

# **WS 23/24 Numerics Notes**

Igor Dimitrov

2023-10-30

# Table of contents

<b>Preface</b>	<b>3</b>
<b>1 Floating Point Numbers</b>	<b>4</b>
1.1 ANSI/IEEE 64 Bit . . . . .	4

# Preface

Notes for the lecture “[WS 23/24 Numerics 0](#)” at Uni Heidelberg.

# 1 Floating Point Numbers

## 1.1 ANSI/IEEE 64 Bit

Let  $\tilde{a}$  be a 64 bit IEEE floating point number.  $\tilde{a}$  is represented as

S E ... E M ... M

Where S is the sign bit, 11 E's are the exponent bits and 52 M's are mantissa bits. Interpretation (Case analysis on value of  $E$ ):

1. S | 0 ... 0 | M:
  1.  $M = 0 \Rightarrow \tilde{a} = (-1)^S 0$
  2.  $M \neq 0 \Rightarrow \tilde{a} = (-1)^S \times 2^{-1022} \times 0.M$  (**subnormal range**)
2.  $1 \leq E \leq 2046 \Rightarrow \tilde{a} = (-1)^S \times 2^{E-1023} \times 1.M$  (**normal range**)
3. S | 1 ... 1 | M:
  1.  $M = 0 \Rightarrow \tilde{a} = (-1)^S \text{inf}$
  2.  $M \neq 0 \Rightarrow \tilde{a} = \text{NaN(Not a Number)}$  (**exceptions**)

See Figure 1.1 for a visual summary.

**Examples:**

- **realmin**: smallest normalized positive machine number in FP64 =  $\llbracket 0|0 \dots 01|0 \dots 0 \rrbracket_{FP64} = 2^{1-1023} \times 1.0 = 2^{-1022}$
- **realmax**: greatest normalized machine number in FP64 =  $\llbracket 0|1 \dots 10|1 \dots 1 \rrbracket_{FP64} \approx 1.7977\text{E}308$
- $1 = 2^0 \times 1.0 = 2^{1023-1023} \times 1.0 = \llbracket 0|01 \dots 1|0 \dots 0 \rrbracket_{FP64}$

FP64 stands for IEEE **F**loating **P**oint **64** bit number representation. Whereas  $[\cdot]_{FP64}$  is the **FP64** evaluation/interpration of the machine number

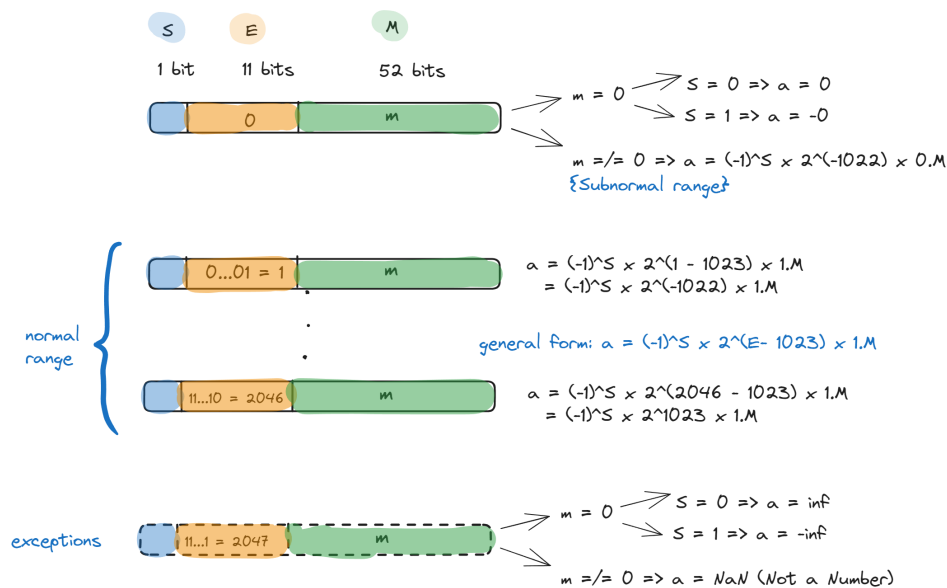


Figure 1.1: floating-point

- **eps**: spacing in the interval  $(1, 2)$ . Note that the spacing is constant for each such interval, but grows as we go further down the number line. That is, the spacing in  $(1000, 1001)$  is constant, but larger.
- number right after 1:  $[0|01\dots1|0\dots1]_{\text{FP64}}$ . Then the spacing, i.e. **eps** from above is  $2^{-52}$