

# xjb: Fast Float to String Algorithm

**ANONYMOUS\*** and **ANONYMOUS\***, Anonymous, Anonymous

Efficiently and accurately converting floating-point numbers to decimal strings is a critical challenge in numerical computation and data exchange. While existing algorithms like Ryū, Dragonbox, and Schubfach satisfy the Steele-White (SW) principle for accuracy, they often suffer from performance bottlenecks due to branch prediction failures and high-precision multiplication overhead. This paper presents a novel floating-point to string conversion algorithm called "xjb", an optimized variant of the Schubfach algorithm designed to deliver superior performance for IEEE754 single-precision (binary32) and double-precision (binary64) floating-point numbers. By minimizing instruction dependencies, reducing multiplication operations, mitigating branch prediction penalties and by utilizing the SIMD instruction set, xjb achieves significant performance gains. The algorithm features concise core implementation, parallel computing support, and excellent portability and scalability. Extensive benchmarking across diverse platforms, including AMD R7-7840H and Apple M1, demonstrates that xjb outperforms state-of-the-art algorithms in most scenarios while maintaining full compliance with the SW principle.

CCS Concepts: • Computing methodologies → Representation of mathematical objects.

Additional Key Words and Phrases: floating-point, printing, algorithm, performance

## ACM Reference Format:

Anonymous and Anonymous. 2026. xjb: Fast Float to String Algorithm. *ACM Trans. Program. Lang. Syst.* 37, 4, Article 111 (August 2026), 29 pages. <https://doi.org/XXXXXX.XXXXXXXX>

## 1 INTRODUCTION

In 1990, Steele and White [1] published the paper *how to print floating-point numbers Accurately* and proposed the optimal principle of floating-point number printing algorithms (hereinafter referred to as the SW principle) :

- **Information preservation:** The print result can be parsed back to the original floating-point number.
- **Minimum length:** The print result should be as short as possible.
- **Correct rounding:** On the basis of satisfying 1 and 2, if there are two candidate values, they should be correctly rounded (i.e., the even value should be selected).
- **Generate from left to right:** The print result is generated from the left.

Floating-point number printing algorithms that satisfy the SW principle convert floating-point numbers into real values with unique and definite results. Over the past few years, a variety of different algorithms have been proposed, such as Grisu3[2], Errol[3], Ryū[4][5], Schubfach[6], Grisu-Exact[7], Dragonbox[8], and yy\_double[9].

---

\*Both authors contributed equally to this research.

---

Authors' address: **Anonymous**, Anonymous.com; **Anonymous**, Anonymous.cn, Anonymous, Anonymous, Anonymous, Anonymous.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0164-0925/2026/8-ART111

<https://doi.org/XXXXXX.XXXXXXXX>

The algorithm in this paper is based on the Schubfach algorithm, and is inspired by algorithms such as yy\_double and Dragonbox. This article only introduces two floating-point number types, IEEE754-binary32 and IEEE754-binary64. To simplify the content, in this article, float represents IEEE754-binary32 and double represents IEEE754-binary64. The article altogether contains nine python code files, and in this paper, the algorithm implementation code you can be found at <https://github.com/xjb714/xjb>.

## 2 IEEE754 FLOATING POINT NUMBER REPRESENTATION

Since the print result of a negative floating-point number only has one more negative sign than the print result of its absolute value, this article only discusses positive floating-point numbers and does not include special values such as 0, NaN, and Inf.

The IEEE754 double-precision floating-point number consists of 64 bits, including 1 sign bit (*sign*), 11 exponent bits (*exp*), and 52 fraction bits (*frac*). *sign*'s range is 0 or 1, *exp*'s range is [0, 2047], and *frac*'s range is  $[0, 2^{52} - 1]$ .

The IEEE754 single-precision floating-point number consists of 32 bits, including 1 sign bit (*sign*), 8 exponent bits (*exp*), and 23 fraction bits (*frac*). *sign*'s range is 0 or 1, *exp*'s range is [0, 255], and *frac*'s range is  $[0, 2^{23} - 1]$ .

When *frac* = 0, it is an irregular floating-point number.

The real value of the positive floating-point number *v* can be expressed as the following expression:

$$\begin{aligned} \text{double:} v &= (\text{frac} + (\text{exp} \neq 0? 2^{52} : 0)) \cdot 2^{\max(\text{exp}, 1) - 1075} = c \cdot 2^q \\ \text{float:} v &= (\text{frac} + (\text{exp} \neq 0? 2^{23} : 0)) \cdot 2^{\max(\text{exp}, 1) - 150} = c \cdot 2^q \end{aligned} \quad (1)$$

There are two cases in total. When *exp* equals 0 (referred to as subnormal floating-point numbers), there are:

$$\begin{aligned} \text{double:} v &= \text{frac} \cdot 2^{-1074} \\ \text{float:} v &= \text{frac} \cdot 2^{-149} \end{aligned} \quad (2)$$

When *exp* is not equal to 0 (referred to as a normal floating-point number), there is:

$$\begin{aligned} \text{double:} v &= (\text{frac} + 2^{52}) \cdot 2^{\text{exp} - 1075} \\ \text{float:} v &= (\text{frac} + 2^{23}) \cdot 2^{\text{exp} - 150} \end{aligned} \quad (3)$$

In the rounding interval  $R_v$  of floating-point numbers, all real numbers will be rounded to this floating-point number when parsed.  $R_v$  is:

$$\begin{aligned} v_l &= \begin{cases} \left(c - \frac{1}{2}\right) \cdot 2^q, & \text{if } \text{frac} \neq 0 \text{ or } \text{exp} \leqslant 1 \\ \left(c - \frac{1}{4}\right) \cdot 2^q, & \text{if } \text{frac} = 0 \end{cases} \\ v_r &= \left(c + \frac{1}{2}\right) \cdot 2^q \\ R_v &= \begin{cases} [v_l, v_r], & \text{if } \text{frac} \% 2 = 0 \\ (v_l, v_r), & \text{if } \text{frac} \% 2 = 1 \end{cases} \end{aligned} \quad (4)$$

When the floating-point number is a regular floating-point number,  $2^{q-1}$  is the rounded radius.

## 3 PRINCIPLE OF ALGORITHM

At present, other algorithms use a large number of branches, which can easily lead to branch prediction failure penalties and excessive high multiplication overhead. The algorithm in this paper

will minimize the overhead of branch prediction failures and reduce the number of multiplication operations to improve performance. Moreover, the core code for the algorithm implementation in this paper is very concise and it also supports parallel computing. The process of printing floating-point numbers is usually divided into two parts: the first part is to convert the floating-point number to a decimal number, and the second part is to convert the decimal number to a string. And this article will only introduce the first part. All double-precision floating-point numbers are classified into two types: irregular values and regular values. An irregular value is one where all the lower 52 bits are 0, meaning the *frac* value is 0. There are a total of 2046 valid irregular values (i.e., *exp* values range from 1 to 2046). Dividing by the irregular values yields the regular value. Similarly, there are a total of 254 irregular values in a single-precision floating-point number. When *exp* is 0, it is called a subnormal floating-point number.

The valid range for *c* and *q* in regular floating-point numbers is:

$$\begin{aligned} \text{float : } & \begin{cases} 1 \leq c \leq 2^{24} - 1, c \neq 2^{23}; q = -149 \\ 2^{23} + 1 \leq c \leq 2^{24} - 1; -148 \leq q \leq 104 \end{cases} \\ \text{double : } & \begin{cases} 1 \leq c \leq 2^{53} - 1, c \neq 2^{52}; q = -1074 \\ 2^{52} + 1 \leq c \leq 2^{53} - 1; -1073 \leq q \leq 971 \end{cases} \end{aligned} \quad (5)$$

The valid range for *c* and *q* in irregular floating-point numbers is:

$$\begin{aligned} \text{float : } & \{c = 2^{23}; -149 \leq q \leq 104\} \\ \text{double : } & \{c = 2^{52}; -1074 \leq q \leq 971\} \end{aligned} \quad (6)$$

The valid range for *c* and *q* in subnormal floating-point numbers is:

$$\begin{aligned} \text{float : } & \{c \leq 2^{23} - 1; q = -149\} \\ \text{double : } & \{c \leq 2^{52} - 1; q = -1074\} \end{aligned} \quad (7)$$

Floating-point numbers that do not fall within the subnormal range are called normal floating-point numbers.

Regular floating-point numbers account for the vast majority of all possible values of floating-point numbers and are the most worthy of discussion part. Therefore, unless otherwise specified, only regular floating-point numbers will be discussed below. Suppose the floating-point number *v* is converted to the optimal solution that satisfies the SW principle as *opt*, *d* is a positive integer and *k* is an integer, which is expressed as:

$$\begin{aligned} v = c \cdot 2^q \rightarrow opt = d \cdot 10^k \\ opt \in R_v; d \in N^+; k \in Z \end{aligned} \quad (8)$$

For example: IEEE754-binary64 floating-point number "1.3", the real value of the floating-point number is 1.300000000000000444089209850062616169452667236328125, hexadecimal representation of floating-point Numbers is 3ff4cccccccccd, Then the *opt* value that meets the SW principle is 1.3. The IEEE754-binary32 floating-point number "1.3" has an actual value of 1.2999999523162841796875, and its hexadecimal representation is 3FA66666. Therefore, the *opt* value that satisfies the SW principle is 1.3.

### 3.1 Review the Schubfach algorithm and the derivation of the algorithm in this paper

According to the Schubfach algorithm, the possible values of *d* can be one of the following four situations:

$$10 \cdot \lfloor v \cdot 10^{-k-1} \rfloor, \lfloor 10 \cdot (v \cdot 10^{-k-1}) \rfloor, \lfloor 10 \cdot (v \cdot 10^{-k-1}) \rfloor + 1, 10 \cdot \lfloor v \cdot 10^{-k-1} \rfloor + 10 \quad (9)$$

148 The calculation method of  $k$  in equation (9) is as follows:

$$149 \quad k = \lfloor q \cdot \lg(2) \rfloor \text{ if } v \in \text{regular} \text{ else } \lfloor q \cdot \lg(2) - \lg\left(\frac{4}{3}\right) \rfloor \quad (10)$$

151 In the range of float and double, equation (10) can be equivalent to:

$$153 \quad \text{double : } k = (q \cdot 315653 - (v \in \text{regular}? 0 : 131072)) \gg 20 \\ 154 \quad \text{float : } k = (q \cdot 1233 - (v \in \text{regular}? 0 : 512)) \gg 12 \quad (11)$$

156 Suppose the integer part of  $v \cdot 10^{-k-1}$  is  $m$  and the decimal part is  $n$ , then we have:

$$157 \quad \lfloor v \cdot 10^{-k-1} \rfloor = m \\ 158 \quad v \cdot 10^{-k-1} = m + n \\ 159 \quad 0 \leq n = v \cdot 10^{-k-1} - \lfloor v \cdot 10^{-k-1} \rfloor < 1 \quad (12)$$

162 Then the decimal part of  $v \cdot 10^{-k}$  is expressed as:

$$163 \quad v \cdot 10^{-k} - \lfloor v \cdot 10^{-k} \rfloor = 10m + 10n - \lfloor 10m + 10n \rfloor = 10n - \lfloor 10n \rfloor \quad (13)$$

165 The possible values of  $d$  obtained from equation (9) are:

$$166 \quad 10m, \lfloor 10(m+n) \rfloor, \lfloor 10(m+n) \rfloor + 1, 10m + 10 \quad (14)$$

168 The possible values of  $d$  in equation (14) can be simplified to:

$$169 \quad 10m, 10m + \lfloor 10n \rfloor, 10m + \lfloor 10n \rfloor + 1, 10m + 10 \quad (15)$$

171 Among them,  $10m$  represents the minimum possible value and  $10m + 10$  represents the maximum possible value. Suppose  $ten$  is used to represent  $10m$ . There are four possible values for  $one$ , with  $d = ten + one$ , denoted as:

$$174 \quad ten = 10m \\ 175 \quad one \in \{0, \lfloor 10n \rfloor, \lfloor 10n \rfloor + 1, 10\} \\ 176 \quad d = ten + one \quad (16)$$

178 Calculating  $d$  will be converted to calculating  $ten$  and  $one$ .

179 The final possible values of  $d$  are as follows:

- 180 •  $10m$

181 When the following conditions are met, the result is  $10m$  (or equivalent to  $one = 0$ ). That  
182 is, the floating-point number  $v$  minus the minimum possible value of  $10m$  is less than the  
183 rounded radius  $2^{q-1}$ .

$$184 \quad c \cdot 2^q - 10m \cdot 10^k < 2^{q-1} \\ 185 \quad c \cdot 2^q - \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor \cdot 10^{k+1} < 2^{q-1} \\ 186 \quad c \cdot 2^q \cdot 10^{-k-1} - \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor < 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 187 \quad n < 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 188 \quad 2^{-1} \cdot 2^q \cdot 10^{-k-1} > n \quad (17)$$

189 Or when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$ ,  $c \% 2 = 0$  must also be satisfied. Therefore, the following  
190 conditions are valid:

$$191 \quad \text{if } 2^{-1} \cdot 2^q \cdot 10^{-k-1} > n \text{ or } \left(2^{-1} \cdot 2^q \cdot 10^{-k-1} = n \text{ \&& } c \% 2 = 0\right) : one = 0 \quad (18)$$

197 •  $10m + 10$

198 When the following conditions are met, the result is  $10m + 10$  (or equivalent to  $one = 10$ ).  
 199 The maximum possible value of  $10m + 10$  minus the floating-point number  $v$  is less than  
 200 the rounded radius  $2^{q-1}$ .

$$\begin{aligned} 203 \quad & (10m + 10) \cdot 10^k - c \cdot 2^q < 2^{q-1} \\ 204 \quad & \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor \cdot 10^{k+1} + 10^{k+1} - c \cdot 2^q < 2^{q-1} \\ 205 \quad & \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor - c \cdot 2^q \cdot 10^{-k-1} + 1 < 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 206 \quad & 1 - n < 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 207 \quad & 2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n \\ 208 \end{aligned} \quad (19)$$

212 Or when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ ,  $c \% 2 = 0$  must also be satisfied. Therefore, the following  
 213 conditions are valid:

$$216 \quad \text{if } 2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n \text{ or } \left( 2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n \text{ \&& } c \% 2 = 0 \right) : one = 10 \quad (20)$$

220 •  $10m + \lfloor 10n \rfloor$  or  $10m + \lfloor 10n \rfloor + 1$

221 When none of the conditions are met as  $d = 10m$  or  $d = 10m + 10$ ,  $d$  is either  $10m + \lfloor 10n \rfloor$   
 222 or  $10m + \lfloor 10n \rfloor + 1$ . The final value is determined based on the decimal part of  $10n$ . If the  
 223 decimal part is 0.5, it is rounded to the nearest even value; if it is not 0.5, it is rounded to  
 224 the nearest value. For irregular floating-point numbers, it is also necessary to determine  
 225 whether  $10m + \lfloor 10n \rfloor$  is within the rounding interval  $R_v$ . If it is not, then  $10m + \lfloor 10n \rfloor + 1$ .

227 In summary, the steps of the Schubfach algorithm variants are as follows process (21), that is,  
 228 the algorithms proposed in this paper. This algorithm process (21) is applicable to float and double  
 229 floating-point numbers(xjb32 for float, xjb64 for double). Taking a floating-point number  $v$  as input,  
 230  $c$  and  $q$  are extracted, and the calculation results  $d$  (line 15) and  $k$  (line 2) are returned. The real value  
 231 represented by the returned results is  $d \cdot 10^k$ , which conforms to the SW principle. The calculation  
 232 process of  $k$  is relatively simple and can be obtained from (11). Therefore, the following only focuses  
 233 on introducing the rapid calculation process of  $d$ . The following will be divided into five parts to  
 234 introduce the algorithm process (21) :

- 236 (1) Introduce the pre-computation process of the algorithm's lookup table.
- 237 (2) Quickly calculate  $m$ .
- 238 (3) Quickly determine whether  $one = 0$  or  $one = 10$ .
- 239 (4) Quickly calculate  $\lfloor 10n \rfloor$  and determine whether  $one = \lfloor 10n \rfloor$  or  $one = \lfloor 10n \rfloor + 1$  based on  
 240 the decimal part of  $10n$ .
- 241 (5) Processing of irregular floating-point numbers.

243 The following will discuss the above content in detail from Section 3.2 to Section 3.6.

```

246
247     input :  $c, q$ 
248     output :  $d, k$ 
249
250     ( 1)  $v = c \cdot 2^q$ 
251     ( 2)  $k = \lfloor q \cdot \lg(2) \rfloor$  if  $v \in regular$  else  $\lfloor q \cdot \lg(2) - \lg(\frac{4}{3}) \rfloor$ 
252
253     ( 3)  $m = \lfloor v \cdot 10^{-k-1} \rfloor, n = v \cdot 10^{-k-1} - m$ 
254
255     ( 4)  $ten = 10m$ 
256
257     ( 5) if  $10n - \lfloor 10n \rfloor = 0.5$  :  $one = \lfloor 10n \rfloor$  if  $(\lfloor 10n \rfloor \% 2 = 0)$  else  $\lfloor 10n \rfloor + 1$ 
258
259     ( 6) if  $10n - \lfloor 10n \rfloor < 0.5$  :  $one = \lfloor 10n \rfloor$ 
260
261     ( 7) if  $10n - \lfloor 10n \rfloor > 0.5$  :  $one = \lfloor 10n \rfloor + 1$ 
262
263     ( 8) if  $v \in irregular$  :
264
265     ( 9) if  $10n - \lfloor 10n \rfloor > 2^{q-2} \cdot 10^{-k}$  :  $one = \lfloor 10n \rfloor + 1$ 
266
267     (10) if  $2^{q-2} \cdot 10^{-k-1} \geq n$  :  $one = 0$ 
268
269     (11) else :
270
271     (12) if  $2^{q-1} \cdot 10^{-k-1} > n$  or  $(2^{q-1} \cdot 10^{-k-1} = n \text{ \&& } c \% 2 = 0)$  :  $one = 0$ 
272
273     (13) endif
274
275     (14) if  $2^{q-1} \cdot 10^{-k-1} > 1 - n$  or  $(2^{q-1} \cdot 10^{-k-1} = 1 - n \text{ \&& } c \% 2 = 0)$  :  $one = 10$ 
276
277     (15)  $d = ten + one$ 
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294

```

### 3.2 Pre-computation of Lookup Table

The algorithm in this paper uses a lookup table to store the values of  $10^{-k-1}$  for  $q$  in the range of  $[-149, 104]$  for float and  $[-1074, 971]$  for double. In the algorithm of this paper, float uses 64-bit precision and double uses 128-bit precision lookup tables. The code implementation in this section is `gen.py`. Suppose the bit length of a single value data in the lookup table is  $B$ . For float, it has  $B = 64$ , and for double, it has  $B = 128$ . Suppose there are integers  $e_{10}$  and real numbers  $e_2$ , where  $1 \leq f < 2$ . There are:

$$f \cdot 2^{\lfloor e_2 \rfloor} = 2^{e_2} = 10^{e_{10}} \quad (22)$$

Then:

$$\lfloor e_2 \rfloor = \lfloor e_{10} \cdot \log_2(10) \rfloor \quad (23)$$

The calculation leads to  $f$ , and the following conclusions are drawn:

$$f = \frac{10^{e_{10}}}{2^{\lfloor e_{10} \cdot \log_2(10) \rfloor}} \quad (24)$$

The way to calculate the lookup table is as follows (using the upward rounding method) :

$$lookup[e_{10}] = \lceil f \cdot 2^{B-1} \rceil = \lceil \frac{10^{e_{10}}}{2^{\lfloor e_{10} \cdot \log_2(10) \rfloor}} \cdot 2^{B-1} \rceil = \lceil 10^{e_{10}} \cdot 2^{B-1-\lfloor e_{10} \cdot \log_2(10) \rfloor} \rceil \quad (25)$$

For float, when  $0 \leq e_{10} \leq 27$ ,  $f \cdot 2^{B-1}$  is an integer in equation (25). For double, when  $0 \leq e_{10} \leq 55$ ,  $f \cdot 2^{B-1}$  is an integer in equation (25). The detailed calculation process is as follows:

- Float

295 The range of  $-k - 1$  is calculated to be  $[-32, 44]$  through the  $q$  value range in equation (5),  
 296 so the lookup table contains representation values from 10 to the power of -32 to 10 to the  
 297 power of 44. The calculation process is as follows:

$$\begin{aligned} & -32 \leq e_{10} \leq 44 \\ & e_2 = \lfloor \lfloor e_{10} \cdot \log_2(10) \rfloor \rfloor - 63 \\ & pow10t = \begin{cases} 2^{e_2} / 10^{\lfloor e_{10} \rfloor}; & \text{if } e_{10} < 0 \\ 10^{\lfloor e_{10} \rfloor} // 2^{e_2}; & \text{if } e_{10} \geq 20 \\ 10^{\lfloor e_{10} \rfloor} \cdot 2^{e_2}; & \text{if } 1 \leq e_{10} \leq 19 \end{cases} \\ & f_{1,e_{10}} = pow10 = pow10t + (e_{10} \geq 0 \& \& e_{10} \leq 27? 0 : 1) \end{aligned} \quad (26)$$

306 When  $0 \leq e_{10} \leq 27$ , the lookup table variable indicates that the values  $f_{1,e_{10}} \cdot 2^{\lfloor e_{10} \cdot \log_2(10) \rfloor - 63}$   
 307 and  $10^{e_{10}}$  are equal. In other cases, the relative error is less than  $2^{-63}$ . Expressed as:  
 308

$$\begin{aligned} & r_{1,e_{10}} = \frac{f_{1,e_{10}} \cdot 2^{\lfloor e_{10} \cdot \log_2(10) \rfloor - 63}}{10^{e_{10}}} \\ & \in \begin{cases} 1; & \text{if } 0 \leq e_{10} \leq 27 \\ (1, 1 + 2^{-63}); & \text{if } e_{10} < 0 \text{ or } e_{10} > 27 \end{cases} \end{aligned} \quad (27)$$

- Double

314 The range of  $-k - 1$  is calculated to be  $[-293, 323]$  through the  $q$  value range in equation  
 315 (5), so the lookup table contains representation values from 10 to the power of -293 to 10  
 316 to the power of 323. The calculation process is as follows:

$$\begin{aligned} & -293 \leq e_{10} \leq 323 \\ & e_2 = \lfloor \lfloor e_{10} \cdot \log_2(10) \rfloor \rfloor - 127 \\ & pow10t = \begin{cases} 2^{e_2} / 10^{\lfloor e_{10} \rfloor}; & \text{if } e_{10} < 0 \\ 10^{\lfloor e_{10} \rfloor} // 2^{e_2}; & \text{if } e_{10} \geq 39 \\ 10^{\lfloor e_{10} \rfloor} \cdot 2^{e_2}; & \text{if } 1 \leq e_{10} \leq 38 \end{cases} \\ & f_{1,e_{10}} = pow10 = pow10t + (e_{10} \geq 0 \& \& e_{10} \leq 55? 0 : 1) \end{aligned} \quad (28)$$

326 When  $0 \leq e_{10} \leq 55$ , the lookup table variable indicates that the values  $f_{1,e_{10}} \cdot 2^{\lfloor e_{10} \cdot \log_2(10) \rfloor - 127}$   
 327 and  $10^{e_{10}}$  are equal. In other cases, the relative error is less than  $2^{-127}$ . Expressed as:  
 328

$$\begin{aligned} & r_{1,e_{10}} = \frac{f_{1,e_{10}} \cdot 2^{\lfloor e_{10} \cdot \log_2(10) \rfloor - 127}}{10^{e_{10}}} \\ & \in \begin{cases} 1; & \text{if } 0 \leq e_{10} \leq 55 \\ (1, 1 + 2^{-127}); & \text{if } e_{10} < 0 \text{ or } e_{10} > 55 \end{cases} \end{aligned} \quad (29)$$

334 The following uses  $r_1$  to represent all possible errors of the lookup table values within the float  
 335 range,  $r_2$  to represent all possible errors of the lookup table values within the double range, and  $r$   
 336 to represent all possible errors of the lookup table values within either the float or double range.  
 337 In algorithm process (21), an approximate representation value of 10 to the power of  $-k - 1$  needs  
 338 to be obtained through a lookup table. From equation (27) and equation (29), the lookup table  
 339 representation value is error-free when  $q$  is within the following range:

$$\begin{aligned} & float : 0 \leq -k - 1 \leq 27 \Rightarrow -93 \leq q \leq -1 \\ & double : 0 \leq -k - 1 \leq 55 \Rightarrow -186 \leq q \leq -1 \end{aligned} \quad (30)$$

When  $q$  is not within the range of equation (30), the error range of the value represented by the lookup table can be concluded as follows:

$$\begin{aligned} \text{float : } & 0 < r_1 - 1 < 2^{-63} \\ \text{double : } & 0 < r_2 - 1 < 2^{-127} \end{aligned} \quad (31)$$

The introduction of the lookup table calculation process is complete. The storage space required for a float range lookup table is 616 bytes, and that for a double range lookup table is 9872 bytes.

### 3.3 Quickly Calculate $m$

Relevant theorems (partially from the Dragonbox algorithm paper) : Suppose there are positive integers  $n, P$ , and  $Q$ , where  $P$  and  $Q$  are coprime,  $P < Q$ ,  $1 \leq n \leq n_{max}$ ,  $Q > n_{max}$ ,  $P^*/Q^*$  is the best rational approximation result greater than or equal to  $P/Q$ ,  $P_*/Q_*$  is the best rational approximation result less than or equal to  $P/Q$ , and it satisfies  $Q^* \leq n_{max}$ ,  $Q_* \leq n_{max}$ . And if  $n \cdot P$  does not divide  $Q$  evenly, it is expressed as:

$$\lfloor n \cdot \frac{P}{Q} \rfloor + 1 = \lceil n \cdot \frac{P}{Q} \rceil \quad (32)$$

Suppose the following holds true:

$$\lfloor n \cdot \frac{P}{Q} \rfloor = \lfloor n \cdot \xi \rfloor \quad (33)$$

Then there are:

$$\frac{P_*}{Q_*} = \max_{1 \leq n \leq n_{max}} \frac{\lfloor n \cdot \frac{P}{Q} \rfloor}{n} \leq \xi < \min_{1 \leq n \leq n_{max}} \frac{\lfloor n \cdot \frac{P}{Q} \rfloor + 1}{n} = \min_{1 \leq n \leq n_{max}} \frac{\lceil n \cdot \frac{P}{Q} \rceil}{n} = \frac{P^*}{Q^*} \quad (34)$$

Therefore, the range of values for  $\xi$  is:

$$\frac{P_*}{Q_*} \leq \xi < \frac{P^*}{Q^*} \quad (35)$$

And the range of the decimal part with  $n \cdot \frac{P}{Q}$  is:

$$\left[ \frac{(Q_*P) \% Q}{Q}, \frac{(Q^*P) \% Q}{Q} \right] \quad (36)$$

That is, when  $n = Q_*$ , the decimal part is the smallest; when  $n = Q^*$ , the decimal part is the largest.

The definition of the best rational approximation function is as follows (this function is implemented on line 15 of the `test1.py` file):

$$(DN, UP) = f(C, P, Q) \quad (37)$$

The function (37) calculate the best rational approximation result with a denominator not exceeding  $C$  based on the mean term theorem of the Farey sequence.  $DN$  and  $UP$  are two adjacent terms in the  $C$ -order Farey sequence  $F_C$ .

In algorithm process (21),  $m$  is calculated as  $\lfloor v \cdot 10^{-k-1} \rfloor$  (line 3). Just prove that the following equation holds:

$$m = \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor c \cdot 2^q \cdot r \cdot 10^{-k-1} \rfloor \quad (38)$$

Where  $r$  is the error of the lookup table value, as defined in equation (27) and equation (29). When the condition (30) is met,  $r$  is 1, and the equation (38) clearly holds. When  $r$  is not 1, there is:

$$\begin{aligned} \text{float : } & 1 < r < 1 + 2^{-63} \\ \text{double : } & 1 < r < 1 + 2^{-127} \end{aligned} \quad (39)$$

393 Calculate the range of  $2^q \cdot 10^{-k-1}$  and we get:

$$394 \quad 395 \quad 2^q \cdot 10^{-k-1} = 10^{-1} \cdot \left( 10^{q \cdot \lg(2) - \lfloor q \cdot \lg(2) \rfloor} \right) \quad (40)$$

396 When  $q$  is not 0, equation (40) exists:

$$397 \quad 398 \quad q \cdot \lg(2) \neq \lfloor q \cdot \lg(2) \rfloor \\ 399 \quad 400 \quad 0 < q \cdot \lg(2) - \lfloor q \cdot \lg(2) \rfloor < 1 \quad (41)$$

401 When  $q$  is 0,  $q \cdot \lg(2) - \lfloor q \cdot \lg(2) \rfloor = 0$ , so the final conclusion is:

$$402 \quad 403 \quad 10^{-1} \leq 2^q \cdot 10^{-k-1} < 1 \quad (42)$$

404 Because there is:

$$405 \quad 406 \quad c \cdot 2^q \cdot 10^{-k-1} = c \cdot \frac{2^{q-k-1}}{5^{k+1}} \in [0.1c, c) \quad (43)$$

407 Therefore:

$$408 \quad 409 \quad 410 \quad 411 \quad c \cdot 2^q \cdot 10^{-k-1} = \begin{cases} \frac{c \cdot 2^{q-k-1}}{5^{k+1}}; q \geq 1 \\ \frac{c}{2^{1+k-q} \cdot 5^{k+1}} = \frac{c}{10}; q = 0 \\ \frac{c \cdot 5^{-k-1}}{2^{1+k-q}}; q < 0 \end{cases} \quad (44)$$

412 Suppose:

$$413 \quad 414 \quad c \cdot 2^q \cdot 10^{-k-1} = c \cdot \frac{x}{y} < c \quad (45)$$

415 Then there are:

$$416 \quad 417 \quad 418 \quad 419 \quad (x, y) = \begin{cases} (2^{q-k-1}, 5^{k+1}); q \geq 1 \\ (1, 10); q = 0 \\ (5^{-k-1}, 2^{1+k-q}); q < 0 \end{cases} \quad (46)$$

420 Suppose:

$$421 \quad 422 \quad \text{float : } c \leq c_{\max} = C_1 = 2^{24} - 1 \\ 423 \quad \text{double : } c \leq c_{\max} = C_2 = 2^{53} - 1 \quad (47)$$

424 The following is represented by  $C$  as  $C_1$  or  $C_2$ ,  $C$  within the float range is  $C_1$ , and  $C$  within the double range is  $C_2$ .

425 When  $y > C$ , calculate the  $P^*$  and  $Q^*$  corresponding to each  $q$  by calling  $f(C, x, y)$  according to function (37). And calculate the minimum *BIT* value when the following conditions are met:

$$426 \quad 427 \quad \frac{x}{y} \left( 1 + 2^{-\text{BIT}} \right) < \frac{P^*}{Q^*} \quad (48)$$

428 When  $y \leq C$ , there is:

$$429 \quad 430 \quad c \cdot \frac{x}{y} \left( 1 + \frac{1}{Cy} \right) = \frac{cx + \frac{c}{C} \cdot \frac{x}{y}}{y} < \frac{cx + 1}{y} \quad (49)$$

431 Therefore:

$$432 \quad 433 \quad \lfloor c \cdot \frac{x}{y} \rfloor = \lfloor c \cdot \frac{x}{y} \left( 1 + \frac{1}{Cy} \right) \rfloor \quad (50)$$

434 Similarly, calculate the minimum *BIT* value:

$$435 \quad 436 \quad \frac{x}{y} \left( 1 + 2^{-\text{BIT}} \right) < \frac{x}{y} \left( 1 + \frac{1}{Cy} \right) \quad (51)$$

442 In summary, the calculation results of the maximum value among the minimum *BIT* values corresponding to different  $q$  are as follows (the running result is in the `test1.py` file, and the running time of this code is only about 1 to 2 seconds) :

$$\begin{aligned} 445 \quad & \text{float : } \text{BIT}_{\max} = 52 \\ 446 \quad & \text{double : } \text{BIT}_{\max} = 113 \end{aligned} \quad (52)$$

448 Therefore, the following conclusions exist:

$$\begin{aligned} 449 \quad & \text{float : } \lfloor c \cdot \frac{x}{y} \rfloor = \lfloor c \cdot \frac{x}{y} \cdot (1 + 2^{-52}) \rfloor = \lfloor c \cdot \frac{x}{y} \cdot r_1 \rfloor \\ 450 \quad & \text{double : } \lfloor c \cdot \frac{x}{y} \rfloor = \lfloor c \cdot \frac{x}{y} \cdot (1 + 2^{-113}) \rfloor = \lfloor c \cdot \frac{x}{y} \cdot r_2 \rfloor \end{aligned} \quad (53)$$

453 This section has been verified. After quickly calculating  $m$ , the value of  $ten = 10m$  can be obtained very quickly.

### 3.4 Quickly Determine Whether $one = 0$ or $one = 10$

457 In algorithm process (21), the conditions for determining  $one = 0$  and  $one = 10$  are on lines 12, and 14. This section will introduce how to quickly determine whether  $one = 0$  or  $one = 10$  holds by using equivalent conditions.

460 When discussing the case of  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  (line 12,  $one$  might be 0), it is equivalent to:

$$\begin{aligned} 461 \quad & c \cdot 2^q \cdot 10^{-k-1} - \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor = 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 462 \quad & (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} = \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor \end{aligned} \quad (54)$$

465 When discussing the case of  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$  (line 14,  $one$  might be 10), it is equivalent to:

$$\begin{aligned} 466 \quad & \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor - c \cdot 2^q \cdot 10^{-k-1} + 1 = 2^{-1} \cdot 2^q \cdot 10^{-k-1} \\ 467 \quad & (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} = \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor + 1 \end{aligned} \quad (55)$$

470 Since equation (42), we have:

$$2^{q-1} \cdot 10^{-k-1} \in [0.05, 0.5] \quad (56)$$

472 Therefore, there is:

$$\begin{aligned} 473 \quad & \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor - 1 < c \cdot 2^q \cdot 10^{-k-1} - 0.5 \\ 474 \quad & < (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} \\ 475 \quad & \leq c \cdot 2^q \cdot 10^{-k-1} - 0.05 < \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor + 1 \end{aligned} \quad (57)$$

478 Therefore, for equation (54), when  $(2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is an integer, it must be equal to  $\lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor$ . Similarly, for equation (55), there is:

$$\begin{aligned} 480 \quad & \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor < c \cdot 2^q \cdot 10^{-k-1} + 0.05 \\ 481 \quad & \leq (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} \\ 482 \quad & < c \cdot 2^q \cdot 10^{-k-1} + 0.5 < \lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor + 2 \end{aligned} \quad (58)$$

485 Therefore, for equation (55), when  $(2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is an integer, it must be equal to  $\lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor + 1$ .

487 In conclusion, it is equivalent to discussing whether  $(2c \pm 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is an integer. There are:

$$(2c \pm 1) \cdot 2^{q-1} \cdot 10^{-k-1} = (2c \pm 1) \cdot 2^{q-k-2} \cdot 5^{-k-1} \quad (59)$$

491 According to the range of  $q$ , there are:

$$\begin{cases} q - k - 2 \geq 0, -k - 1 < 0; q \geq 2 \\ q - k - 2 < 0, -k - 1 < 0; 1 \geq q \geq 0 \\ q - k - 2 < 0, -k - 1 \geq 0; q < 0 \end{cases} \quad (60)$$

496 Therefore, equation (59) is equivalent to:

$$(2c \pm 1) \cdot 2^{q-1} \cdot 10^{-k-1} = \begin{cases} \frac{(2c \pm 1) \cdot 2^{q-k-2}}{5^{k+1}}; q \geq 2 \\ \frac{(2c \pm 1)}{2^{2+k-q} \cdot 5^{k+1}}; 1 \geq q \geq 0 \\ \frac{(2c \pm 1) \cdot 5^{-k-1}}{2^{2+k-q}}; q < 0 \end{cases} \quad (61)$$

501 According to the different ranges of  $q$ , the following situations are discussed:

- 503 •  $q \geq 2$

504 From  $q \geq 2$ , we get  $k \geq 0$ . When  $q \geq 2$ , it is equivalent to discussing whether  $(2c \pm 1) \cdot 2^{q-k-2}$   
 505 is divisible by  $5^{k+1}$ . Since 2 and 5 are coprime, it is equivalent to discussing whether  $(2c \pm 1)$   
 506 is divisible by  $5^{k+1}$ .

$$(2c \pm 1) \% 5^{k+1} = 0 \quad (62)$$

509 Suppose  $t$  is a positive integer:

$$510 \quad 2c \pm 1 = t \cdot 5^{k+1}; t \geq 1 \quad (63)$$

511 Since  $2c \pm 1$  is odd,  $t$  is also odd. Because the following conditions exist:

$$\begin{aligned} 513 \quad float : 2c - 1 &\in [2^{24} + 1, 2^{25} - 3]; 2c + 1 \in [2^{24} + 3, 2^{25} - 1]; \\ 514 \quad double : 2c - 1 &\in [2^{53} + 1, 2^{54} - 3]; 2c + 1 \in [2^{53} + 3, 2^{54} - 1]; \end{aligned} \quad (64)$$

516 Therefore, the following satisfies:

$$\begin{aligned} 517 \quad float : 2^{24} + 1 &\leq t \cdot 5^{k+1} \leq 2^{25} - 1 \\ 518 \quad double : 2^{53} + 1 &\leq t \cdot 5^{k+1} \leq 2^{54} - 1 \end{aligned} \quad (65)$$

520 Therefore, the following conclusions are drawn:

$$\begin{aligned} 522 \quad float : \frac{2^{24} + 1}{5^{k+1}} &\leq t \leq \frac{2^{25} - 1}{5^{k+1}}; \\ 523 \quad double : \frac{2^{53} + 1}{5^{k+1}} &\leq t \leq \frac{2^{54} - 1}{5^{k+1}}; \end{aligned} \quad (66)$$

526 For the above equation (66), the maximum value of  $k$  when  $t$  can obtain at least one odd  
 527 number is:

$$\begin{aligned} 528 \quad float : k_{\max} = 9 &\Rightarrow q_{\max} = 33, t = 3 \\ 529 \quad double : k_{\max} = 22 &\Rightarrow q_{\max} = 76, t = 1 \end{aligned} \quad (67)$$

531 Therefore, the maximum value of  $k$  is 9 within the float range and 22 within the double  
 532 range. Therefore, when  $k$  exceeds the above range,  $(2c \pm 1)$  is not divisible by  $5^{k+1}$ .

- 533 •  $1 \geq q \geq 0$

534 Because the denominator  $2^{2+k-q} \cdot 5^{k+1}$  is even and the numerator  $(2c \pm 1)$  is odd, the con-  
 535 dition is not met.

- 536 •  $q < 0$

537 Because the denominator  $2^{2+k-q}$  is even and the numerator  $(2c \pm 1) \cdot 5^{-k-1}$  is odd, the  
 538 condition is not met.

540 In summary, the situations when  $(2c \pm 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is an integer are as follows:

$$\begin{aligned} 541 \quad & \text{float : } 2 \leq q \leq 33 \ \&& \ (2c \pm 1) \% 5^{k+1} = 0; \\ 542 \quad & \text{double : } 2 \leq q \leq 76 \ \&& \ (2c \pm 1) \% 5^{k+1} = 0; \end{aligned} \quad (68)$$

543 And, the range of  $-k - 1$  is:

$$\begin{aligned} 544 \quad & \text{float : } -10 \leq -k - 1 \leq -1 \\ 545 \quad & \text{double : } -23 \leq -k - 1 \leq -1 \end{aligned} \quad (69)$$

546 When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$ , the following conclusions can be drawn:

$$\begin{aligned} 547 \quad & \text{float : } \{2^{35} \cdot 2^q \cdot 10^{-k-1} = 2^{36} \cdot n \Rightarrow \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor\} \\ 548 \quad & \text{double : } \{2^{63} \cdot 2^q \cdot 10^{-k-1} = 2^{64} \cdot n \Rightarrow \lfloor 2^{63} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{64} \cdot n \rfloor\} \end{aligned} \quad (70)$$

549 When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ , the following conclusions can be drawn:

$$\begin{aligned} 550 \quad & \text{float : } \left\{ \begin{array}{l} 2^{35} \cdot 2^q \cdot 10^{-k-1} = 2^{36} - 2^{36} \cdot n \Rightarrow \\ \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor = 2^{36} - 1 - \lfloor 2^{36} \cdot n \rfloor \end{array} \right. \\ 551 \quad & \text{double : } \left\{ \begin{array}{l} 2^{63} \cdot 2^q \cdot 10^{-k-1} = 2^{64} - 2^{64} \cdot n \Rightarrow \\ \lfloor 2^{63} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{64} - 2^{64} \cdot n \rfloor = 2^{64} - 1 - \lfloor 2^{64} \cdot n \rfloor \end{array} \right. \end{aligned} \quad (71)$$

552 The discussion on whether  $\lfloor 2^{36} - 2^{36} \cdot n \rfloor = 2^{36} - 1 - \lfloor 2^{36} \cdot n \rfloor$  in equation (71) holds true, that is, whether  $2^{36} \cdot n$  in equation (71) is an integer, or equivalent to discussing whether the following values are integers when equation (68) holds true (the same applies to double) :

$$\begin{aligned} 553 \quad & \text{float : } 2^{36} \cdot (m + n) = c \cdot 2^{q+36} \cdot 10^{-k-1} = c \cdot 2^{q-k+35} \cdot 5^{-k-1} = c \cdot \frac{2^{q-k+35}}{5^{k+1}} \\ 554 \quad & \text{double : } 2^{64} \cdot (m + n) = c \cdot 2^{q+64} \cdot 10^{-k-1} = c \cdot 2^{q-k+63} \cdot 5^{-k-1} = c \cdot \frac{2^{q-k+63}}{5^{k+1}} \end{aligned} \quad (72)$$

555 Suppose  $c$  can divide  $5^{k+1}$  evenly (where  $t$  is a temporary integer variable):

$$556 \quad c = t \cdot 5^{k+1}; t \geq 1 \quad (73)$$

557 Therefore, when equation (73) was established, there were:

$$558 \quad 2c \pm 1 = 2 \cdot t \cdot 5^{k+1} \pm 1 \quad (74)$$

559 Expression (74) cannot divide  $5^{k+1}$  evenly, which contradicts equation (68), so  $c$  cannot divide  $5^{k+1}$  evenly. Therefore, for float,  $c \cdot 2^{q+36} \cdot 10^{-k-1}$  and  $2^{36} \cdot n$  are not integers; For double,  $c \cdot 2^{q+64} \cdot 10^{-k-1}$  and  $2^{64} \cdot n$  are not integers, that is:

$$\begin{aligned} 560 \quad & \text{float : } \lfloor 2^{36} - 2^{36} \cdot n \rfloor = 2^{36} + \lfloor -2^{36} \cdot n \rfloor = 2^{36} - 1 - \lfloor 2^{36} \cdot n \rfloor \\ 561 \quad & \text{double : } \lfloor 2^{64} - 2^{64} \cdot n \rfloor = 2^{64} + \lfloor -2^{64} \cdot n \rfloor = 2^{64} - 1 - \lfloor 2^{64} \cdot n \rfloor \end{aligned} \quad (75)$$

562 Therefore, the conclusion (71) is correct. Discuss the necessary and sufficient conditions for whether  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor$  is  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$ . The same applies to double, expressed as:

$$\begin{aligned} 563 \quad & \text{float : } 2^{-1} \cdot 2^q \cdot 10^{-k-1} = n \Leftrightarrow \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor \\ 564 \quad & \text{double : } 2^{-1} \cdot 2^q \cdot 10^{-k-1} = n \Leftrightarrow \lfloor 2^{63} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{64} \cdot n \rfloor \end{aligned} \quad (76)$$

589 Similarly, the necessary and sufficient conditions for whether  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor$   
 590 is  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ . The same applies to double, expressed as:

$$\begin{aligned} 591 \quad \text{float : } 2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n &\Leftrightarrow \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor \\ 592 \quad \text{double : } 2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n &\Leftrightarrow \lfloor 2^{63} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{64} - 2^{64} \cdot n \rfloor \end{aligned} \quad (77)$$

594 The sufficient conditions of equations (76) and (77) are obviously established. Introduce the proof  
 595 that equation (76) holds. For float, only the necessary conditions need to be discussed, that is,  
 596 whether  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  must hold true when  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor$  holds, or equivalent  
 597 to  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor \neq \lfloor 2^{36} \cdot n \rfloor$  must hold true when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq n$ . The following is proved  
 598 by proof by contradiction.

599 Assume that  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor$  holds when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq n$ . Then there is:

$$\begin{aligned} 600 \quad \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor &= \lfloor 2^{36} \cdot n \rfloor \\ 601 \quad \Rightarrow 0 < \left| 2^{35} \cdot 2^q \cdot 10^{-k-1} - 2^{36} \cdot n \right| &< 1 \\ 602 \quad \Rightarrow 0 < \left| (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} - m \right| &< 2^{-36} \end{aligned} \quad (78)$$

603 As is known from equation (57), there is:

$$604 \quad m - 1 < (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} < m + 1 \quad (79)$$

605 Suppose the decimal part of  $(2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is represented as  $n^-$ , thus we have:

$$606 \quad \left| (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} - m \right| = \begin{cases} n^-; & \text{if } (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} > m \\ 1 - n^-; & \text{if } (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} < m \end{cases} \quad (80)$$

607 Substitute expression (80) into expression (78), and we get:

$$\begin{aligned} 608 \quad 0 < \left| (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} - m \right| &< 2^{-36} \\ 609 \quad \Rightarrow 0 < n^- < 2^{-36} \text{ or } 0 < 1 - n^- < 2^{-36} \end{aligned} \quad (81)$$

610 Similarly, it can be known that the double range is the range of  $n^-$ . Therefore, there is:

$$\begin{aligned} 611 \quad \text{float : } n^- &\in (0, 2^{-36}) \cup (1 - 2^{-36}, 1) \\ 612 \quad \text{double : } n^- &\in (0, 2^{-64}) \cup (1 - 2^{-64}, 1) \end{aligned} \quad (82)$$

613 When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq n$ , it is known from equation (54) that  $(2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is not an  
 614 integer. Therefore, there is:

$$615 \quad 0 < n^- < 1 \quad (83)$$

616 It is only necessary to prove that equation (82) does not hold. Discuss the range of the decimal  
 617 part  $n^-$  when  $(2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is not an integer. According to equation (61), there are:

$$618 \quad (2c - 1) \cdot 2^{q-1} \cdot 10^{-k-1} = (2c - 1) \cdot \frac{x}{y} = \begin{cases} \frac{(2c-1) \cdot 2^{q-k-2}}{5^{k+1}}; & q \geq 2 \\ \frac{(2c-1)}{2^{2+k-q} \cdot 5^{k+1}}; & 1 \geq q \geq 0 \\ \frac{(2c-1) \cdot 5^{-k-1}}{2^{2+k-q}}; & q < 0 \end{cases} \quad (84)$$

619 The maximum value of  $2c - 1$  is:

$$\begin{aligned} 620 \quad \text{float : } (2c - 1)_{\max} &= 2^{25} - 3 \\ 621 \quad \text{double : } (2c - 1)_{\max} &= 2^{54} - 3 \end{aligned} \quad (85)$$

622 Discuss based on the denominator range in equation (84).

- $y \leq (2c - 1)_{\max}$

When  $y \leq (2c - 1)_{\max}$ ,  $y_{\max}$  is the expression (85), the following holds true:

$$\begin{aligned} \frac{1}{y_{\max}} &\leq n^- \leq 1 - \frac{1}{y_{\max}} \\ \frac{1}{y_{\max}} &\leq 1 - n^- \leq 1 - \frac{1}{y_{\max}} \end{aligned} \quad (86)$$

Therefore, when  $y \leq (2c - 1)_{\max}$ , equation (82) does not hold true.

- $y > (2c - 1)_{\max}$

Call function (37) to calculate the approximation results  $P_* / Q_*$  and  $P^* / Q^*$  of all possible upper and lower limit rational numbers:

$$\left( \frac{P_*}{Q_*}, \frac{P^*}{Q^*} \right) = f((2c - 1)_{\max}, x, y) \quad (87)$$

Therefore, for  $n^-$ , the following conclusion can be drawn from formula (36).

$$n^- \in \left[ \frac{(Q_*x) \% y}{y}, \frac{(Q^*x) \% y}{y} \right] \quad (88)$$

By exhausting all possibilities, we thus have (the test code file is `test3.py`):

$$\begin{aligned} float : 2^{-33} < n^- < 1 - 2^{-29} \\ double : 2^{-62} < n^- < 1 - 2^{-63} \end{aligned} \quad (89)$$

$$\begin{aligned} float : &\left[ \frac{(Q_*x) \% y}{y}, \frac{(Q^*x) \% y}{y} \right] \cap (0, 2^{-36}) = \emptyset \\ &\left[ \frac{(Q_*x) \% y}{y}, \frac{(Q^*x) \% y}{y} \right] \cap (1 - 2^{-36}, 1) = \emptyset \\ double : &\left[ \frac{(Q_*x) \% y}{y}, \frac{(Q^*x) \% y}{y} \right] \cap (0, 2^{-64}) = \emptyset \\ &\left[ \frac{(Q_*x) \% y}{y}, \frac{(Q^*x) \% y}{y} \right] \cap (1 - 2^{-64}, 1) = \emptyset \end{aligned} \quad (90)$$

Therefore, when  $y > (2c - 1)_{\max}$ , equation (82) does not hold true.

In summary, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq n$ , equation (82) does not hold true, that is,  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor \neq \lfloor 2^{36} \cdot n \rfloor$  must hold true. Therefore, when  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} \cdot n \rfloor$  holds,  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  must hold true. Therefore, equation (76) holds.

Similarly, it can be proved that when  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor$  holds,  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$  must hold true. The same applies to double. Similarly, by proof of contradiction, for float, it is assumed that when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq 1 - n$  holds,  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor$  holds. That is:

$$\begin{aligned} \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor &= \lfloor 2^{36} - 2^{36} \cdot n \rfloor \\ \Rightarrow 0 < &\left| 2^{35} \cdot 2^q \cdot 10^{-k-1} - 2^{36} + 2^{36} \cdot n \right| < 1 \\ \Rightarrow 0 < &\left| 2^{q-1} \cdot 10^{-k-1} - 1 + n \right| < 2^{-36} \\ \Rightarrow -2^{-36} < &(2c+1) \cdot 2^{q-1} \cdot 10^{-k-1} - m - 1 < 2^{-36} \end{aligned} \quad (91)$$

687 As is known from equation (58), there is:

$$688 \quad m < (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} < m + 2 \quad (92)$$

690 Suppose the decimal part of  $(2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is represented as  $n^+$ , thus we have:

$$692 \quad (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} - m - 1 = \begin{cases} n^+; & \text{if } (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} > m + 1 \\ 1 - n^+; & \text{if } (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} < m + 1 \end{cases} \quad (93)$$

694 Substitute expression (93) into expression (91), and we get:

$$696 \quad 0 < \left| (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} - m - 1 \right| < 2^{-36} \quad (94)$$

$$697 \quad \Rightarrow 0 < 1 - n^+ < 2^{-36} \text{ or } 0 < n^+ < 2^{-36}$$

699 Similarly, it can be known that the double range is the range of  $n^+$ . Therefore, there is:

$$701 \quad \text{float : } n^+ \in (0, 2^{-36}) \cup (1 - 2^{-36}, 1) \quad (95)$$

$$702 \quad \text{double : } n^+ \in (0, 2^{-64}) \cup (1 - 2^{-64}, 1)$$

704 When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq 1 - n$ , it is known from equation (55) that  $(2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is not  
705 an integer. Therefore, there is:

$$706 \quad 0 < n^+ < 1 \quad (96)$$

708 It is only necessary to prove that equation (95) does not hold. Discuss the range of the decimal  
709 part  $n^+$  when  $(2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1}$  is not an integer. According to equation (61), there are:

$$710 \quad (2c + 1) \cdot 2^{q-1} \cdot 10^{-k-1} = (2c + 1) \cdot \frac{x}{y} = \begin{cases} \frac{(2c+1) \cdot 2^{q-k-2}}{5^{k+1}}; & q \geq 2 \\ \frac{(2c+1)}{2^{2+k-q} \cdot 5^{k+1}}; & 1 \geq q \geq 0 \\ \frac{(2c+1) \cdot 5^{-k-1}}{2^{2+k-q}}; & q < 0 \end{cases} \quad (97)$$

714 The maximum value of  $2c + 1$  is:

$$716 \quad \text{float : } (2c + 1)_{\max} = 2^{25} - 1 \quad (98)$$

$$717 \quad \text{double : } (2c + 1)_{\max} = 2^{54} - 1$$

719 Discuss based on the denominator range in equation (97).

- 720 •  $y \leq (2c + 1)_{\max}$

721 When  $y \leq (2c + 1)_{\max}$ ,  $y_{\max}$  is the expression (98), the following holds true:

$$723 \quad \frac{1}{y_{\max}} \leq n^+ \leq 1 - \frac{1}{y_{\max}}$$

$$724 \quad \frac{1}{y_{\max}} \leq 1 - n^+ \leq 1 - \frac{1}{y_{\max}}$$

$$725 \quad \frac{1}{y_{\max}} \leq 1 - n^+ \leq 1 - \frac{1}{y_{\max}} \quad (99)$$

727 Therefore, when  $y \leq (2c + 1)_{\max}$ , equation (95) does not hold true.

- 728 •  $y > (2c + 1)_{\max}$

729 Call function (37) to calculate the approximation results  $P_* / Q_*$  and  $P^* / Q^*$  of all possible  
730 upper and lower limit rational numbers:

$$733 \quad \left( \frac{P_*}{Q_*}, \frac{P^*}{Q^*} \right) = f((2c + 1)_{\max}, x, y) \quad (100)$$

Therefore, for  $n^+$ , the following conclusion can be drawn from formula (36).

$$n^+ \in \left[ \frac{(Q_*x) \%y}{y}, \frac{(Q^*x) \%y}{y} \right] \quad (101)$$

By exhausting all possibilities, we thus have (the test code file is `test7.py`) :

$$\begin{aligned} float &: 2^{-33} < n^+ < 1 - 2^{-29} \\ double &: 2^{-62} < n^+ < 1 - 2^{-63} \end{aligned} \quad (102)$$

$$\begin{aligned} float &: \left[ \frac{(Q_*x) \%y}{y}, \frac{(Q^*x) \%y}{y} \right] \cap (0, 2^{-36}) = \emptyset \\ &\quad \left[ \frac{(Q_*x) \%y}{y}, \frac{(Q^*x) \%y}{y} \right] \cap (1 - 2^{-36}, 1) = \emptyset \\ double &: \left[ \frac{(Q_*x) \%y}{y}, \frac{(Q^*x) \%y}{y} \right] \cap (0, 2^{-64}) = \emptyset \\ &\quad \left[ \frac{(Q_*x) \%y}{y}, \frac{(Q^*x) \%y}{y} \right] \cap (1 - 2^{-64}, 1) = \emptyset \end{aligned} \quad (103)$$

Therefore, when  $y > (2c + 1)_{\max}$ , equation (95) does not hold true.

In summary, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} \neq 1 - n$ , equation (95) does not hold true, that is,  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor \neq \lfloor 2^{36} - 2^{36} \cdot n \rfloor$  must hold true. Therefore, when  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor 2^{36} - 2^{36} \cdot n \rfloor$  holds,  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$  must hold true. The same is true for double. Therefore, equation (77) holds.

The following conclusions hold:

$$\begin{aligned} float : \lfloor 2^{36} - 2^{36} \cdot n \rfloor &= \begin{cases} 2^{36} - 1 - \lfloor 2^{36} \cdot n \rfloor; & \text{if } c \cdot 2^{36+q} \cdot 10^{-k-1} \notin Z \\ 2^{36} - \lfloor 2^{36} \cdot n \rfloor; & \text{if } c \cdot 2^{36+q} \cdot 10^{-k-1} \in Z \end{cases} \\ double : \lfloor 2^{64} - 2^{64} \cdot n \rfloor &= \begin{cases} 2^{64} - 1 - \lfloor 2^{64} \cdot n \rfloor; & \text{if } c \cdot 2^{64+q} \cdot 10^{-k-1} \notin Z \\ 2^{64} - \lfloor 2^{64} \cdot n \rfloor; & \text{if } c \cdot 2^{64+q} \cdot 10^{-k-1} \in Z \end{cases} \end{aligned} \quad (104)$$

Discuss whether the following equation (105) holds when conditions (68) and (69) are met:

$$\begin{aligned} float &: \lfloor c \cdot \frac{2^{q+35-k}}{5^{k+1}} \rfloor = \lfloor c \cdot \frac{2^{q+35-k}}{5^{k+1}} \cdot r \rfloor \\ &= \lfloor c \cdot \frac{2^{q+35-k}}{5^{k+1}} \cdot \frac{\left(2^{63-\lfloor(-k-1)\cdot\log_2(10)\rfloor}/10^{k+1}\right)+1}{10^{-k-1}} \cdot 2^{\lfloor(-k-1)\cdot\log_2(10)\rfloor-63} \rfloor \\ double &: \lfloor c \cdot \frac{2^{q+63-k}}{5^{k+1}} \rfloor = \lfloor c \cdot \frac{2^{q+63-k}}{5^{k+1}} \cdot r \rfloor \\ &= \lfloor c \cdot \frac{2^{q+63-k}}{5^{k+1}} \cdot \frac{\left(2^{127-\lfloor(-k-1)\cdot\log_2(10)\rfloor}/10^{k+1}\right)+1}{10^{-k-1}} \cdot 2^{\lfloor(-k-1)\cdot\log_2(10)\rfloor-127} \rfloor \end{aligned} \quad (105)$$

There are:

$$\begin{aligned} float &: \lfloor c \cdot \frac{2^{q+35-k}}{5^{k+1}} \rfloor = \lfloor 2^{36} \cdot (m+n) \rfloor = 2^{36} \cdot m + \lfloor 2^{36} \cdot n \rfloor \\ double &: \lfloor c \cdot \frac{2^{q+63-k}}{5^{k+1}} \rfloor = \lfloor 2^{64} \cdot (m+n) \rfloor = 2^{64} \cdot m + \lfloor 2^{64} \cdot n \rfloor \end{aligned} \quad (106)$$

It has been proven earlier that  $m$  can be accurately calculated. Then, when (105) holds true, the values  $\lfloor 2^{36} \cdot n \rfloor$  and  $\lfloor 2^{64} \cdot n \rfloor$  on the right side of equations (70) and (71) can be accurately calculated.

785 From equation (63), we have:

$$786 \quad 787 \quad 788 \quad c = \frac{t \cdot 5^{k+1} - 1}{2} \quad (107)$$

789 Substituting equation (107) into equation (105), we have:

$$790 \quad 791 \quad 792 \quad 793 \quad 794 \quad 795 \quad \begin{aligned} float : c \cdot \frac{2^{q+35-k}}{5^{k+1}} &= t \cdot 2^{q+34-k} - \frac{2^{q+34-k}}{5^{k+1}} \\ double : c \cdot \frac{2^{q+62-k}}{5^{k+1}} &= t \cdot 2^{q+62-k} - \frac{2^{q+62-k}}{5^{k+1}} \end{aligned} \quad (108)$$

796 When conditions (68) and (69) are met,  $t \cdot 2^{q+34-k}$  and  $t \cdot 2^{q+62-k}$  are integers. Under the condition  
797 of meeting condition (68), the decimal part of expression (108) is represented as:

$$798 \quad 799 \quad 800 \quad 801 \quad 802 \quad 803 \quad \begin{aligned} float : \frac{2^{q+34-k} \% 5^{k+1}}{5^{k+1}}; 2 \leq q \leq 33 \\ double : \frac{2^{q+62-k} \% 5^{k+1}}{5^{k+1}}; 2 \leq q \leq 76 \end{aligned} \quad (109)$$

804 It is only necessary to prove that the increase in the value  $c \cdot \frac{2^{q+35-k}}{5^{k+1}} \cdot r$  on the right side of the  
805 expression compared to the value  $c \cdot \frac{2^{q+35-k}}{5^{k+1}}$  on the left side plus the decimal part of the value on  
806 the left side is less than 1 for equation (105) to hold true. That is:

$$807 \quad 808 \quad 809 \quad 810 \quad 811 \quad 812 \quad 813 \quad \begin{aligned} float : \frac{2^{q+34-k} \% 5^{k+1}}{5^{k+1}} + \left( c \cdot \frac{2^{q+35-k}}{5^{k+1}} \cdot r - c \cdot \frac{2^{q+35-k}}{5^{k+1}} \right) < 1 \\ double : \frac{2^{q+62-k} \% 5^{k+1}}{5^{k+1}} + \left( c \cdot \frac{2^{q+63-k}}{5^{k+1}} \cdot r - c \cdot \frac{2^{q+63-k}}{5^{k+1}} \right) < 1 \end{aligned} \quad (110)$$

814 By exhaustional calculating the maximum possible  $c$  value under each  $q$  and substituting it into  
815 equation (110), it holds. The calculation result is in [test2.py](#). The calculation results show that for  
816 the float range and the double range, equation (110) always holds true. Therefore, equation (105)  
817 holds true, and thus the values of  $\lfloor 2^{36} \cdot n \rfloor$  and  $\lfloor 2^{64} \cdot n \rfloor$  on the right side of equations (70) and (71)  
818 can be accurately calculated. The values of  $\lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor$  and  $\lfloor 2^{63} \cdot 2^q \cdot 10^{-k-1} \rfloor$  on the left side  
819 of equations (70) and (71) can be calculated through lookup tables.

$$820 \quad 821 \quad 822 \quad 823 \quad 824 \quad \begin{aligned} float : \lfloor 2^{35} \cdot 2^q \cdot 10^{-k-1} \rfloor &= pow10 \gg (28 - q - \lfloor (-k - 1) \cdot \log_2(10) \rfloor) \\ double : \lfloor 2^{64} \cdot 2^q \cdot 10^{-k-1} \rfloor &= pow10 \gg (64 - q - \lfloor (-k - 1) \cdot \log_2(10) \rfloor) \end{aligned} \quad (111)$$

825 The code file for verifying the validity of equation (111) is [test4.py](#). Therefore, when conditions (68)  
826 and (69) are met, the values of both sides of equations (70) and (71) can be accurately calculated.

827 Discuss the relationship between the following two values within all ranges of floating-point  
828 numbers:

$$829 \quad 830 \quad 831 \quad 832 \quad 833 \quad \begin{aligned} float : \lfloor c \cdot 2^{q+36} \cdot 10^{-k-1} \rfloor; \lfloor c \cdot 2^{q+36} \cdot r \cdot 10^{-k-1} \rfloor; \\ double : \lfloor c \cdot 2^{q+64} \cdot 10^{-k-1} \rfloor; \lfloor c \cdot 2^{q+64} \cdot r \cdot 10^{-k-1} \rfloor; \end{aligned} \quad (112)$$

When  $r = 1$ , it is obvious that the two values in expression (112) are equal. When  $r \neq 1$ , or equivalent to  $r > 1$ , has:

*float :*

$$\begin{aligned} c \cdot 2^{q+36} \cdot r \cdot 10^{-k-1} &= c \cdot 2^{q+36} \cdot 10^{-k-1} + c \cdot 2^{q+36} \cdot (r-1) \cdot 10^{-k-1} \\ &< c \cdot 2^{q+36} \cdot 10^{-k-1} + 2^{24} \cdot 2^{36} \cdot 2^q \cdot 10^{-k-1} \cdot (r-1) \\ &< c \cdot 2^{q+36} \cdot 10^{-k-1} + 2^{-3} \end{aligned}$$

$$\lfloor c \cdot 2^{q+36} \cdot r \cdot 10^{-k-1} \rfloor \leq \lfloor c \cdot 2^{q+36} \cdot 10^{-k-1} \rfloor + 1 \quad (113)$$

*double :*

$$\begin{aligned} c \cdot 2^{q+64} \cdot r \cdot 10^{-k-1} &= c \cdot 2^{q+64} \cdot 10^{-k-1} + c \cdot 2^{q+64} \cdot (r-1) \cdot 10^{-k-1} \\ &< c \cdot 2^{q+64} \cdot 10^{-k-1} + 2^{53} \cdot 2^{64} \cdot 2^q \cdot 10^{-k-1} \cdot (r-1) \\ &< c \cdot 2^{q+64} \cdot 10^{-k-1} + 2^{-10} \end{aligned}$$

$$\lfloor c \cdot 2^{q+64} \cdot r \cdot 10^{-k-1} \rfloor \leq \lfloor c \cdot 2^{q+64} \cdot 10^{-k-1} \rfloor + 1$$

Therefore, there is:

$$\text{float : } 0 \leq \lfloor c \cdot 2^{q+36} \cdot r \cdot 10^{-k-1} \rfloor - \lfloor c \cdot 2^{q+36} \cdot 10^{-k-1} \rfloor \leq 1 \quad (114)$$

$$\text{double : } 0 \leq \lfloor c \cdot 2^{q+64} \cdot r \cdot 10^{-k-1} \rfloor - \lfloor c \cdot 2^{q+64} \cdot 10^{-k-1} \rfloor \leq 1$$

Because there is:

$$\lfloor c \cdot 2^q \cdot 10^{-k-1} \rfloor = \lfloor c \cdot 2^q \cdot r \cdot 10^{-k-1} \rfloor = m \quad (115)$$

$$\text{float : } \lfloor c \cdot 2^{q+36} \cdot 10^{-k-1} \rfloor = 2^{36} \cdot m + \lfloor 2^{36} \cdot n \rfloor \quad (116)$$

$$\text{double : } \lfloor c \cdot 2^{q+64} \cdot 10^{-k-1} \rfloor = 2^{64} \cdot m + \lfloor 2^{64} \cdot n \rfloor$$

Suppose:

$$n_r = c \cdot 2^q \cdot r \cdot 10^{-k-1} - m \quad (117)$$

Therefore, the following conclusion can be drawn: when condition (68) is met, from equation (105), we have:

$$\text{float : } 2 \leq q \leq 33 \& (2c \pm 1) \% 5^{k+1} = 0 \Rightarrow \lfloor 2^{36} \cdot n \rfloor = \lfloor 2^{36} \cdot n_r \rfloor \quad (118)$$

$$\text{double : } 2 \leq q \leq 76 \& (2c \pm 1) \% 5^{k+1} = 0 \Rightarrow \lfloor 2^{64} \cdot n \rfloor = \lfloor 2^{64} \cdot n_r \rfloor$$

Within the range of floating-point numbers, there exists:

$$\text{float : } \lfloor 2^{36} \cdot n \rfloor \leq \lfloor 2^{36} \cdot n_r \rfloor \leq \lfloor 2^{36} \cdot n \rfloor + 1 \quad (119)$$

$$\text{double : } \lfloor 2^{64} \cdot n \rfloor \leq \lfloor 2^{64} \cdot n_r \rfloor \leq \lfloor 2^{64} \cdot n \rfloor + 1$$

To simplify the expression, *even* is used to indicate whether  $c$  is an even number:

$$\text{even} = (c + 1) \% 2 \in \{0, 1\} \quad (120)$$

When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ ,  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  is the boundary condition for *one* = 0, and  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$  is the boundary condition for *one* = 10. Whether *one* is 0 or 10 is determined based on whether  $c$  is an even number. Therefore, the following exists:

$$\text{float : } \begin{cases} \text{one} = 0 : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \text{even} > \lfloor 2^{36} \cdot n_r \rfloor \\ \text{one} = 10 : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \text{even} > 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor \end{cases} \quad (121)$$

$$\text{double : } \begin{cases} \text{one} = 0 : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \text{even} > \lfloor 2^{64} \cdot n_r \rfloor \\ \text{one} = 10 : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \text{even} > 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor \end{cases}$$

Therefore, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ , we can use the condition (122) to determine whether  $one = 0$  or  $one = 10$ .

$$\begin{aligned} & float : \begin{cases} \text{if } \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even > \lfloor 2^{36} \cdot n_r \rfloor : one = 0 \\ \text{if } \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even > 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor : one = 10 \end{cases} \\ & double : \begin{cases} \text{if } \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even > \lfloor 2^{64} \cdot n_r \rfloor : one = 0 \\ \text{if } \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even > 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor : one = 10 \end{cases} \end{aligned} \quad (122)$$

When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n$ , We can also use the above condition (122) to determine whether  $one = 0$  or  $one = 10$ . When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < 1 - n$ , we can also use the above condition (122) to determine whether  $one \neq 0$  or  $one \neq 10$ . There are a total of four situations. The proof is as follows:

(1) When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < n$ , there must exist  $one \neq 0$ , and there is:

$$\begin{aligned} & float : 2^{-1} \cdot 2^q \cdot 10^{-k-1} - n = n^- - 1 \in (2^{-33} - 1, -2^{-29}) \\ & double : 2^{-1} \cdot 2^q \cdot 10^{-k-1} - n = n^- - 1 \in (2^{-62} - 1, -2^{-63}) \end{aligned} \quad (123)$$

Therefore, the following exists:

$$\begin{aligned} & float : 2^{q+35} \cdot 10^{-k-1} - 2^{36} \cdot n \in (2^3 - 2^{36}, -2^7) \\ & double : 2^{q+63} \cdot 10^{-k-1} - 2^{64} \cdot n \in (4 - 2^{64}, -2) \end{aligned} \quad (124)$$

Suppose there are two real numbers  $a$  and  $b$ , and the following relationship must exist:

$$\begin{aligned} & 0 \leq b - \lfloor b \rfloor < 1 \\ & a - \lfloor a \rfloor - 1 < b - \lfloor b \rfloor < 1 + a - \lfloor a \rfloor \\ & a - b - 1 < \lfloor a \rfloor - \lfloor b \rfloor < a - b + 1 \end{aligned} \quad (125)$$

When  $a = 2^{q+35} \cdot 10^{-k-1}$  and  $b = 2^{36} \cdot n$  or  $a = 2^{q+63} \cdot 10^{-k-1}$  and  $b = 2^{64} \cdot n$ , the following exists:

$$\begin{aligned} & float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor - \lfloor 2^{36} \cdot n \rfloor < 2^{q+35} \cdot 10^{-k-1} - 2^{36} \cdot n + 1 \\ & double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor - \lfloor 2^{64} \cdot n \rfloor < 2^{q+63} \cdot 10^{-k-1} - 2^{64} \cdot n + 1 \end{aligned} \quad (126)$$

From equation (124), we have:

$$\begin{aligned} & float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor - \lfloor 2^{36} \cdot n \rfloor < 1 - 2^7 < 0 \\ & double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor - \lfloor 2^{64} \cdot n \rfloor < 1 - 2 < 0 \end{aligned} \quad (127)$$

Therefore, there is:

$$\begin{aligned} & float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even \leq \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + 1 \\ & \quad < \lfloor 2^{36} \cdot n \rfloor \leq \lfloor 2^{36} \cdot n_r \rfloor \\ & \Rightarrow \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even < \lfloor 2^{36} \cdot n_r \rfloor \\ & double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even \leq \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + 1 \\ & \quad < \lfloor 2^{64} \cdot n \rfloor \leq \lfloor 2^{64} \cdot n_r \rfloor \\ & \Rightarrow \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even < \lfloor 2^{64} \cdot n_r \rfloor \end{aligned} \quad (128)$$

Therefore, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < n$ , the condition (122) can be used to determine that  $one \neq 0$ .

(2) When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > n$ , there must exist  $one = 0$ , and there is:

$$\begin{aligned} & float : 2^{-1} \cdot 2^q \cdot 10^{-k-1} - n = n^- \in (2^{-33}, 1 - 2^{-29}) \\ & double : 2^{-1} \cdot 2^q \cdot 10^{-k-1} - n = n^- \in (2^{-62}, 1 - 2^{-63}) \end{aligned} \quad (129)$$

932 Therefore, the following exists:

$$\begin{aligned} 933 \quad \text{float} : 2^{q+35} \cdot 10^{-k-1} - 2^{36} \cdot n &\in (2^3, 2^{36} - 2^7) \\ 934 \quad \text{double} : 2^{q+63} \cdot 10^{-k-1} - 2^{64} \cdot n &\in (4, 2^{64} - 2) \end{aligned} \quad (130)$$

936 When  $a = 2^{q+35} \cdot 10^{-k-1}$  and  $b = 2^{36} \cdot n$  or  $a = 2^{q+63} \cdot 10^{-k-1}$  and  $b = 2^{64} \cdot n$ , from equation (125),  
937 the following exists:

$$\begin{aligned} 938 \quad \text{float} : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor - \lfloor 2^{36} \cdot n \rfloor &> 2^{q+35} \cdot 10^{-k-1} - 2^{36} \cdot n - 1 \\ 939 \quad \text{double} : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor - \lfloor 2^{64} \cdot n \rfloor &> 2^{q+63} \cdot 10^{-k-1} - 2^{64} \cdot n - 1 \end{aligned} \quad (131)$$

942 From equation (130), we have:

$$\begin{aligned} 943 \quad \text{float} : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor - \lfloor 2^{36} \cdot n \rfloor &> 2^3 - 1 \geq 0 \\ 944 \quad \text{double} : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor - \lfloor 2^{64} \cdot n \rfloor &> 4 - 1 \geq 0 \end{aligned} \quad (132)$$

946 Therefore, there is:

$$\begin{aligned} 947 \quad \text{float} : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \text{even} &\geq \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor \\ 948 \quad &> \lfloor 2^{36} \cdot n \rfloor + 1 \geq \lfloor 2^{36} \cdot n_r \rfloor \\ 949 \quad \Rightarrow \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \text{even} &> \lfloor 2^{36} \cdot n_r \rfloor \\ 950 \quad \text{double} : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \text{even} &\geq \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor \\ 951 \quad &> \lfloor 2^{64} \cdot n \rfloor + 1 \geq \lfloor 2^{64} \cdot n_r \rfloor \\ 952 \quad \Rightarrow \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \text{even} &> \lfloor 2^{64} \cdot n_r \rfloor \end{aligned} \quad (133)$$

956 Therefore, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > n$ , the condition (122) can be used to determine that  $\text{one} = 0$ .  
957 (3) When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < 1 - n$ , there must exist  $\text{one} \neq 10$ , and there is:

$$\begin{aligned} 958 \quad \text{float} : 2^{-1} \cdot 2^q \cdot 10^{-k-1} + n &= n^+ \in (2^{-33}, 1 - 2^{-29}) \\ 959 \quad \text{double} : 2^{-1} \cdot 2^q \cdot 10^{-k-1} + n &= n^+ \in (2^{-62}, 1 - 2^{-63}) \end{aligned} \quad (134)$$

962 Therefore, the following exists:

$$\begin{aligned} 963 \quad \text{float} : 2^{q+35} \cdot 10^{-k-1} + 2^{36} \cdot n &\in (2^3, 2^{36} - 2^7) \\ 964 \quad \text{double} : 2^{q+63} \cdot 10^{-k-1} + 2^{64} \cdot n &\in (4, 2^{64} - 2) \end{aligned} \quad (135)$$

966 Suppose there are two real numbers  $a$  and  $b$ , and the following relationship must exist:

$$\begin{aligned} 967 \quad a - 1 < \lfloor a \rfloor \leq a \\ 968 \quad b - 1 < \lfloor b \rfloor \leq b \\ 969 \quad a + b - 2 < \lfloor a \rfloor + \lfloor b \rfloor \leq a + b \end{aligned} \quad (136)$$

971 When  $a = 2^{q+35} \cdot 10^{-k-1}$  and  $b = 2^{36} \cdot n$  or  $a = 2^{q+63} \cdot 10^{-k-1}$  and  $b = 2^{64} \cdot n$ , the following exists:

$$\begin{aligned} 972 \quad \text{float} : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \lfloor 2^{36} \cdot n \rfloor &\leq 2^{q+35} \cdot 10^{-k-1} + 2^{36} \cdot n \\ 973 \quad \text{double} : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \lfloor 2^{64} \cdot n \rfloor &\leq 2^{q+63} \cdot 10^{-k-1} + 2^{64} \cdot n \end{aligned} \quad (137)$$

976 From equation (135), we have:

$$\begin{aligned} 977 \quad \text{float} : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \lfloor 2^{36} \cdot n \rfloor &< 2^{36} - 2^7 \\ 978 \quad \text{double} : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \lfloor 2^{64} \cdot n \rfloor &< 2^{64} - 2 \end{aligned} \quad (138)$$

Therefore, there is:

$$\begin{aligned}
 982 \quad float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even &\leq \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + 1 \\
 983 \quad &< 2^{36} - 2 - \lfloor 2^{36} \cdot n \rfloor \\
 984 \quad &< 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor \\
 985 \quad \Rightarrow \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even &< 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor \\
 986 \quad double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even &\leq \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + 1 \tag{139} \\
 987 \quad &< 2^{64} - 2 - \lfloor 2^{64} \cdot n \rfloor \\
 988 \quad &< 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor \\
 989 \quad \Rightarrow \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even &< 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor
 \end{aligned}$$

Therefore, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < 1 - n$ , the condition (122) can be used to determine that  $one \neq 10$ .  
(4) When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n$ , there must exist  $one = 10$ , and there is:

$$\begin{aligned}
 996 \quad float : 2^{-1} \cdot 2^q \cdot 10^{-k-1} + n &= n^+ + 1 \in (1 + 2^{-33}, 2 - 2^{-29}) \\
 997 \quad double : 2^{-1} \cdot 2^q \cdot 10^{-k-1} + n &= n^+ + 1 \in (1 + 2^{-62}, 2 - 2^{-63}) \tag{140}
 \end{aligned}$$

Therefore, the following exists:

$$\begin{aligned}
 1000 \quad float : 2^{q+35} \cdot 10^{-k-1} + 2^{36} \cdot n &\in (2^3 + 2^{36}, 2^{37} - 2^7) \\
 1001 \quad double : 2^{q+63} \cdot 10^{-k-1} + 2^{64} \cdot n &\in (4 + 2^{64}, 2^{65} - 2) \tag{141}
 \end{aligned}$$

When  $a = 2^{q+35} \cdot 10^{-k-1}$  and  $b = 2^{36} \cdot n$  or  $a = 2^{q+63} \cdot 10^{-k-1}$  and  $b = 2^{64} \cdot n$ , from equation (136), the following exists:

$$\begin{aligned}
 1006 \quad float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \lfloor 2^{36} \cdot n \rfloor &> 2^{q+35} \cdot 10^{-k-1} + 2^{36} \cdot n - 2 \\
 1007 \quad double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \lfloor 2^{64} \cdot n \rfloor &> 2^{q+63} \cdot 10^{-k-1} + 2^{64} \cdot n - 2 \tag{142}
 \end{aligned}$$

From equation (141), we have:

$$\begin{aligned}
 1010 \quad float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + \lfloor 2^{36} \cdot n \rfloor &> 2^{36} + 2^3 - 2 > 2^{36} \\
 1011 \quad double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + \lfloor 2^{64} \cdot n \rfloor &> 2^{64} + 2 - 2 > 2^{64} \tag{143}
 \end{aligned}$$

Therefore, there is:

$$\begin{aligned}
 1014 \quad float : \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even &\geq \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor \\
 1015 \quad &> 2^{36} - \lfloor 2^{36} \cdot n \rfloor \\
 1016 \quad &> 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor \\
 1017 \quad \Rightarrow \lfloor 2^{q+35} \cdot 10^{-k-1} \rfloor + even &> 2^{36} - 1 - \lfloor 2^{36} \cdot n_r \rfloor \\
 1018 \quad double : \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even &\geq \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor \tag{144} \\
 1019 \quad &> 2^{64} - \lfloor 2^{64} \cdot n \rfloor \\
 1020 \quad &> 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor \\
 1021 \quad \Rightarrow \lfloor 2^{q+63} \cdot 10^{-k-1} \rfloor + even &> 2^{64} - 1 - \lfloor 2^{64} \cdot n_r \rfloor
 \end{aligned}$$

Therefore, when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n$ , the condition (122) can be used to determine that  $one = 10$ .

From the above proof, it can be seen that when condition (68) is met, the condition (122) can be used to determine whether  $one = 0$  or  $one = 10$  when  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} = 1 - n$ .

1030 When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} > 1 - n$ , the condition (122) can be used to determine  
 1031 whether  $one = 0$  or  $one = 10$ . When  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < n$  or  $2^{-1} \cdot 2^q \cdot 10^{-k-1} < 1 - n$ , the condition  
 1032 (122) can be used to determine whether  $one \neq 0$  or  $one \neq 10$ .

1033 The proof process of this section is completed. In the code implementation, the two judgment  
 1034 conditions can be quickly calculated using addition and subtraction shift operations, and can be  
 1035 compiled by the compiler into cmov instructions, thereby reducing the impact of branch prediction  
 1036 failure on performance.

### 1037 3.5 Determine whether $one = \lfloor 10n \rfloor$ or $one = \lfloor 10n \rfloor + 1$

1038 Determine whether  $one$  is  $\lfloor 10n \rfloor$  or  $\lfloor 10n \rfloor + 1$  based on the decimal part of  $10n$ . There are two cases:  
 1039 the decimal part of  $10n$  is 0.5 and it is not 0.5.

1040 3.5.1  $10n - \lfloor 10n \rfloor = 0.5$ . When the decimal part of  $10n$  is 0.5, there must be:

$$\begin{aligned} 1041 \quad 10n - \lfloor 10n \rfloor &= 0.5 \\ 1042 \quad \Rightarrow 10 \cdot c \cdot 2^q \cdot 10^{-k-1} - \lfloor 10 \cdot c \cdot 2^q \cdot 10^{-k-1} \rfloor &= 0.5 \\ 1043 \quad \Rightarrow c \cdot 2^q \cdot 10^{-k} - \lfloor c \cdot 2^q \cdot 10^{-k} \rfloor &= 0.5 \\ 1044 \quad \Rightarrow c \cdot 2^q \cdot 10^{-k} &= \lfloor c \cdot 2^q \cdot 10^{-k} \rfloor + 0.5 \\ 1045 \quad \Rightarrow 2c \cdot 2^q \cdot 10^{-k} &= 2\lfloor c \cdot 2^q \cdot 10^{-k} \rfloor + 1 \end{aligned} \tag{145}$$

1046 So  $2c \cdot 2^q \cdot 10^{-k}$  is an odd number. Then the following expression is odd:

$$c \cdot 2^{q+1} \cdot 10^{-k} = c \cdot 2^{q-k+1} \cdot 5^{-k} \tag{146}$$

1047 According to the range of  $q$ , there are:

$$c \cdot 2^{q+1} \cdot 10^{-k} = \begin{cases} \frac{c \cdot 2^{q-k+1}}{5^k}; q \geq 0 \\ c \cdot 2 \cdot 5^{-k}; q = -1 \\ \frac{c \cdot 5^{-k}}{2^{k-q-1}}; q \leq -2 \end{cases} \tag{147}$$

1048 According to the range of  $q$ , the following situations are discussed:

- $q \geq 0$

1049 When  $q \geq 0$ , it can be concluded that  $q - k + 1 \geq 1$ , the numerator  $c \cdot 2^{q-k+1}$  is even and  
 1050 the denominator  $5^k$  is odd, which does not meet the condition.

- $q = -1$

1051 When  $q = -1$ , it can be concluded that  $c \cdot 2 \cdot 5^{-k}$  is even, which does not meet the condition.

- $q \leq -2$

1052  $5^{-k}$  is an odd number.  $c$  is an odd multiple of  $2^{k-q-1}$ . So:

$$\begin{aligned} 1053 \quad float : c \geq 2^{k-q-1} \Rightarrow k - q - 1 &\leq 22 \Rightarrow q \geq -34 \\ 1054 \quad double : c \geq 2^{k-q-1} \Rightarrow k - q - 1 &\leq 51 \Rightarrow q \geq -75 \end{aligned} \tag{148}$$

1055 Therefore, when  $q$  meets the above conditions,  $c$  must be an odd multiple of  $2^{k-q-1}$  to meet  
 1056 the condition. Therefore, when the following conditions are met, expression (146) is an odd  
 1057 number:

$$\begin{aligned} 1058 \quad float : -34 \leq q \leq -2 \&& c \% 2^{k-q} = 2^{k-q-1} \\ 1059 \quad double : -75 \leq q \leq -2 \&& c \% 2^{k-q} = 2^{k-q-1} \end{aligned} \tag{149}$$

1060 When  $q$  is within the above range (149),  $r = 1$  is derived from equation (30). Therefore, there  
 1061 is:

$$n_r = n \tag{150}$$

1079 The following equation holds:

$$1080 \quad 1081 \quad 20m + 20n = c \cdot 2^q \cdot 10^{-k+1} = c \cdot 2^{q-k+1} \cdot 5^{-k} = \frac{c}{2^{k-q-1}} \cdot 5^{-k} \quad (151)$$

1082 Since  $-k \geq 1$ ,  $5^{-k}$  is multiple of 5 and is an odd number. Since  $\frac{c}{2^{k-q-1}}$  and  $5^{-k}$  are both  
1083 odd numbers,  $20m$  is an even number,  $20n$  is multiple of 5 and is an odd number. Therefore,  
1084 there is:

$$1085 \quad 20n \in \{5, 15\} \\ 1086 \quad \Rightarrow n \in \{0.25, 0.75\} \\ 1087 \quad \Rightarrow n_r \in \{0.25, 0.75\} \quad (152)$$

1089 The result of *one* is an even number between  $\lfloor 10n \rfloor$  and  $\lfloor 10n \rfloor + 1$ . Therefore, when the  
1090 following conditions are met:

$$1091 \quad 1092 \quad one = \begin{cases} \lfloor 10n \rfloor = 2, \text{if } n = 0.25 \\ \lfloor 10n \rfloor + 1 = 8, \text{if } n = 0.75 \end{cases} \Rightarrow one = \lfloor 20n + 1 \rfloor // 2 - (n = 0.25 ? 1 : 0) \quad (153)$$

1094 3.5.2  $10n - \lfloor 10n \rfloor \neq 0.5$ . When the decimal part of  $10n$  is not 0.5, round to the nearest integer  
1095 value based on the decimal part of  $10n$ . Therefore, there is:

$$1096 \quad 1097 \quad one = \begin{cases} \lfloor 10n \rfloor, \text{if } 10n - \lfloor 10n \rfloor < 0.5 \\ \lfloor 10n \rfloor + 1, \text{if } 10n - \lfloor 10n \rfloor > 0.5 \end{cases} \Rightarrow one = \lfloor 10n + 0.5 \rfloor = \lfloor 20n + 1 \rfloor // 2 \quad (154)$$

1099 Since  $\lfloor 20n + 1 \rfloor = \lfloor 20n \rfloor + 1$ , it is only necessary to accurately calculate the value of  $\lfloor 20n \rfloor$ . And,  
1100 there is:

$$1101 \quad d = ten + one \\ 1102 \quad = 10m + \lfloor 20n + 1 \rfloor // 2 \\ 1103 \quad = (\lfloor 20m + 20n \rfloor + 1) // 2 \quad (155)$$

1105 Suppose there are:

$$1106 \quad 1107 \quad 20m + 20n = c \cdot 2^{q+1} \cdot 10^{-k} = c \cdot 2^{q-k+1} \cdot 5^{-k} = c \cdot \frac{x}{y} \quad (156)$$

1108 Suppose the decimal part of  $20n$  is  $n_{20}$ .

1109 When  $y \leq c_{\max} = C$ , the range of the decimal part must include:

$$1110 \quad 1111 \quad float : \frac{1}{2^{24}-1} = \frac{1}{C} \leq n_{20} \leq 1 - \frac{1}{C} = \frac{2^{24}-2}{2^{24}-1} \\ 1112 \quad 1113 \quad double : \frac{1}{2^{53}-1} = \frac{1}{C} \leq n_{20} \leq 1 - \frac{1}{C} = \frac{2^{53}-2}{2^{53}-1} \quad (157)$$

1114 When  $y > c_{\max} = C$ , the range of the decimal part must include(the test file is [test5.py](#)):

$$1115 \quad 1116 \quad float : 2^{-32} < n_{20} < 1 - 2^{-30} \\ 1117 \quad 1118 \quad double : 2^{-64} < n_{20} < 1 - 2^{-62} \quad (158)$$

1119 Therefore, the range of  $n_{20}$  satisfies equation (158). In the code implementation, for float, only the  
1120 high 36 bits of  $n_r$  are retained, and for double, only the high 70 bits of  $n_r$  are retained. Suppose  
1121 the discarded part of a float is represented as  $n_{36}$ , and similarly, the discarded part of a double is  
1122 represented as  $n_{70}$ . Therefore, there is:

$$1123 \quad 1124 \quad float : n_{36} \in [0, 2^{-36}) \\ 1125 \quad 1126 \quad double : n_{70} \in [0, 2^{-70}) \quad (159)$$

1128 Calculate the boundary conditions of the following expression:

$$\begin{aligned} 1129 \quad \text{float : } F &= 20 \cdot \left( c \cdot 2^q \cdot r \cdot 10^{-k-1} - n_{36} \right) \\ 1130 \quad \text{double : } F &= 20 \cdot \left( c \cdot 2^q \cdot r \cdot 10^{-k-1} - n_{70} \right) \end{aligned} \quad (160)$$

1133 Therefore, there is:

$$\begin{aligned} 1135 \quad \text{float : } F_{\min} &> 20 \cdot \left( c \cdot 2^q \cdot 10^{-k-1} - 2^{-36} \right) \\ 1136 \quad &= 20m + 20n - 20 \cdot 2^{-36} \\ 1138 \quad F_{\max} &< 20 \cdot \left( c \cdot 2^q \cdot (1 + 2^{-63}) \cdot 10^{-k-1} - 0 \right) \\ 1139 \quad &< 20m + 20n + 20 \cdot 2^{-63} \cdot c \\ 1140 \quad &< 20m + \lfloor 20n \rfloor + 1 \\ 1142 \quad \text{double : } F_{\min} &> 20 \cdot \left( c \cdot 2^q \cdot 10^{-k-1} - 2^{-70} \right) \\ 1143 \quad &= 20m + 20n - 20 \cdot 2^{-70} \\ 1144 \quad &> 20m + \lfloor 20n \rfloor \\ 1147 \quad F_{\max} &< 20 \cdot \left( c \cdot 2^q \cdot (1 + 2^{-127}) \cdot 10^{-k-1} - 0 \right) \\ 1148 \quad &< 20m + 20n + 20 \cdot 2^{-127} \cdot c \\ 1149 \quad &< 20m + \lfloor 20n \rfloor + 1 \\ 1151 \end{aligned} \quad (161)$$

1152 Therefore, there is:

$$\begin{aligned} 1153 \quad \text{float : } \lfloor F \rfloor &= 20m + \lfloor 20n \rfloor \\ 1154 \quad \text{double : } \lfloor F \rfloor &= 20m + \lfloor 20n \rfloor \end{aligned} \quad (162)$$

1156 In fact, in the above proof process, for float,  $\lfloor F_{\min} \rfloor \neq 20m + \lfloor 20n \rfloor$  may exist, but the code implementation has passed the exhaustive test, so this not-so-perfect proof process can be ignored.  
1157 Therefore, the calculation of  $d$  can be simplified as follows:  
1158

$$\begin{aligned} 1159 \quad d &= \text{ten} + \text{one} \\ 1160 \quad &= (\lfloor F \rfloor + 1) // 2 \\ 1161 \quad &= (\lfloor 20 \cdot (c \cdot 2^q \cdot r \cdot 10^{-k-1} - n_x) \rfloor + 1) // 2 \end{aligned} \quad (163)$$

1163 For the float range,  $n_x = n_{36}$ ; for the double range,  $n_x = n_{70}$ .

1164 For double, quickly determine that  $n == 0.25$  in equation (153).

1165 When  $n = 0.25$ ,  $\lfloor 2^{64} \cdot n_r \rfloor = \lfloor 2^{64} \cdot n \rfloor = 2^{62}$ . Therefore, the following condition can be used to quickly  
1166 determine whether  $n = 0.25$ :

$$1168 \quad \text{double : } n = 0.25 \text{ if } \lfloor 2^{64} \cdot n_r \rfloor = 2^{62} \quad (164)$$

1169 When  $n \neq 0.25$ , calculate the range of the decimal part of the following expression:

$$1171 \quad 4m + 4n = c \cdot 2^{q+2} \cdot 10^{-k-1} \quad (165)$$

1173 Therefore, when equation (165) is not an integer, we have(the test file is [test6.py](#)):  
1174

$$1175 \quad 2^{-62} < 4n - \lfloor 4n \rfloor < 1 - 2^{-62} \quad (166)$$

1177 Calculate the two boundary cases of  $4n$  that are closest to 1:

$$\begin{aligned} 1178 \quad [4n] &= 0 \Rightarrow 4n - 0 < 1 - 2^{-62} \Rightarrow \lfloor 2^{64} \cdot n \rfloor \leq 2^{62} - 2 \\ 1179 \quad [4n] &= 1 \Rightarrow 4n - 1 > 2^{-62} \Rightarrow \lfloor 2^{64} \cdot n \rfloor \geq 2^{62} + 1 \end{aligned} \quad (167)$$

1181 Then there are:

$$\begin{aligned} 1183 \quad \lfloor 2^{64} \cdot n \rfloor &\neq 2^{62} \&& \lfloor 2^{64} \cdot n \rfloor + 1 \neq 2^{62} \\ 1184 \quad &\Rightarrow \lfloor 2^{64} \cdot n_r \rfloor \neq 2^{62} \end{aligned} \quad (168)$$

1186 Therefore, the following condition can be used to quickly determine whether  $n \neq 0.25$ :

$$1188 \quad \text{double : } n \neq 0.25 \text{ if } \lfloor 2^{64} \cdot n_r \rfloor \neq 2^{62} \quad (169)$$

1189 In summary, for double, the following condition can be used to quickly determine whether  $n =$   
1190 0.25:

$$\begin{aligned} 1192 \quad \text{double : } n &= 0.25 \text{ if } \lfloor 2^{64} \cdot n_r \rfloor = 2^{62} \\ 1193 \quad \text{double : } n &\neq 0.25 \text{ if } \lfloor 2^{64} \cdot n_r \rfloor \neq 2^{62} \end{aligned} \quad (170)$$

1195 In the double range, introduce another faster way to calculate *one*:

$$1196 \quad \text{double : } \text{one} = \left\lfloor \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 + (n = 0.25) ? 0 : \left( 2^{-1} + \frac{6}{2^{64}} \right) \right\rfloor \quad (171)$$

1199 The proof of equation (171) is as follows:

1200 when  $n = 0.25$ ,  $\lfloor \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 \rfloor = \lfloor 10n \rfloor = 2$ ;

1201 when  $n \neq 0.25$ , equation (171) can be equivalent to the following:

$$1203 \quad \text{double : } \text{one} = \left\lfloor \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 + 2^{-1} + \frac{6}{2^{64}} \right\rfloor \quad (172)$$

1206 According to the  $10n - \lfloor 10n \rfloor$  range, *one* is represented as:

$$1207 \quad \text{double : } \text{one} = \begin{cases} \lfloor 10n \rfloor, & \text{if } 10n - \lfloor 10n \rfloor < 0.5 \\ 8, & \text{if } 10n - \lfloor 10n \rfloor = 0.5 \\ \lfloor 10n \rfloor + 1, & \text{if } 10n - \lfloor 10n \rfloor > 0.5 \end{cases} = \lfloor 20n + 1 \rfloor // 2 \quad (173)$$

1211 Therefore, when  $n \neq 0.25$ , we need to prove that the following equation holds:

$$1213 \quad \left\lfloor \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 + 2^{-1} + \frac{6}{2^{64}} \right\rfloor = \begin{cases} \lfloor 10n \rfloor, & \text{if } 10n - \lfloor 10n \rfloor < 0.5 \\ 8, & \text{if } 10n - \lfloor 10n \rfloor = 0.5 \\ \lfloor 10n \rfloor + 1, & \text{if } 10n - \lfloor 10n \rfloor > 0.5 \end{cases} = \lfloor 20n + 1 \rfloor // 2 \quad (174)$$

1216 From the range of  $n$ , there is:

$$1218 \quad \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \in (n_r - 2^{-64}, n_r] \quad (175)$$

1221 Because the following conditions exist:

$$\begin{aligned} 1222 \quad c \cdot 2^q \cdot 10^{-k-1} &= m + n \\ 1223 \quad c \cdot 2^q \cdot r \cdot 10^{-k-1} &= m + n_r \end{aligned} \quad (176)$$

1226 Therefore, the following relationship can be concluded:

$$\begin{aligned} 1227 \quad n_r - n &= (r - 1) \cdot c \cdot 2^q \cdot 10^{-k-1} \\ 1228 \quad n_r &= (r - 1) \cdot (m + n) + n \\ 1229 \quad \Rightarrow n &\leq n_r < 2^{-127} \cdot c + n \\ 1230 \quad \Rightarrow n &\leq n_r < 2^{-127} \cdot 2^{53} + n \\ 1231 \quad \Rightarrow n &\leq n_r < 2^{-74} + n \\ 1232 \quad \Rightarrow n &\leq n_r < 2^{-74} + n \\ 1233 \end{aligned} \tag{177}$$

1234 From equation (175) and (177), it can be concluded that:

$$\begin{aligned} 1235 \quad \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} &\in (n - 2^{-64}, n + 2^{-74}) \\ 1236 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 &\in (10n - 10 \cdot 2^{-64}, 10n + 10 \cdot 2^{-74}) \\ 1237 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (20n - 20 \cdot 2^{-64}, 20n + 20 \cdot 2^{-74}) \\ 1238 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1239 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1240 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1241 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1242 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1243 \quad \Rightarrow \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 &\in (\lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64}, \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74}) \\ 1244 \quad \text{Discuss the range of values of } x \text{ when the following conditions are met.} \\ 1245 \quad \lfloor \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 + 1 + x \rfloor / 2 = \lfloor 20n + 1 \rfloor / 2 = \text{one} \\ 1246 \quad \text{Therefore, the following conclusions can be drawn:} \\ 1247 \quad \lfloor 20n \rfloor + n_{20} - 20 \cdot 2^{-64} + 1 + x &\geq \lfloor 20n + 1 \rfloor \Rightarrow x \geq 20 \cdot 2^{-64} - n_{20} \\ 1248 \quad \lfloor 20n \rfloor + n_{20} + 20 \cdot 2^{-74} + 1 + x &< \lfloor 20n + 2 \rfloor \Rightarrow x < 1 - 20 \cdot 2^{-74} - n_{20} \\ 1249 \quad \text{Suppose } x = 12 \cdot 2^{-64}. \text{ Through the exhaustive method, all floating-point numbers that do not meet} \\ 1250 \quad \text{the following conditions can be obtained.} \\ 1251 \quad x = 12 \cdot 2^{-64} \geq 20 \cdot 2^{-64} - n_{20} \\ 1252 \quad \text{All floating-point numbers that do not meet condition (181) are as follows (in hexadecimal):} \\ 1253 \quad 0xd17c0747bd76fa1, \\ 1254 \quad 0xd27c0747bd76fa1, \\ 1255 \quad 0x4d73de005bd620df, \\ 1256 \quad 0x4d83de005bd620df, \\ 1257 \quad 0x4d93de005bd620df, \\ 1258 \quad \text{Through the exhaustive method, all floating-point numbers that do not meet the following conditions can be obtained.} \\ 1259 \quad x = 12 \cdot 2^{-64} < 1 - 20 \cdot 2^{-74} - n_{20} \\ 1260 \quad \text{All floating-point numbers that do not meet condition (183) are as follows (in hexadecimal):} \\ 1261 \quad 0x612491daad0ba280, \\ 1262 \quad 0x6159b651584e8b20, \\ 1263 \quad 0x619011f2d73116f4, \\ 1264 \quad 0x61c4166f8cf8cb1, \\ 1265 \quad 0x61d4166f8cf8cb1, \\ 1266 \quad \text{ACM Trans. Program. Lang. Syst., Vol. 37, No. 4, Article 111. Publication date: August 2026.}$$

1275 There are:

$$2\left(\frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 10 + 2^{-1} + \frac{6}{2^{64}}\right) = \frac{\lfloor 2^{64} \cdot n_r \rfloor}{2^{64}} \cdot 20 + 1 + x \quad (185)$$

1276 When the floating-point number is not within the above range (182) and (184), the condition (180)  
 1277 is satisfied. We have tested all floating-point numbers within the above-mentioned range (182) and  
 1278 (184), and the algorithm implementation code has output the correct result, that is, it satisfies the  
 1279 SW principle. The test process file is [test8.py](#).

1280 In summary, equation (174) and equation (171) holds. Therefore, equation (171) can be used to  
 1281 quickly calculate one.

### 1282 3.6 Irregular Number

1283 Due to the limited and small number of irregular floating-point numbers, there are a total of 2046  
 1284 double floating-point numbers and 254 float floating-point numbers. The correctness of the algo-  
 1285 rithm code in this paper can be proved by the exhaustive method. Therefore, it is not introduced  
 1286 in this article. For the specific implementation process, please refer to the source code.

1287 Table 1. All algorithms in the benchmark test.

1288 algorithm	float	double	description
Schubfach	Schubfach32	Schubfach64	author:Raffaello Giulietti, <a href="https://github.com/c4f7fcce9cb06515/Schubfach">https://github.com/c4f7fcce9cb06515/Schubfach</a> .
Schubfach_xjb	Schubfach32_xjb	Schubfach64_xjb	It is improved by Schubfach and has the same output result.
Ryu	Ryu32	Ryu64	author:Ulf Adams, <a href="https://github.com/ulfjack/ryu">https://github.com/ulfjack/ryu</a> .
Dragonbox	Dragonbox32	Dragonbox64	author:Junekey Jeon, <a href="https://github.com/jk-jeon/Dragonbox">https://github.com/jk-jeon/Dragonbox</a> .
fmt[10]	fmt32	fmt64	author:Victor Zverovich, <a href="https://github.com/fmtlib/fmt">https://github.com/fmtlib/fmt</a> version:12.1.0
yy_double	-	yy_double	author:GuoYaoYuan,link: <a href="#">yy_double</a> .
yy_json[11]	yy_json32	yy_json64	author:Guo YaoYuan, <a href="https://github.com/ibireme/yyjson">https://github.com/ibireme/yyjson</a> version:0.12.0
teju_jagua[12]	teju32	teju64	author:Cassio Neri, <a href="https://github.com/cassioneri/teju_jagua">https://github.com/cassioneri/teju_jagua</a> .
xjb	xjb32	xjb64	this paper, <a href="https://github.com/xjb714/xjb">https://github.com/xjb714/xjb</a> .
zmij[13]	zmij32	zmij64	author:Victor Zverovich, <a href="https://github.com/vitaut/zmij">https://github.com/vitaut/zmij</a> .
jnum[14]	jnum32	jnum64	author:Jing Leng, <a href="https://github.com/lengjingzju/json/jnum.c">https://github.com/lengjingzju/json/jnum.c</a> .

## 1309 4 BENCHMARK RESULT

1310 In fact, this article only discusses the binary to decimal part and does not discuss the decimal  
 1311 to string part. In the decimal to string section, the neon instruction set is adopted for the arm64  
 1312 architecture, and SSE2/SSE4.1/AVX512IFMA instruction set is used for the x86-64 architecture to  
 1313 accelerate the conversion process. Please refer to the source code design. In the performance test  
 1314 comparison, we compared the time spent by the following several different algorithms converting  
 1315 floating-point numbers to decimal results and string, as shown in Table (1). Test process: Generate  
 1316  $2^{24}$  random numbers without 0, NaN, and Inf, measure the total time spent converting all floating-  
 1317 point numbers to decimal results, and obtain the average time for converting a single floating-  
 1318 point number to decimal and string. The compiler used for AMD R7-7840H is icpx 2025.0.4, and  
 1319 the compiler used for Apple M1 is apple clang 17.0.0. The compilation option for all compilers is  
 1320 "-O3 -march=native". We conducted benchmark tests on two processors, and the test results are  
 1321 shown in Fig (1a) to Fig (1d).

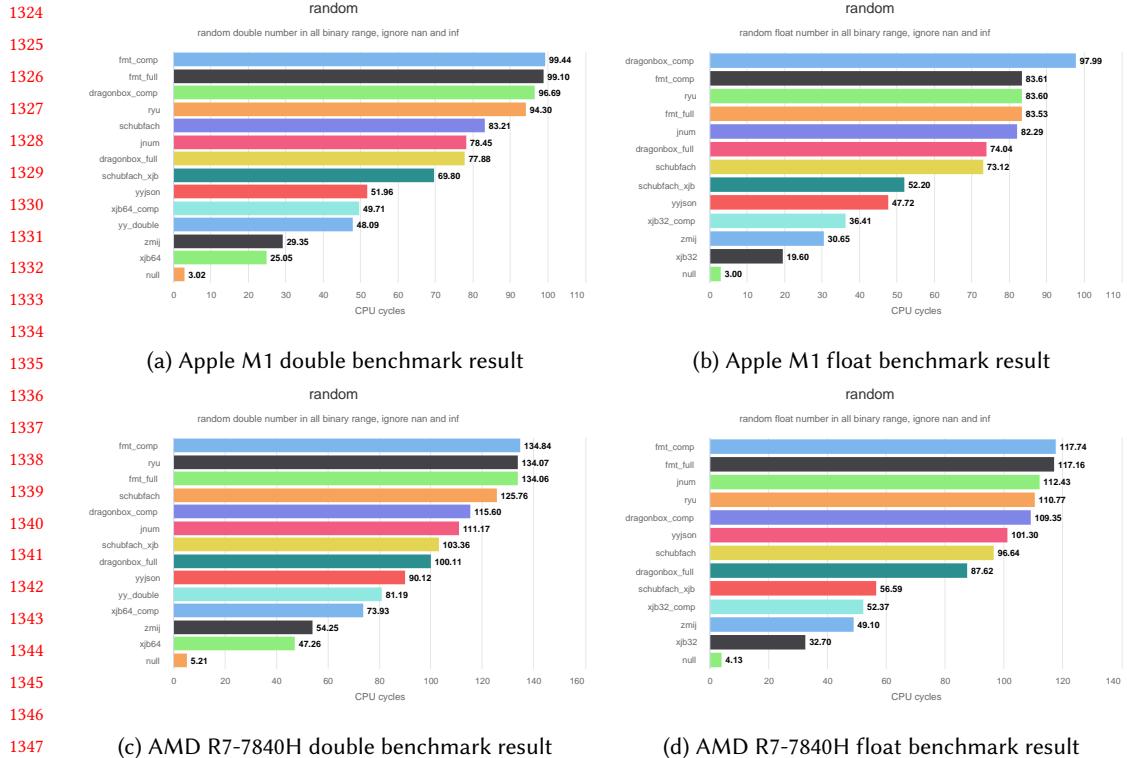


Fig. 1. Benchmark results

Special note: The algorithm of teju\_jagua only supports float/double to decimal, because its author did not implement the source code of decimal to string. yy\_double only supports double. Dragonbox\_comp, fmt\_comp and xjb\_comp represent the versions of the compressed constant lookup table. Dragonbox\_full and fmt\_full represent uncompressed constant lookup table.

From the benchmark results, it can be seen that the performance of the algorithm in this paper is better than other algorithms in most cases.

## 5 CONCLUSIONS AND FUTURE WORK

This paper proposes a new floating-point number to string conversion algorithm. The algorithm improves the calculation process of Schubfach algorithms, reduces the number of multiplication operations, and optimizes some calculation steps. The algorithm has been implemented in C/C++ language and passed exhaustive tests. The benchmark results show that the performance of the algorithm is better than most existing algorithms in most cases. Future work includes further optimization of the algorithm to improve performance, especially for parallel computing on x86-64 and arm64 architecture, and compatibility with the msvc compiler.

## REFERENCES

- [1] Guy L. Steele and Jon L. White. 1990. How to print floating-point numbers accurately. *SIGPLAN Not.* 25, 6 (June 1990), 112–126. [doi:10.1145/93548.93559](https://doi.org/10.1145/93548.93559)
- [2] Florian Loitsch. 2010. Printing floating-point numbers quickly and accurately with integers. *SIGPLAN Not.* 45, 6 (June 2010), 233–243. [doi:10.1145/1809028.1806623](https://doi.org/10.1145/1809028.1806623)

- 1373 [3] Marc Andryesco, Ranjit Jhala, and Sorin Lerner. 2016. Printing floating-point numbers: a faster, always correct method.  
1374 *SIGPLAN Not.* 51, 1 (Jan. 2016), 555–567. doi:[10.1145/2914770.2837654](https://doi.org/10.1145/2914770.2837654)
- 1375 [4] Ulf Adams. 2018. Ryū: fast float-to-string conversion. *SIGPLAN Not.* 53, 4 (June 2018), 270–282. doi:[10.1145/3296979.3192369](https://doi.org/10.1145/3296979.3192369)
- 1376 [5] Ulf Adams. 2019. Ryū revisited: printf floating point conversion. *Proc. ACM Program. Lang.* 3, OOPSLA, Article 169  
1377 (Oct. 2019), 23 pages. doi:[10.1145/3360595](https://doi.org/10.1145/3360595)
- 1378 [6] R. Giulietti. 2020. The Schubfach Way to Render Doubles. (Sept. 2020). [https://drive.google.com/file/d/1KLtG\\_LalbK9ETXI290zqCvxBW94dj058/view](https://drive.google.com/file/d/1KLtG_LalbK9ETXI290zqCvxBW94dj058/view)
- 1379 [7] Junekey Jeon. 2020. Grisu-Exact: A Fast and Exact Floating-Point Printing Algorithm. (Sept. 2020). [https://github.com/jk-jeon/Grisu-Exact/blob/master/other\\_files/Grisu-Exact.pdf](https://github.com/jk-jeon/Grisu-Exact/blob/master/other_files/Grisu-Exact.pdf)
- 1380 [8] Junekey Jeon. 2024. Dragonbox: A New Floating-Point Binary-to-Decimal Conversion Algorithm. (July 2024). [https://github.com/jk-jeon/dragonbox/blob/master/other\\_files/Dragonbox.pdf](https://github.com/jk-jeon/dragonbox/blob/master/other_files/Dragonbox.pdf)
- 1381 [9] Guo YaoYuan. 2024. (Nov. 2024). [https://github.com/ibireme/c\\_numconv\\_benchmark/blob/master/vendor/yy\\_double/yy\\_double.c](https://github.com/ibireme/c_numconv_benchmark/blob/master/vendor/yy_double/yy_double.c)
- 1382 [10] Victor Zverovich. 2025. (Oct. 2025). <https://github.com/fmtlib/fmt>
- 1383 [11] Guo YaoYuan. 2025. (Aug. 2025). <https://github.com/ibireme/yyjson>
- 1384 [12] Cassio Neri. 2025. (Nov. 2025). [https://github.com/cassioneri/teju\\_jagua](https://github.com/cassioneri/teju_jagua)
- 1385 [13] Victor Zverovich. 2026. (Jan. 2026). <https://github.com/vitaut/zmij>
- 1386 [14] Jing Leng. 2025. (Nov. 2025). <https://github.com/lengjingzju/json/jnum.c>

1389 Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421