APC 523 HW 1

Amlan Sinha
amlans
03.07.2019
Due 03.13.2019

1. $x = \pm \left(\sum_{\ell=1}^{\infty} b_{-\ell} 2^{-\ell} \right) \cdot 2^{e}$

Rounding

Consider the following case where we round an infinite series representing x to p significant figures.

$$\underbrace{\qquad\qquad\qquad}_{p}\ \underbrace{\quad}_{q}\ \underbrace{\qquad\qquad\qquad}_{r}$$

So, the error due to rounding is given by $q+r$. But, the error may depend on whether we are rounding up or rounding down. Whether we are rounding up or rounding down depends on q. If it is 0, we round down and if it is 1, we round up.

Round-down

$$|x - r_d(x)| = \left(\sum_{\ell=1}^{\infty} b_{-\ell} 2^{-\ell} \right) \cdot 2^{e} - \left(\sum_{\ell=1}^{p} b_{-\ell} 2^{-\ell} \right) \cdot 2^{e}$$

$$= \left(\sum_{\ell=p+1}^{\infty} b_{-\ell} 2^{-\ell} \right) \cdot 2^{e}$$

$$\max |x - r_d(x)| = 0 + \left(\sum_{\ell=p+2}^{\infty} 2^{-\ell} \right) \cdot 2^{e}$$

$$= 2^{-p-1} \cdot 2^{e}$$

maximum abs. error obtained when $q = 0$, $r = 1$'s

$$\frac{\max |x - r_d(x)|}{\min |x|} = \frac{2^{-p-1} \cdot 2^{e}}{2^{-1} \cdot 2^{e}}$$

$$= 2^{-p}$$

Round-up

$$|x - r_u(x)| = \left(\sum_{l=1}^{\infty} b_{-l} 2^{-l}\right) \cdot 2^e - \left(\sum_{l=1}^{p} b_{-l} 2^{-l}\right) \cdot 2^e$$

$$= \left(\sum_{l=p+1}^{\infty} b_{-l} 2^{-l}\right) \cdot 2^e$$

$$\max|x - r_u(x)| = 2^{-p-1} \cdot 2^e + 0$$

$$= 2^{-p-1} \cdot 2^e$$

maximum abs. error obtained when

$q = 1$, $r = 0$'s

$$\frac{\max|x - r_u(x)|}{\min|x|} = \frac{2^{-p-1} \cdot 2^e}{2^{-1} \cdot 2^e}$$

$$= 2^{-p}$$

$$\left|\frac{x - rd(x)}{x}\right| \leq 2^{-p}$$

2. d. (iii) and (iv) give at least 100% error.

(i) and (ii) perform almost similarly. In fact, the error on both is exactly the same. It should be noted that subtraction is the operation which results in the highest error. In both (i) and (ii), we are doing similar operations as far as subtractions are concerned. This is why they yield similar error.

e. Instead of computing $e^{-5.5}$ directly, we could compute $\frac{1}{e^{5.5}}$. That way, we will not be doing any subtraction operations.

$$e^{-5.5} = \frac{1}{\sum_{n=0}^{\infty} \frac{x^n}{n!}}$$

Achieves an error of $\approx 0.007383\%$ wrt true value of $e^{-5.5}$

3.a.i. Suppose $x \in \mathbb{R}$ is a machine number.

$$fl(x) = x$$

What happens when we multiply x by x:

$$fl(x \cdot x) = x^2(1+\epsilon_m)$$
$$fl(fl(x \cdot x) \cdot x) = x^3(1+\epsilon_m)^2 = x^3(1+2\epsilon_m+\epsilon_m^2) \approx x^3(1+2\epsilon_m)$$
$$fl(fl(fl(x \cdot x) \cdot x)) = x^4(1+2\epsilon_m)(1+\epsilon_m) = x^4(1+3\epsilon_m+2\epsilon_m^2) \approx x^4(1+3\epsilon_m)$$

$fl(x^n) = x^n(1+(n-1)\epsilon_m)$ where $\epsilon_m \leq eps$, error $\leq (n-1)eps$

ii. Suppose $x \in \mathbb{R}$ is a machine number, $n \in \mathbb{R}$ is also a machine number

$$fl(x) = x$$

$$fl(\ln x) = \ln x\,(1+\epsilon_\ell)$$
$$fl(n\,fl(\ln x)) = n \ln x\,(1+\epsilon_m)(1+\epsilon_\ell)$$
$$fl(\exp(fl(n\,fl(\ln x)))) = \exp((n \ln x)(1+\epsilon_m)(1+\epsilon_\ell))(1+\epsilon_e)$$
$$= \exp(n \ln x\,(1+\epsilon_m+\epsilon_\ell+\epsilon_m\epsilon_\ell))(1+\epsilon_e)$$
$$\approx \exp(n \ln x\,(1+\epsilon_m+\epsilon_\ell))(1+\epsilon_e)$$
$$= \exp(n \ln x)\exp(n \ln x\,(\epsilon_m+\epsilon_\ell))(1+\epsilon_e)$$
$$= \exp(n \ln x)(1+n \ln x\,(\epsilon_m+\epsilon_\ell))(1+\epsilon_e)$$
$$= \exp(n \ln x)(1+\epsilon_e+n \ln x\,(\epsilon_m+\epsilon_\ell)+n \ln x\,\epsilon_e(\epsilon_m+\epsilon_\ell))$$
$$= \exp(n \ln x)(1+\epsilon_e+n \ln x\,(\epsilon_m+\epsilon_\ell))$$

where $\epsilon_m, \epsilon_\ell, \epsilon_e \leq eps$, error $\leq eps\,(1+2n \ln x)$

When the value of $x \ll 1$, $\ln x \to -\infty$, and so, the error due to exponentiation is larger than the one due to repeated multiplication. When the value of $x \gg 1$, the growth of $\ln x$ slows down and the error due to exponentiation depends more on n. But since it depends on $2n$ while the error due to repeated multiplication depends on $(n-1)$ the error due to repeated multiplication is lower than that due to exponentiation.

b.i. Given,

$x \in \mathbb{R}$ is a machine number

$a \in \mathbb{R}$ is not a machine number

$$x^{a(1+\epsilon_a)} = x^a \cdot x^{a\epsilon_a}$$
$$= x^a \exp(a\epsilon_a \ln x)$$
$$= x^a (1 + a\epsilon_a \ln x) \quad \text{where } \epsilon_a \leq \text{eps}, \text{ error} \leq (a \ln x)\,\text{eps}$$

ii. Given,

$x \in \mathbb{R}$ is not a machine number

$a \in \mathbb{R}$ is a machine number

$$(x(1+\epsilon_x))^a = x^a (1+\epsilon_x)^a$$
$$= x^a (1 + a\epsilon_x) \quad \text{where } \epsilon_x \leq \text{eps}, \text{ error} \leq a\,\text{eps}$$

The propagated error in the first scenario depends on $\ln x$. So, for $x << 1$, $\ln x \to -\infty$ and the error can become substantial.

4. $f(x) = 1 - e^{-x}$ on the interval $[0, 1]$

a. $$(\text{cond } f)(x) = \left| \frac{x f'(x)}{f(x)} \right|$$

$$= \left| \frac{x e^{-x}}{1 - e^{-x}} \right|$$

$$= \left| \frac{x}{e^{x} - 1} \right|$$

To show it's bounded by 1 on the interval $[0, 1]$, we will evaluate $(\text{cond } f)(x=0)$ and show that $(\text{cond } f)'(x=0) < 0$

@ $x = 0$,

$$(\text{cond } f)(x=0) \underset{\uparrow}{=} \left| \frac{1}{e^{0}} \right| = 1$$

L'Hopital's Rule

$$(\text{cond } f)'(x) = \frac{(e^{x} - 1) - x e^{x}}{(e^{x} - 1)^2} = \frac{1}{e^{x} - 1} - \frac{x e^{x}}{(e^{x} - 1)^2}$$

$$(\text{cond } f)'(x=0) \underset{\uparrow}{=} - \frac{(2+0)e^{0}}{2e^{0}(2e^{0} - 1)} = -1$$

L'Hopital's Rule

b. Suppose $x \in \mathbb{R}$ is a machine number

$fl(x) = x$

Steps in the algorithm:

1. $fl(-x) = -x$

$fl(\exp(fl(-x))) = \exp(-x)(1 + \epsilon_e)$

$fl(1 - fl(\exp(fl(-x)))) = (1 - \exp(-x)(1 + \epsilon_e))(1 + \epsilon_s)$

$$= (1 + \epsilon_s - \exp(-x)(1 + \epsilon_e)(1 + \epsilon_s))$$

$$= 1 + \epsilon_s - \exp(-x)(1 + \epsilon_e + \epsilon_s)$$

$$= 1 + \epsilon_s - \exp(-x) - \epsilon_e \exp(-x) - \epsilon_s \exp(-x)$$

$$= (1 - \exp(-x))\left(1 + \frac{\epsilon_s - \exp(-x)(\epsilon_e + \epsilon_s)}{1 - \exp(-x)}\right)$$

We know,

$f_A(x) = f(x_A)$

$$\Rightarrow (1 - \exp(-x))\left(1 + \frac{\epsilon_s - \exp(-x)(\epsilon_e + \epsilon_s)}{1 - \exp(-x)}\right) = 1 - \exp(-x_A)$$

$$\Rightarrow 1 + \epsilon_s - \exp(-x)(1 + \epsilon_e + \epsilon_s) = 1 - \exp(-x_A)$$

$$\Rightarrow -\exp(-x_A) = \epsilon_s - \exp(-x)(1 + \epsilon_e + \epsilon_s)$$

$$\Rightarrow -\exp(-x_A) = \exp(-x)\left(\frac{\epsilon_s}{\exp(-x)} - (1 + \epsilon_e + \epsilon_s)\right)$$

$$\Rightarrow \exp(-x_A) = \exp(-x)\left(-\frac{\epsilon_s}{\exp(-x)} + (1 + \epsilon_e + \epsilon_s)\right)$$

$$\Rightarrow -x_A = -x + \ln\left(-\frac{\epsilon_s}{\exp(-x)} + (1 + \epsilon_e + \epsilon_s)\right)$$

$$\Rightarrow x_A - x = -\ln\left(1+\epsilon_e+\epsilon_s\left(1-\frac{1}{\exp(-x)}\right)\right)$$

$$\Rightarrow |x_A - x| = \ln\left(1+\epsilon_e+\epsilon_s\left(1-\frac{1}{\exp(-x)}\right)\right)$$

$$= \epsilon_e + \epsilon_s(1-\exp(x))$$

$$\therefore \left|\frac{x_A - x}{x}\right| = \frac{\epsilon_e+\epsilon_s(1-\exp(x))}{x} \quad \text{where } |\epsilon_e|, |\epsilon_s| \leq eps$$

To maximise $\left|\frac{x_A - x}{x}\right|$, choose $\epsilon_e = eps$, $\epsilon_s = -eps$

$$\left|\frac{x_A - x}{x}\right| \leq \underbrace{\frac{\exp(x)}{x}}_{\text{cond } A(x)} eps$$

$$\text{cond } A'(x) = \frac{x\exp(x) - \exp(x)}{x^2} = 0$$

$$\Rightarrow \exp(x)(x-1) = 0$$

$$\Rightarrow \exp(x) = 0 \quad \text{or} \quad x - 1 = 0$$

$$\Rightarrow x = 1$$

$$\text{cond } A'(x^*) = \exp(1)$$

$\uparrow$

note this is a minimum

$\therefore$ cond $A(x) > \exp(1) > 1 \ \forall x \in [0,1]$

On the interval $[0,1]$, $\frac{\exp(x)}{x} > 1$ as shown in the plot attached.

c) See plots attached.

As $x \to 0$, cond $A(x \to 0) = \frac{\exp(x\to 0)}{x \to 0} \to \infty$. So, it becomes more and more ill-conditioned for smaller values of x due to x being in the denominator of cond $A(x)$

d) $$2^{-b} \leq 1 - e^{-x} \leq 2^{-a} \quad \forall a \leq b$$

For 1 bit of significance lost, $b = 1$.

$$2^{-1} \leq 1 - e^{-x}$$

$$\Rightarrow \frac{1}{2} \leq 1 - e^{-x}$$

$$\Rightarrow e^{-x} \leq \frac{1}{2}$$

$$\Rightarrow -x \leq \ln\frac{1}{2}$$

$$\therefore x \geq -\ln\frac{1}{2}$$

$$\geq 0.693 = -\ln\frac{1}{2}$$

For 2 bit of significance lost, $b = 2$

$$x \geq 0.288 = -\ln\frac{3}{4}$$

For 3 bit of significance lost, $b = 3$

$$x \geq 0.134 = -\ln\frac{7}{8}$$

For 4 bit of significance lost, $b = 4$

$$x \geq 0.065 = -\ln\frac{15}{16}$$

$$\left|\frac{y_A - y}{y}\right| \leqslant \text{cond } f(x) \left(\underbrace{\varepsilon_{rd,x}}_{=0} + \text{cond } A(x) \cdot \text{eps} \right)$$

$$= \text{cond } f(x) \left(\text{cond } A(x) \cdot \text{eps} \right)$$

$$= \frac{x}{e^x - 1} \frac{e^x}{x} \text{eps}$$

$$= \frac{e^x}{e^x - 1} \text{eps}$$

@ $x = -\ln \frac{1}{2}$

$$\left|\frac{y_A - y}{y}\right| \leqslant 2$$

@ $x = -\ln \frac{3}{4}$

$$\left|\frac{y_A - y}{y}\right| \leqslant 4$$

@ $x = -\ln \frac{7}{8}$

$$\left|\frac{y_A - y}{y}\right| \leqslant 8$$

@ $x = -\ln \frac{15}{16}$

$$\left|\frac{y_A - y}{y}\right| \leqslant 16$$

f. $f(x) = 1 - e^{-x}$

The algorithm works well for values of x which are not close to 0. Therefore, we could use the same algorithm when $x >> 0$ in the interval $[0, 1]$. When it's close to 0, however; we can use a different method such as using a taylor series approximation for e^{-x}.

Taylor Series

$$f(x) \approx 1 - \left(1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots\right)$$

$$\approx x - \frac{x^2}{2!} + \frac{x^3}{3!} - \cdots$$

We can truncate the series at a point which gives us a satisfactory truncation error.

Code converged at iteration number 18.

As $n \to \infty$, $\frac{1}{n} \to 0$

$$n = 10^{-16} = 2^{\frac{\log 10^{-16}}{\log 2}} = 2^{-53}$$

Note: In double precision, the mentissa has 53 bits

This means that if we increase n further than this, there is not going to be enough slots in the mentissa to represent the value of $\frac{1}{n}$ with. So, the value of epsilon at $n = 10^{-16}$ and $n = 10^{-17}$ is going to be the same.

6. Let $x = x^{2^{-i}}$ where i represents the number of iterations.

$$x^{2^{-i}} = e^{2^{-i} \ln x}$$

$$= 1 + \frac{\ln x}{2^i} + \cdots$$

$$\approx 1 + \frac{n}{2^i} \quad \left[x = e^n\right]$$

If $i = 52$,

$$x^{2^{-52}} \approx 1 + \frac{n}{2^{52}}$$

In double precision, the mentissa has 53 bits.

If $n = 1$, then it can be fully represented by the 52-bit mentissa. But n is slightly larger than 1 or slightly smaller than 2, the information cannot be fully represented by the 52 bit mentissa. This is why, we see the same result for $e \leq x < e^2$ when $i = 52$.

Similarly, if $i = 51$,

$$x^{2^{-51}} \approx 1 + \frac{n}{2^{51}}$$

Here, half integer powers of e appear as jumps because the first binary decimal place will not be lost due to precision limits. Similar arguments can be made for other values of i.

$$\left|\frac{y_A - y}{y}\right| \leq \text{cond } f(x) \left(\underbrace{\varepsilon_{rd,x}}_{=0} + \text{cond } A(x) \cdot \text{eps} \right)$$

$$= \text{cond } f(x) \left(\text{cond } A(x) \cdot \text{eps} \right)$$

$$= \frac{x}{e^x - 1} \frac{e^x}{x} \text{eps}$$

$$= \frac{e^x}{e^x - 1} \text{eps}$$

@ $x = -\ln \frac{1}{2}$

$$\left|\frac{y_A - y}{y}\right| \leq 2$$

@ $x = -\ln \frac{3}{4}$

$$\left|\frac{y_A - y}{y}\right| \leq 4$$

@ $x = -\ln \frac{7}{8}$

$$\left|\frac{y_A - y}{y}\right| \leq 8$$

@ $x = -\ln \frac{15}{16}$

$$\left|\frac{y_A - y}{y}\right| \leq 16$$

f. $f(x) = 1 - e^{-x}$

The algorithm works well for values of x which are not close to 0. Therefore, we could use the same algorithm when $x \gg 0$ in the interval $[0, 1]$. When it's close to 0, however, we can use a different method such as using a taylor series approximation for e^{-x}.

Taylor Series

$$f(x) \approx 1 - \left(1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots\right)$$

$$\approx x - \frac{x^2}{2!} + \frac{x^3}{3!} - \cdots$$

We can truncate the series at a point which gives us a satisfactory truncation error.

7.b. Both the newton-raphson and the built-in polynomial root finding algorithm converge to the largest root within acceptable margin of error.

c. As the value of δ increases, the root found by both the newton-raphson and the built-in polynomial root finding algorithm is drastically different from the actual root. It moves farther and farther away as the value of δ increases.

d. Both the roots have become complex.

e.i. $\Omega_k \rightarrow \Omega_k + \delta\Omega_k$

$a_\ell \rightarrow a_\ell + \delta a_\ell$

Given,

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + x^n$$

$$p'(x) = a_1 + 2a_2 x + \cdots + (n-1)a_{n-1} x^{n-2} + n x^{n-1}$$

$$\underset{a_\ell \rightarrow a_\ell + \delta a_\ell}{p}(\Omega_k + \delta\Omega_k) = a_0 + a_1(\Omega_k + \delta\Omega_k) + a_2(\Omega_k + \delta\Omega_k)^2 + \cdots + (a_\ell + \delta a_\ell)(\Omega_k + \delta\Omega_k)^\ell + \cdots + (\Omega_k + \delta\Omega_k)^n$$

$$\Rightarrow \quad 0 = a_0 + a_1\Omega_k\left(1 + \frac{\delta\Omega_k}{\Omega_k}\right) + a_2\Omega_k^2\left(1 + \frac{\delta\Omega_k}{\Omega_k}\right)^2 + \cdots + (a_\ell + \delta a_\ell)\Omega_k^\ell\left(1 + \frac{\delta\Omega_k}{\Omega_k}\right)^\ell + \cdots + \Omega_k^n\left(1 + \frac{\delta\Omega_k}{\Omega_k}\right)^n$$

$$= a_0 + a_1(\Omega_k + \delta\Omega_k) + a_2(\Omega_k^2 + 2\Omega_k\delta\Omega_k) + \cdots + (a_\ell + \delta a_\ell)(\Omega_k^\ell + \ell\Omega_k^{\ell-1}\delta\Omega_k) + \cdots + (\Omega_k^n + n\Omega_k^{n-1}\delta\Omega_k)$$

$$= \underbrace{a_0 + a_1\Omega_k + a_2\Omega_k^2 + \cdots + a_\ell\Omega_k^\ell + \Omega_k^n}_{p(\Omega_k) = 0} + \underbrace{a_1\delta\Omega_k + 2a_2\Omega_k\delta\Omega_k + \ell a_\ell\Omega_k^{\ell-1}\delta\Omega_k + n\Omega_k^{n-1}\delta\Omega_k}_{p'(\Omega_k)\delta\Omega_k}$$

$$+ \delta a_\ell \Omega_k^\ell$$

$$\Rightarrow p'(\Omega_k)\delta\Omega_k + \delta a_\ell \Omega_k^\ell = 0$$

$$\Rightarrow \frac{\delta\Omega_k}{\delta a_\ell} = -\frac{\Omega_k^\ell}{p'(\Omega_k)}$$

$$T_{k,\ell} = \left| \frac{a_\ell}{\Omega_k} \frac{\Omega_k^\ell}{p'(\Omega_k)} \right| = \left| \frac{a_\ell \Omega_k^{\ell-1}}{p'(\Omega_k)} \right|$$

$$\text{cond } \Omega_k(\vec{a}) = \sum_{\ell=0}^{n-1} \left| \frac{a_\ell \Omega_k^{\ell-1}}{p'(\Omega_k)} \right|$$

ii. $r = 14$, cond $\Omega_k(\vec{a}) = 6.0336 \times 10^{13}$

$r = 16$, cond $\Omega_k(\vec{a}) = 3.9825 \times 10^{13}$

$r = 17$, cond $\Omega_k(\vec{a}) = 1.7052 \times 10^{13}$

$r = 20$, cond $\Omega_k(\vec{a}) = 1.3798 \times 10^{13}$

iii. No.

The condition number demonstrates how the roots vary with the coefficients of the polynomial. It is inherent to the problem itself and not the algorithm. Therefore, no matter what algorithm we use, it will not help us.

$$y_{n+1} = e - (n+1)y_n$$

$$y_{k+1} = e - (k+1)y_k$$

$$\Rightarrow y_k = \frac{e - y_{k+1}}{k+1}$$

$$\Rightarrow y_{n-1} = \frac{e - y_n}{n}$$

$$y_k = \frac{k!}{N!}(a - (b - (\cdots (e - y_N))))$$

$$\frac{|dy_k|}{|dy_N|} = \frac{k!}{N!}$$

$$\therefore \text{cond } g_k(y_N) \leq \left|\frac{k!\, y_N}{N!\, y_k}\right|$$

b. $\left|\frac{\epsilon_k}{\epsilon_N}\right| = \left|\frac{k!\, y_N}{N!\, y_k}\right|$

From the definition of $y_N = \int_0^1 e^x x^N \, dx$, then $y_N < y_k \ \forall \ N > k$

$$\left|\frac{y_N}{y_k}\right| < 1$$

$$\epsilon \geq \frac{k!}{N!}$$

$$\Rightarrow N! \geq \frac{k!}{\epsilon}$$

c. For $k = 20$, $\epsilon = 2^{-53}$

$$\frac{k!}{\epsilon} = 2.19 \times 10^{34}$$

Choose N s.t. $N! \geq 2.19 \times 10^{34}$

$\therefore N = 32$

d. It matches with the output from Wolfram Alpha.

WolframAlpha computational intelligence.

integrate e^x*x^20 from 0 to 1

Browse Examples

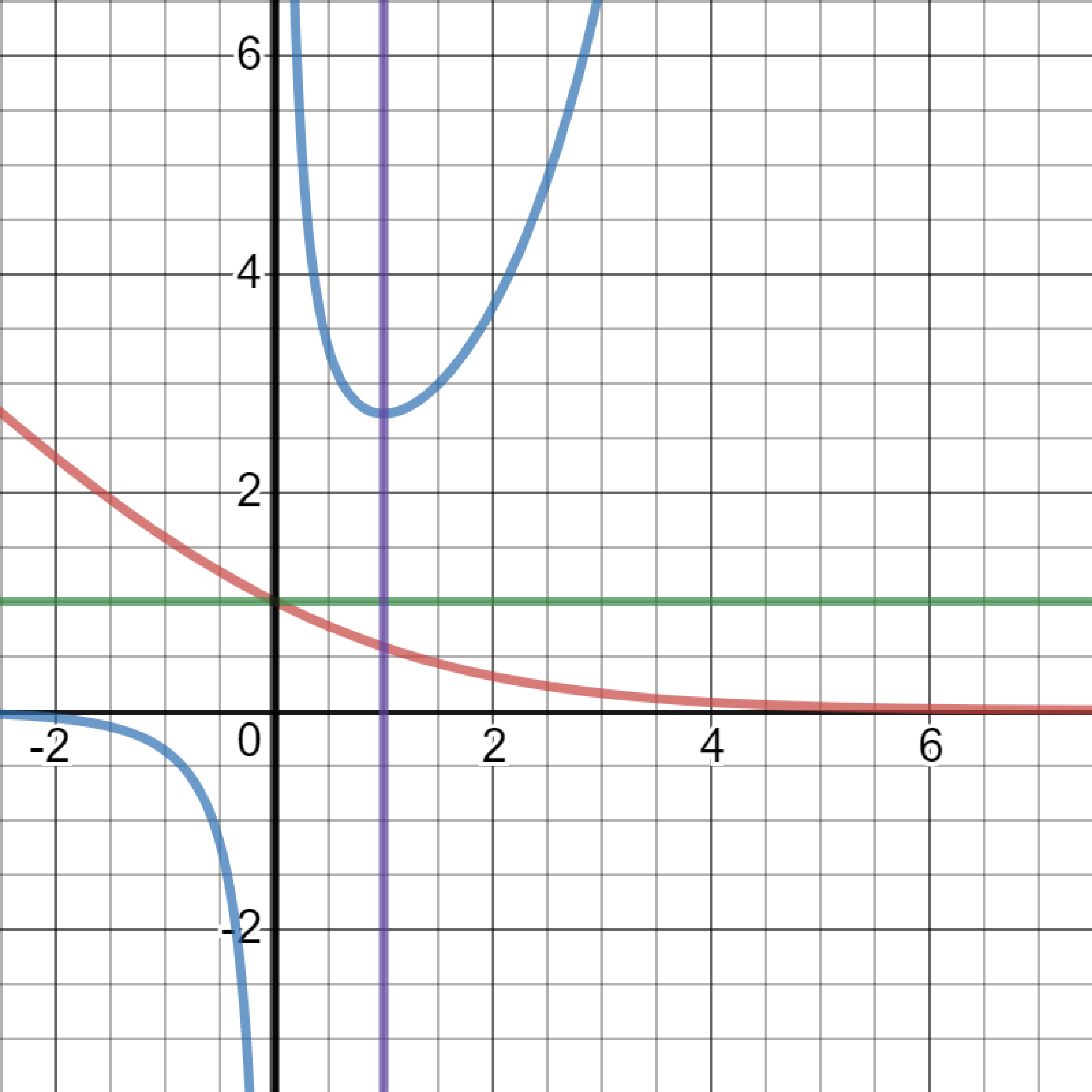
Surprise Me

Definite integral:

More digits

Step-by-step solution

$$\int_0^1 e^x x^{20}\, dx = 209\,(4\,282\,366\,656\,425\,369\, e - 11\,640\,679\,464\,960\,000) \approx 0.12380$$


6
4
2
0
-2
-2
2
4
6