}10^{-M}$, i.e.$p \ge 1 + 100N\frac{\log\alpha}{\log 2} + M\frac{log10}{\log 2}$, then the M-digit output will be within one ulp of the correct result. Note: since $\frac{\log\alpha}{\log 2} \approx 4.383$, this gives $p \approx 442N$.

Alas, this approach does not work as is. Indeed, since $a_n = \frac{6^{n+1}+5^{n+1}}{6^n+5^n}$, we have $a_{100N} \approx 6 - (5/6)^{100N}$, and thus a_{100N} is of the form $5.999\ldots999$, with about $7.9N$ consecutive "9". This means that with rounding to nearest, we need about $M \approx 7.9N$ to be able to round correctly the output.

Problem P15: Compute the first N decimal digits after the decimal point of the harmonic number h_{10N}. We recall $h_n = 1+1/2+\cdots+1/n$. We can compute h_n efficiently using the "binary splitting" method. Define $P(a,b)$ and $Q(a,b)$ as follows: if $b = a+1$, then $P(a,b) = 1$ and $Q(a,b) = b$, otherwise

$$P(a,b) = P(a,c)Q(c,b) + Q(a,c)P(c,b), \quad Q(a,b) = Q(a,c)Q(c,b), \tag{1}$$

for $c = \lfloor (a+b)/2 \rfloor$. We can easily check that $P(a,b)/Q(a.b) = 1/(a+1) + \cdots + 1/b$, and thus $h_n = P(0,n)/Q(0,n)$.

However, to get the first N decimal digits after the decimal point of h_{10N}, computing $P(0,10N)$ and $Q(0,10N)$ exactly is not very efficient. Indeed, we have $Q(0,10) = (10N)!$, which has about $10N\log_{10}(10N)$ digits, whereas we want only N digits!

To solve this problem, we use the following idea. We use a working precision p large enough to get N correct decimal digits at the end. We compute p-bit approximations of

$P(0,n)/Q(0,n)$. Once we have computed $P(a,b)$ and $Q(a,b)$ as in Eq. (1), if both exceed p bits, we truncate them by 2^k so that the smallest one has exactly p bits, with rounding to nearest. The relative error on each truncation is bounded by 2^{1-p}.

Lemma. If the maximal number of truncations along a branch of the recursive call tree is t, then the computed values $P(a,b)$ and $Q(a,b)$ satisfy $P(a,b)/Q(a,b) = h(a,b)(1+u)^t$ for $|u| \le 2^{1-p}$.

We prove the lemma by induction on $b-a$. If $b = a+1$, then P and Q are exact — we assume the working precision is large enough so that b can be represented exactly, i.e. $10N \le 2^p$ —, so the lemma holds. Assume now we have computed approximations of $P(a,c)$, $P(c,b)$, $Q(a,c)$ and $Q(c,b)$, with t_1 truncations for $P(a,c)$ and $Q(a,c)$, and t_2 runcations for $P(c,b)$ and $Q(c,b)$. We thus have $P(a,c)/Q(a,c) = h(a,c)(1+u)^{t_1}$ and $P(c,b)/Q(c,b) = h(c,b)(1+v)^{t_2}$, with $|u|, |v| \le 2^{1-p}$. If no truncation occurs for $h(a,b)$, then we have $P(a,b)/Q(a,b) = P(a,c)/Q(a,c) + P(c,b)/Q(c,b)$ exactly, thus $P(a,b)/Q(a,b) = h(a,c)(1+u)^{t_1} + h(c,b)(1+v)^{t_2}$. Since all values are positive, we can write $P(a,b)/Q(a,b) = h(a,b)(1+w)^{\max(t_1,t_2)}$. If a truncation occurs on $P(a,b)$ and $Q(a,b)$, then it induces a relative error of at most 2^{1-p} on the ratio — since both errors go in opposite directions — thus we can write $P(a,b)/Q(a,b) = h(a,b)(1+w)^{1+\max(t_1,t_2)}$.

We can easily bound the maximal number of truncations. Now since $Q(a,b) = (a+1)\cdots b$, we have $Q(a,b) \le n^{b-a}$, thus as long as $n^(b-a) < 2^p$, there can be no truncation. Here, we have $n = 10N$ and we take $2^p \ge 10^N$, so as long as $(10N)^(b-a) < 10^N$, i.e. $b-a < N\frac{\log 10}{\log(10N)}$, there is no truncation. The number of levels where there can be truncation is thus at most $\lceil \log_2(10\frac{\log(10N)}{\log 10}\rceil$. For $N \le 10^7$, this is at most 7.

After we have computed a rational approximation P/Q of h_{10N}, we convert P and Q to p-bit floating-point numbers with rounding to nearest, and we divide the two approximations. Since at least one of P and Q fits exactly into p bits, the additional error due to this conversion corresponds to $(1+u)^2$ with $|u| \le 2^{-p}$. Thus the final value is within $(1+u)^2(1+2u)^t$ of h_{10N}. For $t \le 8$ and $p \ge 4$, the relative error is bounded by 2^{5-p}.

Problem P16: Compute the first N non zero digits of $f(N)$. The sequence $f(i)$ is defined by

$$f(i) = \pi - (3 + \frac{1\cdot 1}{3\cdot 4\cdot 5}(8 + \frac{2\cdot 3}{3\cdot 7\cdot 8}(\cdots(5i-2+\frac{i(2i-1)}{3(3i+1)(3i+2)})))).$$

We can compute $f(i)$ by the following program:

$r \leftarrow 1$
for $i := N$ downto 1 do
$r \leftarrow ri(2i-1)/3/(3i+1)/(3i+2)$
$r \leftarrow 5i-2+r$
$r \leftarrow \pi - r$

The computation in the loop are done as follows:

$r \leftarrow \circ(ir)$
$r \leftarrow \circ((2i-1)r)$
$r \leftarrow \circ(r/3)$
$r \leftarrow \circ(r/(3i+1))$
$r \leftarrow \circ(r/(3i+2))$
$r \leftarrow \circ(r+(5i-2))$

The ratio between the computed value of r and the exact value after the kth iteration can be written $(1+u)^{6k}$ for $|u| \leq 2^{-p}$. This is true for $k = 0$. Assume this is true for $k \geq 0$. Then after $r \leftarrow \circ(r/(3i+2))$ the ratio can be written $(1+u)^{6k+5}$; since both r and $5i-2$ are positive, we can write $r(1+u)^{6k+5} + (5i-2) = [r+(5i-2)](1+u')^{6k+5}$, thus we get $(1+u'')^{6k+6}$ after rounding.

When $i \to \infty$, $f(i)$ converges to 0. When truncated to $i = N$, it is easy to see that $f(N) = O((2/27)^N)$. Thus to get N significant digits of $f(N)$, we need the final error to be less than 135^{-N}.

The error on r before $\pi - r$ is of the form $(1+u)^{6N}$; using $|u| \leq 2^{-p} \leq 135^{-N}$, it can be shown that $|(1+u)^{6N} - 1| \leq 7Nu$. The error when computing π is bounded by 2^{1-p}, and that of rounding $\pi - r$ too (the latter is much smaller due to the cancellation, but this bound is enough). Thus the final error is bounded by $7Nur + 2^{2-p} \leq (7N+1)2^{2-p}$.

Problem P17: Compute the first N decimal digits after the decimal point of $\zeta(2)\zeta(3)+\zeta(5)$. We have $\zeta(2)\zeta(3)+\zeta(5) \approx 3.014$, so we just need to compute $N+1$ significant digits and discard the first one.

Since the Riemann Zeta function is native in MPFR, we simply compute with precision p and rounding to nearest:

$u \leftarrow \circ(\zeta(2))$
$v \leftarrow \circ(\zeta(3))$
$w \leftarrow \circ(\zeta(5))$
$t \leftarrow \circ(uv)$
$s \leftarrow \circ(t+w)$

If θ denotes a generic quantity such that $|\theta| \leq 2^{-p}$, we have $u = \zeta(2)(1+\theta)$, $v = \zeta(3)(1+\theta)$, $w = \zeta(5)(1+\theta)$, $t = \zeta(2)\zeta(3)(1+\theta)^3$, thus since all quantities are positive, $t + w = (\zeta(2)\zeta(3)+\zeta(5))(1+\theta)^3$, and $s = (\zeta(2)\zeta(3)+\zeta(5))(1+\theta)^4$. For $p \geq 6$, we have $s \leq 25/8$; we can write $(1+\theta)^4$ as $1+5\theta$, thus the final absolute error is bounded by $5 \cdot 2^{-p}s \leq 2^{4-p}$.

Note: the `mpfr_zeta` function is quite slow for evaluating $\zeta(i)$ for i a small integer. A close look at the implementation shows that the bottleneck lies in the computation of the Bernoulli numbers, which takes more than 99% of the computing time. Also, the computation of the Bernoulli numbers could be cached and thus shared between the three evaluations of ζ, which is not the case.

Note 2: we could replace $\zeta(2)$ by $\pi^2/6$, which would give a gain of about 33% since the computation of π is quite efficient, but we thought this was not in the spirit of the competition.

Problem P18: Compute the first N decimal digits after the decimal point of Euler's γ constant. Euler's γ constant is defined as $\gamma = \lim_{n\to\infty}(1+\frac{1}{2}+\cdots+\frac{1}{n}) - \log n \approx 0.577$. This is a native MPFR constant, thus we simply compute $x = \circ(\gamma)$ with precision p and rounding to nearest. The largest possible error is $\frac{1}{2}\mathrm{ulp}(x) = 2^{-p-1}$.

Problem P19: Compute the first N decimal digits after the decimal point of $L = \sum_{n=1}^{\infty} 7^{-n^2}$. We have $L \approx 0.143$. This is simply a base-conversion problem. The base-7 representation of L is $(0.100100001\ldots)$. 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 $M \geq N$ 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 P20: Compute the Nth partial quotient from the continued fraction expansion of $\cos(2\pi/7)$. We have $\cos(2\pi/7) \approx 0.623$. Its continued fraction expansion starts with $[1, 1, 1, 1, 1, 9, 1, 2, \ldots]$. We use a subquadratic implementation of Lehmer's method[2]. We first compute an interval enclosing $\cos(2\pi/7)$, with a binary precision of about $3.5N$ bits[3]. The MPFR cos function being too slow, we use an interval Newton iteration.

The quadratic Lehmer's algorithm is used when the input size in bits is less than a given threshold (5000 bits seems near to optimal in our implementation), otherwise a subquadratic variant is used, which looks like the "half-gcd" algorithm for computing gcds.

Problem P21: Compute the first N decimal digits after the decimal point of the solution of $e^{\sin x} = x$. The equation $e^{\sin x} = x$ has a unique solution $\rho \approx 2.219$. We approximate it using Newton's iteration, with the function $f(x) = x - e^{\sin x}$. The explicit second-order expansion of $f(x)$ at $x = \rho$ yields:

$$f(\rho) = f(x) + (\rho - x)f'(x) + \frac{(\rho - x)^2}{2}f''(\theta), \tag{2}$$

for $\theta \in (x, \rho)$. Neglecting the second order term, we get the usual formula for Newton's iteration:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

---

[2]See Equation (3) page 4 of `http://web.comlab.ox.ac.uk/oucl/work/richard.brent/pub/pub166.html`.

[3]It is known from the theory of continued fractions that d decimal digits give about $\frac{6\log 2\log 10}{\pi^2}d$ partial quotients, so to get p partial quotients, we need about $\frac{\pi^2}{6\log 2\log 10}p$ decimal digits, or $\frac{\pi^2}{6\log^2 2}p \approx 3.423$ bits.

Rewriting Eq. (2) gives:

$$\rho = x - \frac{f(x)}{f'(x)} - \frac{(\rho - x)^2}{2} \frac{f''(\theta)}{f'(x)},$$

thus if $|f''| \le M$ and $|f'| \ge m$ on the considered interval, we have $|x_{k+1} - \rho| \le \frac{M}{2m}|x_k - \rho|^2$. We have $f'(x) = 1 - e^{\sin x} \cos x$, and $f''(x) = e^{\sin x}(\sin x - \cos^2 x)$, and $|f''| \le 2$ for $2 \le x \le 3$.

Here, we have $|f'(x)| \ge 2$ and $|f''(x)| \le 2$ for $2 \le x \le 3$, thus $|x_{k+1} - \rho| \le 1/2|x_k - \rho|^2$.

We use the following rounding operations to compute an approximation of x_{k+1} from that of x_k:

$y \leftarrow \circ(\sin x)$
$z \leftarrow \circ(\cos x)$
$t \leftarrow \circ(e^y)$
$u \leftarrow \circ(x - t)$
$v \leftarrow \circ(tz)$
$w \leftarrow \circ(1 - v)$
$r \leftarrow \circ(u/w)$
$s \leftarrow \circ(x - r)$

We can show that when $17/8 \le x \le 9/4$ and the precision p satisfies $p \ge 5$, then $3/4 \le y \le 7/8$, $-21/32 \le z \le -1/2$, $2 \le t \le 5/2$, $-27/16 \le v \le -1$, $2 \le w \le 11/4$.

Assume now that $|x - \rho| \le 2^{-q}$, and we apply one iteration as above. Since $|f'| \le 3$ for $2 \le x \le 3$, we have $|f(x)| \le 3 \cdot 2^{-q}$. The error on y is at most $\frac{1}{2}\mathrm{ulp}(y) = 2^{-p-1}$, that on t is at most $\frac{1}{2}\mathrm{ulp}(t) + e^{\theta} 2^{-p-1}$ for $3/4 \le \theta \le 7/8$, i.e. at most $3.2 \cdot 2^{-p}$. Thus $x - t$ is within $3.2 \cdot 2^{-p}$ of its corresponding exact value $f(x)$. But $|f(x)| \le 3 \cdot 2^{-q}$, we have $|x - t| \le 3 \cdot 2^{-q} + 3.2 \cdot 2^{-p}$. Assume $p \ge 2q$ and $q \ge 2$, then $|x - t| \le \cdot 4 \cdot 2^{-q}$. Thus the error on u is bounded by $\frac{1}{2}\mathrm{ulp}(u) + 3.2 \cdot 2^{-p} \le 2^{1-q-p} + 3.2 \cdot 2^{-p} \le 3.7 \cdot 2^{-p}$ since $q \ge 2$.

The error on z is at most $\frac{1}{2}\mathrm{ulp}(z) \le 2^{-p-1}$, that on v is bounded by $\frac{1}{2}\mathrm{ulp}(v) + \mathrm{err}(t)(|z| + \mathrm{err}(z)) + |t|\mathrm{err}(z) \le 2^{-p} + 3.2 \cdot 2^{-p}(0.68) + 5/22^{-p-1} \le 4.5 \cdot 2^{-p}$, that on w is bounded by $\frac{1}{2}\mathrm{ulp}(w) + 4.5 \cdot 2^{-p} \le 2^{1-p} + 4.5 \cdot 2^{-p} \le 6.5 \cdot 2^{-p}$. We can write $1/w = 1/f'(x_k) + 6.5 \cdot 2^{-p}/\theta^2$ for $\theta \in (w, f'(x_k))$, thus $1/w = 1/f'(x_k) + 1.7\epsilon$ with $|\epsilon| \le 2^{-p}$. This gives an error on r bounded by $\frac{1}{2}\mathrm{ulp}(r) + \mathrm{err}(u)(1/w + 1.72^{-p}) + |u|(1.72^{-p}) \le 2^{-2p} + 3.7 \cdot 2^{-p}(1/2 + 1.72^{-p}) + 2^{2-q}(1.72^{-p}) \le 3.7 \cdot 2^{-p}$ for $p \ge 6$. Then the final error on s — i.e. the difference with x_{k+1} as computed in infinite precision — is bounded by $\frac{1}{2}\mathrm{ulp}(s) + 3.7 \cdot 2^{-p} \le 2^{1-p} + 3.7 \cdot 2^{-p} \le 5.7 \cdot 2^{-p}$.

Therefore, if $p \ge 2q + 4$, then $5.7 \cdot 2^{-p} \le 2^{-2q-1}$, and since $|x_{k+1} - \rho| \le 2^{-2q-1}$, then $|s - \rho| \le 2^{-2q}$, so we get a quadratic convergence.

Note: we don't need to compute $r = \circ(u/v)$ to full precision p, since we know in advance that r is of the order of 2^{-q}, so only the $q \approx p/2$ most significant bits of r are needed. This implies in turn that u and w can be computed with precision $\approx p/2$ too. In fact, only y and t need to be computed to full precision p, since there is a cancellation in $x - t$. However the expected speedup is small, since the most expensive operations are the computations of $\sin x$, $\cos x$ and e^y.

Problem P22: Compute the first N decimal digits after the decimal point of $I = \int_0^1 \sin(\sin x)dx$. We have $I \approx 0.430$. We use here an implementation by Laurent Fousse of Gauss-Legendre quadrature, with a rigorous bound on the total error, i.e. both the error due to the quadrature method and the rounoff error.

Problem P23: Compute the first 10 decimal digits of the element $(N-1, N-3)$ of M_1. The matrix M_1 is the inverse of the $N \times N$ Hilbert matrix, whose entries are $(\frac{1}{i+j-1})$ for $1 \le i, j \le N$. For example, for $N = 7$, we have

$$M_1 = \begin{bmatrix} 49 & -1176 & 8820 & -29400 & 48510 & -38808 & 12012 \\ -1176 & 37632 & -317520 & 1128960 & -1940400 & 1596672 & -504504 \\ 8820 & -317520 & 2857680 & -10584000 & 18711000 & -15717240 & 5045040 \\ -29400 & 1128960 & -10584000 & 40320000 & -72765000 & 62092800 & -20180160 \\ 48510 & -1940400 & 18711000 & -72765000 & 133402500 & -115259760 & 37837800 \\ -38808 & 1596672 & -15717240 & 62092800 & -115259760 & 100590336 & -33297264 \\ 12012 & -504504 & 5045040 & -20180160 & 37837800 & -33297264 & 11099088 \end{bmatrix},$$

and here the element $(N-1, N-3)$ is 62092800. For $N = 10$, the element $(N-1, N-3)$ is 1766086882560, so the answer should be 1766086882. It can be seen that the entries of M_1 are integral. We assume the element $(N-1, N-3)$ cannot be represented exactly as a 10-digit floating-point number, which seems to be the case for $N \ge 10$.

We use the following approach. Using the MPFI library developed by Nathalie Revol and Fabrice Rouillier[4], we perform a naive Gaussian elimination to solve the linear system $Hx = b$, where all b entries are zero, except $b_{N-3} = 1$. The entry x_{n-1} is a binary floating-point interval $[u, v]$ enclosing the exact value of the element $(N-1, N-3)$ of M_1. If both u and v agree, when converted to 10-digit decimal floating-point values with rounding to nearest, then this common value is the wanted answer.

Experimentally, it seems that using a working precision $p \ge 4.2N \log N$ is enough. (For $N = 100$, this gives $p = 1965$, whereas $p = 1375$ is the minimal precision that works.)

Problem P24: Compute the first 10 decimal digits of the element $(N-1, N)$ of M_2. The matrix M_1 is the inverse of the $I_N + H_N$, where I_N is the $N \times N$ identity matrix, and H_N is the $N \times N$ Hilbert matrix. For $n = 4$, we have:

$$M_2 = \begin{bmatrix} \frac{10213696}{17799777} & -\frac{1084840}{5933259} & -\frac{72880}{659251} & -\frac{1377740}{17799777} \\ -\frac{1084840}{5933259} & \frac{1688800}{1977753} & -\frac{75300}{659251} & -\frac{550480}{5933259} \\ -\frac{72880}{659251} & -\frac{75300}{659251} & \frac{593280}{659251} & -\frac{57400}{659251} \\ -\frac{1377740}{17799777} & -\frac{550480}{5933259} & -\frac{57400}{659251} & \frac{16391200}{17799777} \end{bmatrix},$$

[4] http://perso.ens-lyon.fr/nathalie.revol/software.html

thus the element $(N-1, N)$ is $\frac{-57400}{659251} \approx -0.08706850653$, and the answer should be 8706850653. As in P23, we assume that element cannot be represented exactly as a 10-digit decimal floating-point value.

We use the same technique as in P23, with the MPFI library. The only difference is that, the matrix $I_N + H_N$ being much less singular, the necessary working precision is much smaller. We found experimentally that up to $N = 1000$, a precision of 46 bits is enough.

Timings

We give timings obtained on the competition machine "harif" (AMD Opteron 144 under Debian GNU/Linux "sid" unstable i386 in 32 bit mode, with 4GB of RAM). Here, the column N stands for 10^N digits, as in the original practice problems.

We used version 4.1.4 of GMP, tuned for harif: go to repository `tune`, type `make tune`, and replace the file `gmp-mparam.h` by the results obtained, in particular:

```
#define MUL_KARATSUBA_THRESHOLD          24
#define MUL_TOOM3_THRESHOLD             177
#define DIV_DC_THRESHOLD                 68
#define POWM_THRESHOLD                  116
#define GET_STR_DC_THRESHOLD             23
#define GET_STR_PRECOMPUTE_THRESHOLD     35
#define SET_STR_THRESHOLD              3962
#define MUL_FFT_TABLE  { 784, 1824, 3456, 7680, 22528, 57344, 0 }
#define MUL_FFT_MODF_THRESHOLD          848
#define MUL_FFT_THRESHOLD              8448
```

We used the cvs version from MPFR from 20 September 2005 (`cvs -D 20050920 co mpfr`), tuned for harif too (simply type `make tune` in the mpfr build directory):

```
#define MPFR_MUL_THRESHOLD 18
#define MPFR_EXP_2_THRESHOLD 32
#define MPFR_EXP_THRESHOLD 25081
```

We used MPFI version 1.3.3, with a small patch to make it work with the cvs version from MPFR.

Finally, we used INTLIB version 0.0.20050913, a numerical quadrature library from Laurent Fousse.

problem	N	cpu time	first. . . last digits
P01	4	0.181	678. . . 573
P01	5	18.062	678. . . 645
P02	4	0.021	772. . . 288
P02	5	0.830	772. . . 320
P02	6	23.310	772. . . 944
P03	4	0.089	410. . . 073
P03	5	8.251	410. . . 508
P04	4	0.090	999. . . 927
P04	5	3.271	999. . . 658
P04	6	81.741	999. . . 707
P05	4	0.151	104. . . 248
P05	5	5.213	104. . . 929
P06	4	0.192	490. . . 462
P06	5	8.056	490. . . 892
P07	4	0.022	226. . . 510
P07	5	0.665	226. . . 841
P07	6	13.853	226. . . 815
P08	4	0.159	613. . . 446
P08	5	5.466	613. . . 362
P09	4	0.235	000. . . 432
P09	5	18.864	000. . . 306
P10	4	0.198	000. . . 000
P10	5	6.684	000. . . 000
P11	4	0.548	145. . . 744
P11	5	22.439	145. . . 390
P12	4	0.460	712. . . 629
P12	5	14.292	712. . . 771

problem	N	cpu time	first. . . last digits
P13	4	8.804	824. . . 580
P14	2	29.591	999. . . 999
P15	4	0.205	090. . . 123
P15	5	6.300	392. . . 432
P16	4	0.391	112. . . 637
P16	5	68.860	326. . . 023
P17	3	3.070	014. . . 886
P18	4	0.790	577. . . 165
P18	5	24.165	577. . . 897
P19	4	0.003	143. . . 377
P19	5	0.135	143. . . 205
P19	6	3.372	143. . . 250
P19	7	70.630	143. . . 382
P20	4	0.047	1
P20	5	1.238	1
P20	6	27.138	1
P21	4	0.229	219. . . 878
P21	5	15.096	219. . . 495
P22	3	15.931	430. . . 309
P23	2	2.698	9844998112
P24	2	0.196	2933301369