H1: Numerical limitations

1.1 basic concepts

1.1.1 absolute error and relative error

absolute error	= approximate value - true value
relative error	= absolute error / true value > if an approximate value has a relative error of 10^{-p} then its decimal representation has p correct significant digits

1.1.2 precision and accuracy

precision	= number of digits with which a number is expressed
accuracy	= number of correct significant digits in an approximation

1.1.3 truncation and rounding error

truncation error	= difference between the true result and the result given by an algorithm > due to approximations
rounding error	= difference between the result produced by a given algorithm using exact arithmetic and the same algorithm rounded arithmetic

1.2 floating-point number systems

floating-point number system

4 integers:

symbol	name
β	Base or radix
p	Precision
$[L, U]$	Exponent range

> the floating-point number thus has the form:

$$\pm\left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_{p-1}}{\beta^{p-1}}\right)\beta^E$$ where d_i and E are integers such that $0 \le d_i \le \beta - 1$ and $L \le E \le U$.

Normalized floating-point number syst.

= fl.-p number syst. for which $d_0 = 1$

> advantageous because:

- each number has a unique representation
- no digits are wasted by leading zero's
  > maximized precision
- in a binary system, β=2, the leading bit is always 1
  > doesn't need to be stored
  > 1 bit extra of precision

Single Precision IEEE (SP)

System	β	p	L	U
IEEE SP	2	24	-126	127

> stored in 4 bytes = 32bits:
- 1 sign bit: 0 = + and 1 = -
- 8 bits for the exponent: [-126, 127]
- 23 bits for the mantissa

<u>calculation of the exponent</u>

ex: 01011001

> take -127 and add 2^n for every 1 in the 8 bits:

$$-127 + 1 + 8 + 16 + 64 = -38$$

<u>calculating the mantissa</u>

Formula: $$\left(1 + \frac{d_1}{2} + \frac{d_2}{4} + \ldots \frac{d_{23}}{2^{23}}\right)$$

in which d_1 = first bit, …., d_{23} = 23th bit

For example: 0 01111110 10000000000000000000000

$$= \left(1 + \frac{1}{2} + \frac{0}{4} + \ldots \frac{0}{2^{23}}\right)2^{(-127+126)} = 1.5 \cdot 2^{-1} = 0.75$$

<table>
<tr><td>Double precision IEEE
(DP)</td><td><table><tr><th>System</th><th>β</th><th>p</th><th>L</th><th>U</th></tr><tr><td>IEEE SP</td><td>2</td><td>24</td><td>-126</td><td>127</td></tr><tr><td>IEEE DP</td><td>2</td><td>53</td><td>-1022</td><td>1023</td></tr></table></td></tr>
<tr><td colspan="2">1.3 Properties of floating-point number systems</td></tr>
<tr><td>Total numbers in a fl.-point number syst.</td><td>The amount of numbers you can represent with a certain system is given by:
$$2(\beta-1)\beta^{p-1}(U-L+1)+1$$
2 choices of sign
$(\beta-1)$ choices for the leading digit of the mantissa $(=d_0)$
β^{p-1} because there are β choices for each of the remaining $p-1$ digits of the mantissa
$(U-L+1)$ possible values for the exponent ($+1$ because the boundaries of [L, U] are being counted)
$+1$ because the number could be zero</td></tr>
<tr><td>underflow level

overflow level</td><td>= smallest positive normalized number
> all bits in the mantissa and all but the last bit are 0, the number equals β^L

= largest number: $\beta^{U+1}(1-\beta^{-p})$ (derivation: see git)</td></tr>
<tr><td>machine numbers</td><td>= real number that are exactly representable in a fl-p system

> if a certain number is not representable its rounded to a nearby number
> has a certain rounding error</td></tr>
<tr><td>machine precision</td><td>= the accuracy of a fl-p system

> if a system rounds to its nearest: the machine precision equals $\epsilon_{\text{mach}} = \frac{1}{2}\beta^{1-p}$</td></tr>
<tr><td>inf

NaN</td><td>= infinity, the result of dividing zero's

= not a number, the result from an undefined operations, such as 0/0</td></tr>
<tr><td colspan="2">1.5 good practices for computer arithmetic</td></tr>
<tr><td>good practices:</td><td>- avoid subtracting two almost identical numbers
- avoid adding small and large numbers
- perform a sequence of additions ordered from smallest to largest number</td></tr>
</table>