Default Rules for Rounding
Fixed Precision Floating Point Arithmetic
Ignoring Over/underflow.

[illegible] following rules would have to be amended only slightly to [illegible] over/underflow, which is a nearly independent and much [illegible] complicated topic. For simplicity here we consider the "representable numbers" to be an infinite discrete subset of the continuum of real numbers.

[illegible] representable numbers must include 0, 1 and, if x then $-x$ too.

[illegible] representable number must be represented uniquely by a symbol string that represents nothing else.

[illegible] arithmetic operation* which, when executed without roundoff error would produce a representable number, must actually be executed without error.

[illegible] discard information unnecessarily.

[illegible] arithmetic operation which [illegible] roundoff error must result in a representable number [illegible] would have been produced in the absence of roundoff [illegible]

The preceding rule is ambiguous when two representable numbers are nearest the unrounded result. This ambiguity [illegible] be resolved in a systematic way which preserves sign symmetry (e.g. $x-y = -(y-x)$) and is "unbiased" in [illegible] sense that "drift" cannot occur; e.g. the sequence [illegible] defined for arbitrary x_0 and y by $x_{n+1} := (x_n+y)-y$ [illegible]
$= x_2 = x_3 = \dots$

* Arithmetic operations include +, −, ×, /, $|\cdot|$, and conversion; [illegible] be extended to include √ and other FORTRAN functions [illegible] above were slightly relaxed.

W. Kahan
Univ. of Calif. @ Berkeley
Sept. 25, 1989