IEEE Floating Point Format

1. **Definition:** A *floating point* number consists of three parts: (i) The sign (+/-), (ii) The Mantissa (string of significant digits [in base 2, bits]); (iii) The exponent.

   A *normalized* floating point number is a number of the form:

   $$\pm 1.bbbbbb...b \times 2^p$$

   where the number of b's will vary depending on the precision, and p is the exponent (from the floating pt number definition).

2. Levels of precision for floating point numbers (the integers below represent how many bits will be used for the representation):

	sign	exponent	mantissa
single	1	8	23
double	1	11	52
long double	1	15	64

   Important Note: Unless otherwise stated, everything we do with floating point numbers will be done in double precision.

3. **Definition:** Machine epsilon, ϵ_{mach} is the distance between the number 1 and the next smallest number greater than 1. In double precision (which is what Matlab will use),

   $$\epsilon_{\text{mach}} = 2^{-52} \approx 2.2 \times 10^{-16}$$

4. IEEE rounding (to nearest neighbor): The bit of interest is in the 53d position (recall we're using double precision).

   - If the bit is 0, round down (truncate after 52d bit)
   - If the bit is 1, check the bits to the right:
     - If they are not all zero (the usual choice), round up.
     - If they are all zero, round up iff bit 52 is 1.

5. **Definition:** $fl(x)$ is the floating point representation of x.

6. Example: Compute $fl(1/3)$ and find the error in this approximation.

   First, we see that

   $$\left(\frac{1}{3}\right)_{10} = \left(0.\overline{01}\right)_2$$

To get this base 2 number into (normalized) floating point form,

$$1.010101\ldots10\underbrace{1}_{52\text{d bit}}010101\ldots\times 2^{-2}$$

so that the error (or tail) is $2^{-52}\cdot 0.\overline{01}\times 2^{-2}=\frac{1}{3}\cdot 2^{-54}$.

That leaves:

$$fl(1/3)=1.010101\ldots101\times 2^{-2}$$

With an error, in terms of ϵ_{mach}: $\frac{1}{3}\cdot 2^{-54}=\frac{1}{12}\epsilon_{\text{mach}}$