Rounding Modes for Interval Arithmetic in WASM

Why Interval Arithmetic (IA)?

- IA is the foundation of analog security in engineering simulations.
- Exact results from simulation are desirable but numerically infeasible.
- But we can guarantee that a given performance metric is at least X and at most Y, hence [X, Y].

Practical Use Cases

- Guaranteeing minimum fuel economy of a new engine design.
- Limiting the maximum angle of descent of an airplane.
- Computing the maximum load that a bridge can support under the fastest anticipated winds.
- Ensuring that the battery in a medical device will sustain a particular current for the maximum required duration.

Why Rounding Modes (RMs)?

- We can implement IA with existing float instructions.
- But the output intervals may be truncated at the ends, so the exact answer might be out of range!
- We need to round the lower limit down and the upper limit up. This is nonperformant without hardware acceleration.
- Even individual IA operations are too architecturally complex for WASM, so we need to invent float instructions with explicit RM instead.

Existing Library Support

- C/C++: Boost and eXtended Scientific Computing (XSC)
- Julia: IntervalArithmetic
- Python: PyInterval & Pyinter
- Rust: Inari

Existing Interval Applications

- Computational Geometry Algorithms Library (CGAL) computes convex hulls of manifolds, which accelerates 3D rendering and facilitates packaging guarantees (Boost).
- Derivative Code by Overloading (DCO) computes multivariate derivatives (Boost).
- Gappa can determine error bounds on numerical feedbacks in chaotic dynamics (Boost).
- Interval Package for Coq is used to prove various inequalities using intervals (Boost).

- IntervalRootFinding optimizes polynomial root searches (Julia).
- IntervalOptimisation finds global extrema, providing guarantees against the existence of spikes between floats (Julia).
- IntervalConstraintProgramming performs constrained optimization with intervals (Julia).
- TaylorModels computes Taylor series with intervals (Julia).
- Modeling systems starting from uncertain data via Dempster–Shafer theory (XSC).

Bifurcated Optimization

- The current proposal is for a minimum viable reference implementation which does not prevent anticipated future optimizations.
- Those optimizations potentially include the use of hardware instructions with explicit RM, as introduced in Intel AVX10, as well as optimized sign extractors which were originally planned for inclusion in this proposal.