00.01 basics: polynomials & floating-point

00 numerical methods

numerical methods, as distinguished from other branches of mathematics and from computer science,

1. work with arbitrary real numbers (including rational **approximations** of irrational number) and
2. consider **cost** and
3. consider **accuracy**.[1]

this class will provide another way to express, to extend your math.

numerical methods are the algorithms; **numerical analysis** is the study of their properties -- ie, accuracy, stability, convergence, efficiency, usw.

01 polynomials

*The most fundamental operations of arithmetic are **addition** and **multiplication**. These are also the operations needed to evaluate a polynomial $p(x)$ at a particular value x. It is no coincidence that polynomials are the basic building blocks for many computational techniques we will construct.*[2]

i) evaluation

eg: $p(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$.

with computational considerations:

1. **approximate** $p(x)$ at x while
2. minimizing **operations** and
3. maximizing **accuracy**.

- method 1, step individually:

$p(x) = a_4 \times x \times x \times x \times x + a_3 \times x \times x \times x + a_2 \times x \times x + a_1 \times x + a_0 \mapsto 14$ operations.

- method 2, cache and reuse:

$x_2 = x \times x, x_3 = x_2 \times x, x_4 = x_3 \times x \mapsto 3$ operations;
$p_4 = a_4 \times x_4, p_3 = a_x \times x_3, p_2 = a_2 \times x_2, p_1 = a_1 \times x_1 \mapsto 4$ operations;
$p(x) = p_4 + p_3 + p_2 + p_1 + a_0 \mapsto 4$ ops $\mapsto 11$ operations total.

- method 3, nested multiplication (horners method):

$p(x) = (((a_4 \times x + a_3) \times x + a_2) \times x + a_1) \times x + a_0 \mapsto 8$ operations.

02 binary notation

binary notation: $\quad \ldots b_2 b_1 b_0 . b_{-1} b_{-2} \ldots.$

i) conversion to decimal

$$\Rightarrow \ldots b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} \ldots$$

eg, 111.11_2.

$$\text{integer:} \quad 1 \times 2^2 + 1 \times 2^1 = 4 + 2 + 1 = 7$$

$$\text{fractional:} \quad 1 \times 2^{-1} + 1 \times 2^{-2} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$$

$$\Downarrow$$

$$111.11_2 = 7_{10} + (\tfrac{3}{4})_{10} = 7.75_{10}.$$

ii) decimal conversion to binary

eg, 111.25_{10}.

$$
\begin{aligned}
\text{integer:} \quad & \tfrac{111}{2} = 55\,R\,1 \\
& \rightarrow \tfrac{55}{2} = 27\,R\,1 \\
& \rightarrow \tfrac{27}{2} = 13\,R\,1 \\
& \rightarrow \tfrac{13}{2} = 6\,R\,1 \\
& \rightarrow \tfrac{6}{2} = 3\,R\,0 \\
& \rightarrow \tfrac{3}{2} = 1\,R\,1 \\
& \rightarrow \tfrac{1}{2} = 0\,R\,1 \\
& \rightarrow 11011111, \quad \text{remainders in reverse order} \\
\\
\text{fractional:} \quad & 0.25 \times 2 = 0.50 + 0 \\
& \rightarrow 0.50 \times 2 = 0.00 + 1 \\
& \rightarrow 0.01, \quad \text{integers in order from left to right} \\
\\
& \Downarrow \\
\\
111.25_{10} &= 1101111_2 + 0.01_2 = 11011111.01_2.
\end{aligned}
$$

03 polynomials in the machine

i) digital representation

$$
\begin{aligned}
x &= [d_{N-1}, \ldots, d_1, d_0] \quad \text{digital vector} \\
&= d_{N-1} \times b^{N-1} + \cdots + d_1 \times b^1 + d_0 \times b^0 \quad \text{with } \textbf{precision}\, N \text{ and } \textbf{base}\, b.
\end{aligned}
$$

eg,

- base 10: $500_{10} = [5, 0, 0]; \quad [5] = 5_{10}.$
- base 02: $[1, 0, 1] = 101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5_{10}.$

ii) fixed/positional representation

using previous example,

- base 02: $101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

where RHS is **fixed representation** and LH subscript is the base or **radix** r. additionally, precision $N \geq 1, r \geq 2$ such that

$x = \sum^N d_k r^k$ has r^N **permutations** and can also be written as

$r^N = (r-1)(\mathbf{r^{N-1}}) + (r_{N-1}) = [r-1]_{N-1}[r]_{N-2} \ldots [r]_1 [r]_0 [r]_{-1} [r]_{-2} \ldots [r]_{N-2} [r]_{N-1}$
where subscripts denote position wrt exponent.

eg, describe set where $N = 3, r = 2$.

$$\text{permutations:} \quad r^N = 2^3 \Rightarrow \{000, 001, 010, 011, 100, 101, 110, 111\};$$

$$\text{magnitude:} \quad \sum^{N-1} d_k r^k \leq \sum^{N-1} (r-1) r^k = \mathbf{r^N - 1} \Rightarrow \quad \text{range}^{[*]} \quad [0, \mathbf{r^N - 1}].$$

[*] note: "range of magnitude" of x is also "range" of x bc representation of x does not allow for sign.

iii) sign

sign extends range.

- method 1: use position d_{N-1} for sign.

$$x = [\pm][d_{N-1}, \ldots, d_1, d_0] \quad \text{and}$$

$$\text{permutations:} \quad r^{N-1} \times 2;$$

$$\text{range:} \quad [-r^{N-1} + 1, 0), [0, +\mathbf{r^{N-1} - 1}].$$

- method 2: use bias to obtain sign.

ie, all positions used for magnitude and bias is an operation.

$$x_{\min} = -B, x_{\max} = r^N - B$$

$$\text{range:} \quad [x_{\min}, x_{\max}] = [1 - r^{N-1}, r^{N-1}(r-1)]$$

with **standard bias** $B = \mathbf{r^{N-1} - 1}$.

eg, describe set where $N = 3, r = 2$ with standard bias.

$$B = r^{N-1} - 1 = (2)^{(3-1)} - 1 = 4 - 1 = 3 \quad \text{and}$$

$$\text{range:} \quad [000, 111]_2 \mapsto [0, 7]_{10} - B = [-3, +4]_{10}.$$

04 floating-point

$x = M.b^E$, where **mantissa** M is an integer represented by sign, magnitude, radix and **precision** m; **exponent** E is an integer represented by bias and same radix. also, M is **normalized** as $1.F$, where "$1.$" is implicit and **fractional** F is

$$F = \sum^{m-2} d_k r^k, r \geq 2 \Rightarrow x = \pm 1.F \times b^E.$$

ie, same r for M, E; m includes sign; $m_E = N - m$; $B = r^{N-1} - 1$ with bias power $N - 1 = m_E - 1$. note: b is the base of the exponent and not the base of the exponents power.

$$x = \pm 1.F \times 2^E = [s] \overbrace{\underbrace{\overbrace{[\ldots][e_1][e_0]}^{m_E = N - m}}_{\ldots + e_1 \times r^1 + e_0 \times r^0} \; 1. \; \underbrace{\overbrace{[f_1][f_2][\ldots]}^{m_F = m - sign}}_{f_1 \times r^{-1} + f_2 \times r^{-2} + \ldots}}^{\text{positions allocated}},$$

example 01

$\mathbb{FP}(N=5, m=3, r=3, b=2)$ with standard bias.[3]

$$x = \pm 1.F \times 2^{E} = [s]\ \underbrace{\overbrace{[e_1][e_0]}^{m_E=5-3=2}\ 1.\ \overbrace{[f_1][f_2]}^{m_F=3-1=2}}_{\text{positions allocated}},$$

where

$$s \in \{0,1\}$$

$$f_j \in \{0,1,2\}_3$$

$$\Downarrow$$

$$F_{\text{magnitude}} = [0.00, 0.22]_3 \quad \text{and}$$

$$e_i \in \{0,1,2\}_3$$

$$B = r^{N-1} - 1 \mapsto r^{m_E - 1} - 1 = 3^{2-1} - 1 = (3-1)_{10} = 2_{10}$$

$$\Downarrow$$

$$E_{\text{range}} = [00, 22]_3 - B = [0, 8]_{10} - 2_{10} = [-2, 6]_{10}.$$

eg,

$$x = [0, 1, 1, 2, 0]_{\mathbb{FP}(5,3,3,2)}$$

$$= (-1)^0 \times 1.20_3 \times 2^E \quad \text{where } E = (11_3 - B) = (4-2)_{10} = 2_{10}$$

$$= +(1. + 2 \times 3^{-1})_{10} \times 2^2 = +(\tfrac{5}{3}) \times 4 = +\tfrac{20}{3} = +6.\bar{6}.$$

example 02

$\mathbb{FP}(N=6, m=4, r=3, b=2)$ with standard bias.[4]

$$x = \pm 1.F \times 2^E = [s]\ \underbrace{\overbrace{[e_1][e_0]}^{m_E=6-4=2}\ 1.\overbrace{[f_1][f_2][f_3]}^{m_F=4-1=3}}_{\text{positions allocated}},$$

where

$$s \in \{0,1\},\ f_j \in \{0,1,2\}_3$$

$$\Rightarrow F_{\text{magnitude}} = [0.000, 0.222]_3 \quad \text{and}$$

$$e_i \in \{0,1,2\}_3$$

$$B = r^{N-1} - 1 \mapsto r^{m_E-1} - 1 = 3^{2-1} - 1 = (3-1)_{10} = 2_{10}$$

$$\Rightarrow E_{\text{range}} = [00, 22]_3 - B = [0, 8]_{10} - 2_{10} = [-2, 6]_{10}.$$

ie,

$$|x_{\min}| = [0, 0, 0, 0, 0, 0]_{\mathbb{FP}(6,4,3,2)}$$

$$= (-1)^0 \times 1.000_3 \times 2^E, \quad E = (00_3 - B) = (0-2)_{10} = -2_{10}$$

$$= +1.0_{10} \times 2^{-2} = +\tfrac{1}{4}.$$

$$|x_{\max}| = [0, 2, 2, 2, 2, 2]_{\mathbb{FP}(6,4,3,2)}$$

$$= (-1)^0 \times 1.222_3 \times 2^E, \quad E = (22_3 - B) = (8-2)_{10} = 6_{10}$$

$$= +[1. + (2 \times 3^{-1} + 2 \times 3^{-2} + 2 \times 3^{-3})_{10}] \times 2^6 = +(1 + \tfrac{26}{27}) \times 64 \approx +125.\overline{629}.$$

i) denormalized vs normalized

a base-2 floating-point number will always start with "1", so its inclusion is implied. explicitly, 1×2^0 is a given so the position it might have used is given over to the fractional part of the mantissa. that is the normalized mantissa.

however, if the biased exponent is zero, the mantissa is **denormalized**. ie, there is no implicit "1". *(note: this is a feature of the standard, IEEE-754 and not necessarily a feature of other FPS.)*

$$\begin{aligned}\text{eg, } B_{\text{IEEE 754}} &= 126 \Rightarrow [0][00000000]0.[00010\ldots0] \\ &= +(1 \times 2^{-4}) \times (2^{0-126}) = +2^{-130}.\end{aligned}$$ [5]

ii) IEEE 754

the standard is IEEE 754-2019, $\mathbb{FP}(N-1, m, r, b) = \mathbb{FP}(64, 53, 2, 2)$, where 32-bit is single precision and 64-bit is double-precision.

iii) hexadecimal vs binary

IEEE 754 stores floating-point numbers using binary format; however, hexadecimal (base 16) representation of those bits is considered more human friendly.

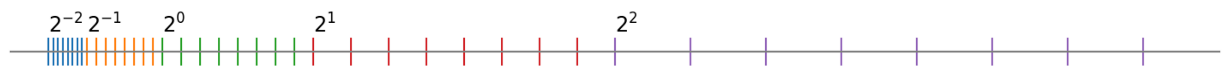
consider the approximation of π:

```
π = 3.14159265358979

IEEE 754: 01000000001001000001111110111010111
Sign Bit: 0
Exponent: 10000000 (128 in decimal, after subtracting the bias of 127)
Mantissa: 00100100000111110111010111

Hex     : 0x40490FDB
Sign Bit: 0
Hex Flag: x
Exponent: 40
Mantissa: 490FDB
```

iv) observations

- gaps between adjacent numbers scale with magnitude of number represented. (ie, consider negative exponents vs positive exponents.)

```
if __name__ == "__main__":

```



- machine epsilon, ϵ_{mach}, is the gap between 1 and the next FPN.
- unit roundoff, $\mu_{\text{mach}} = \frac{1}{2}\epsilon_{\text{mach}}$.
- for all x there exists a floating-point x' such that $|x - x'| \leq \mu_{\text{mach}} \times |x|$.
- when M normalized, zero represented by $\epsilon = \epsilon_{\text{min}} - 1$.
- $\pm\infty$ returned when and operation overflows.
- $\frac{x}{\pm\infty}$ returns 0 and $\frac{x}{0}$ returns $\pm\infty$.
- "not a number" (NaN) is returned if no well-defined finite or infinite result.
- [und so weiter](#).

resources

- horners method @wiki
- telescoping sum @wiki
- floating-point @wiki @youtube #1 #2-pt1 #2-pt2
- unit in last place (ulp) @wiki
- machine epsilon @wiki
- IEEE 754-2019

references

1. johnson, sg. *18.335, introduction to numerical methods*, mit.ocw, spring 2015.
2. sauer, tim. *numerical analysis, 2nd edition*, pearson education, 2012, p1.
3. martinez, vincent. *math 685*, hunter, spring 2023.
4. *ibid*.
5. nerdfirst. *denormal numbers*, 0612 tv, 2020.