FP Arithmetic $x = a * e^b$

$x_0 \pm x_1 = (a_0 \pm (a_1 * e^{b_1 - b_0})) * e^{b_0}$

(may require several shifts to normalise)

Representation Single: p = 24, b = 127
Double: p = 53, b = 1023
Hidden bit for leading 1
Exponent in "excess" format: so fp cmp = int cmp for +VE

Exponent Hack 0...0: m = 0 ? zero : denorm
1…1: m = 0 ? infinity : NaN

Basic Ops Do perfect mathematical operations on assumed-precise inputs and round this to the nearest representable IEEE number. In the case of a tie choose the number with an even LSB

IEEE Rounding Unbiased/towards 0/ towards +∞/towards -∞

Error $\varepsilon = |a - b|$, $\eta = \frac{|a-b|}{|a|}$

+, -: add absolute
*, /: add relative
Rounding err. (finite rep.)
Truncation err. (finite comp.)

Gradual Loss Of Significance Many sf of precision but with reducing sf of accuracy

Machine Eps Difference between 1.0 and the smallest greater representable number (allows for denorms at 0.0)

Ill-Conditioned Outputs are excessively dependent on small variations in the inputs

Quadratic Example Watch for –b+b in $x = \frac{-b+\sqrt{b^2-4ac}}{2a}$, rearrange for small root only: $x = \frac{-2c}{b+\sqrt{b^2-4ac}}$

Differentiation Example Use $f'(x) = \frac{f(x+h)-f(x)}{h}$, but get truncation for large h and rounding for small h: choose h where errors equal
Taylor expansion of formula:

$f(x+h) \approx f(x) + hf'(x) + \frac{h^2 f''(x)}{2}$

Trunc. error: $\frac{hf''(x)}{2}$ (determine this by substitution into f′(x))
Round. error: $\frac{macheps}{h}$ (subs. Taylor into f′(x) and call terms above h "± macheps")

Taylor Series Example Watch for truncation and cancelling intermediate terms

Use range reduction if possible (e.g. with sin), this can allow you to fix iterations and unroll the summing loop

In Practice IEEE is unbiased so for many programs errors act more like random errors than accum. ones: *often* find k-ops program has error of $macheps.\sqrt{k}$ not $macheps.\frac{k}{2}$

Interval Arithmetic Use two real values, one guaranteed higher than real and another lower than real
Can naturally cope with input value uncertainty
Hack this up with IEEE round.
Computed bounds may not reflect accuracy (e.g. with convergent algorithms)
A range can span positive and negative infinity due to division w/ range spanning 0

Arbitrary Precision For any fixed precision has the same problems as IEEE, but for adaptive precision can do better: if result cancels repeat computation with more accuracy. Calculating 0 is a problem (undecidable!)

Exact Real Aritmetic Better techniques than digit streams (linear maps, fraction expansions…)