

SPERR WAVELET-BASED COMPRESSOR

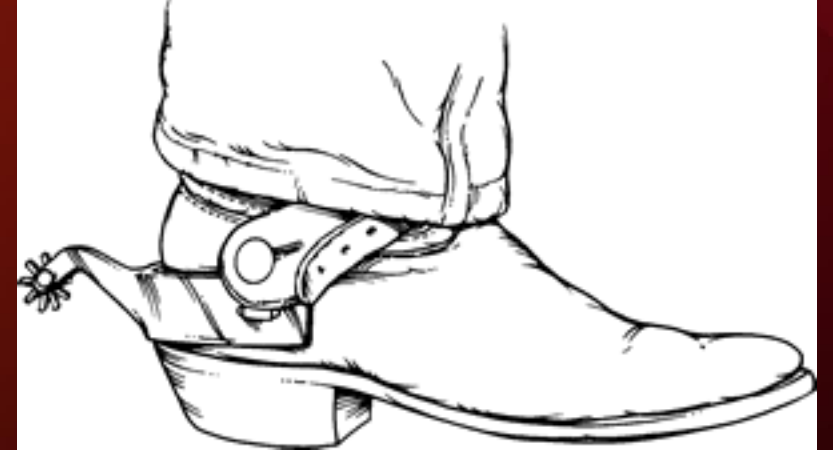
Peter Lindstrom (LLNL)

Sam Li (NVIDIA), John Clyne (NCAR)



WHAT MAKES SPERR UNIQUE?

- It's got a weird name
 - Pronounced like *spur*
- Based on wavelet transform
 - Excellent decorrelation
- Natural support for “flexible-rate decoding”
 - A prefix (subset from the beginning) of the compressed bitstream is still valid for reconstruction, though less accurate
- Natural support for “multi-resolution decoding”
 - A hierarchy of reconstructions with lower resolutions can be obtained without storage or computation overhead



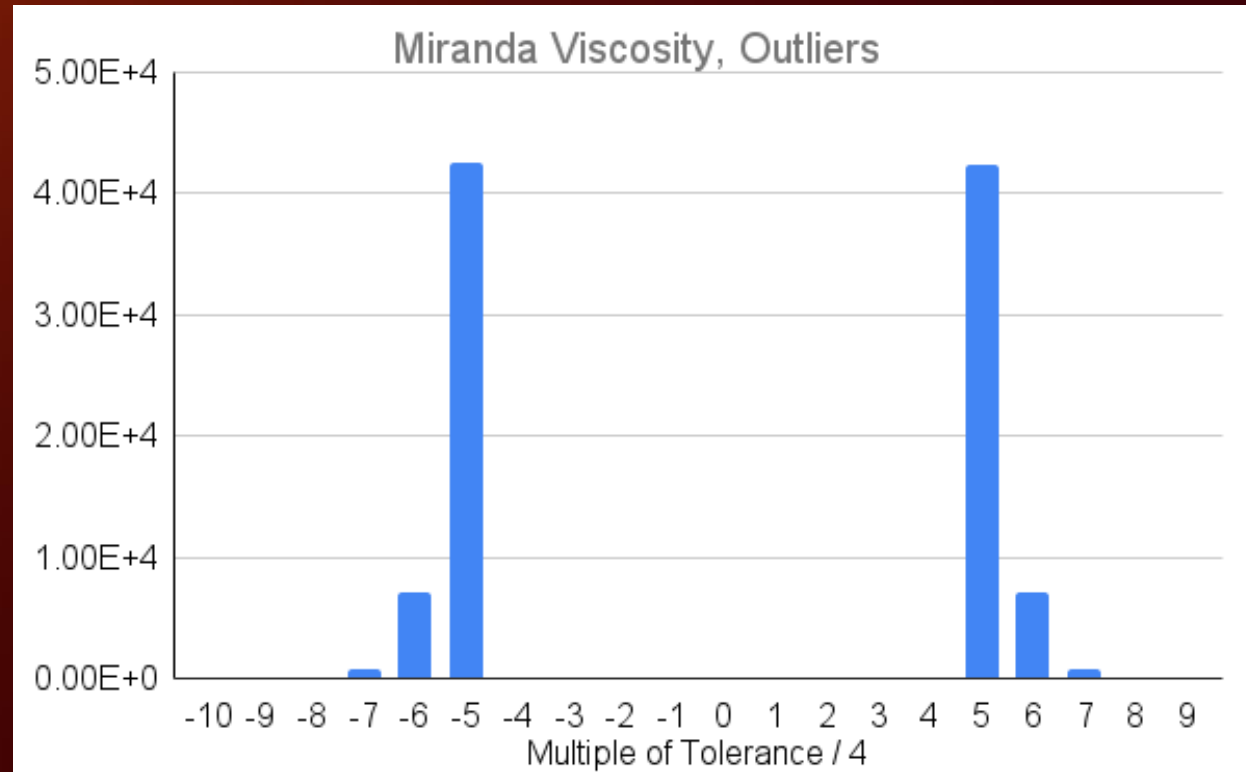
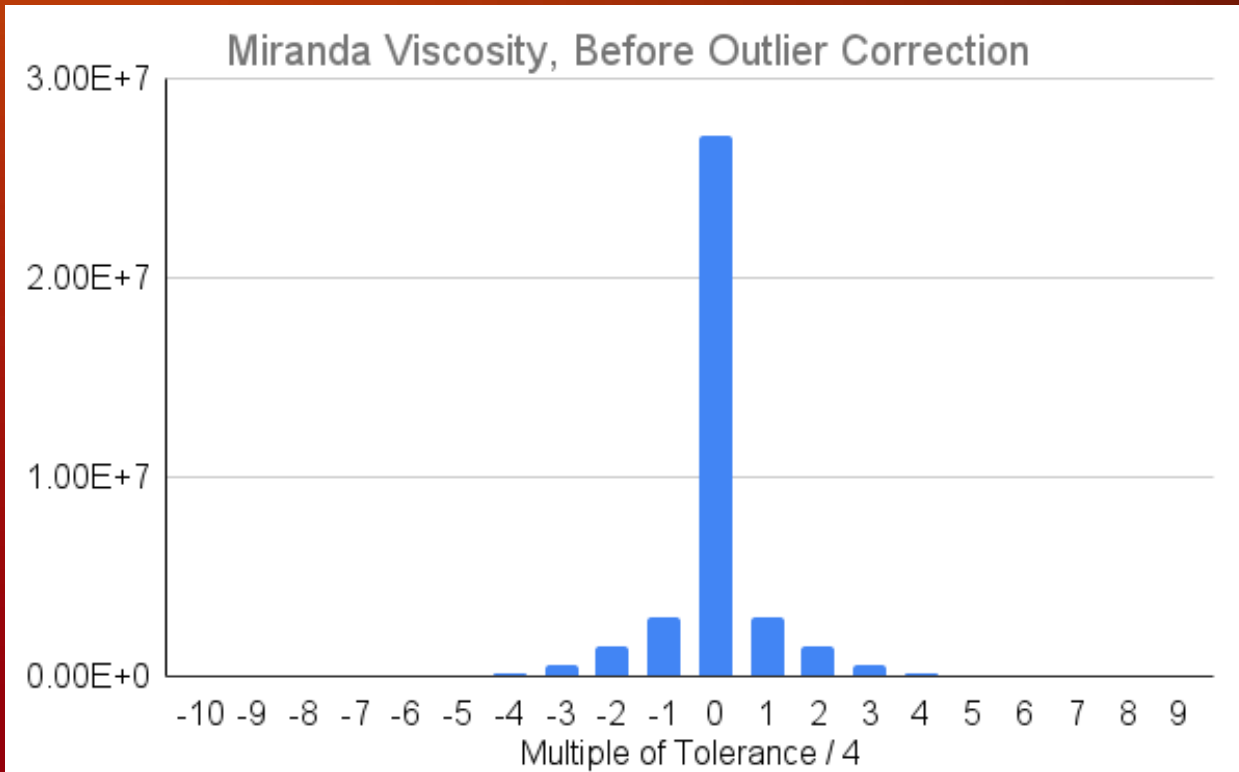
DESIGN CONSIDERATION / MOTIVATION

- Embedded wavelet coding naturally supports **fixed-rate** compression
- **Error-bounded** compression is a must have to be acceptable for scientific data compression
 - Maximum point-wise error (PWE) to be specific
- Observation: error distribution is approximately a bell curve
 - Typically very few data points have large errors
 - Approach: explicitly encode/correct data points that violate tolerance: **outliers**

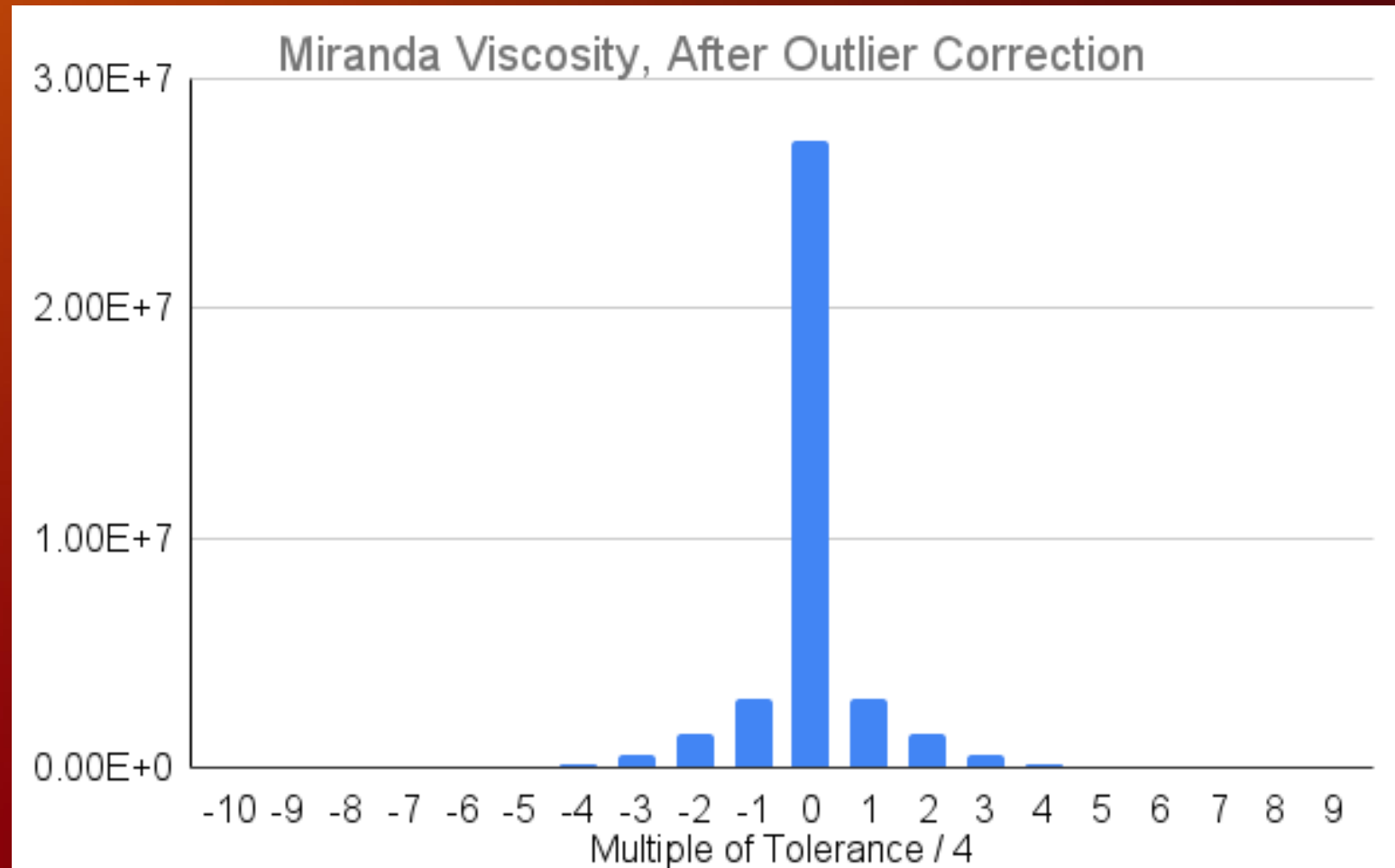


ERROR DISTRIBUTION AND OUTLIER CORRECTION

- Example: Miranda Viscosity field: ~37M data points. Tolerance = $1 \text{e-}8$

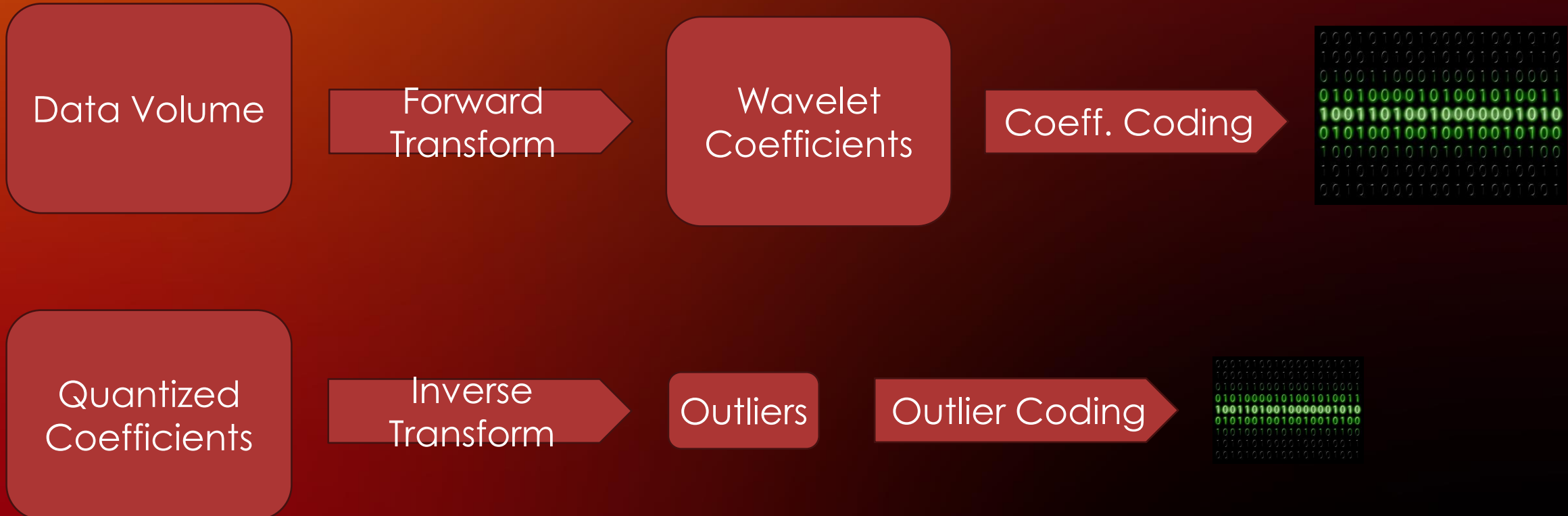


ERROR DISTRIBUTION AND OUTLIER CORRECTION



COMPRESSION PIPELINE

- Two-step process: wavelet compression + outlier correction

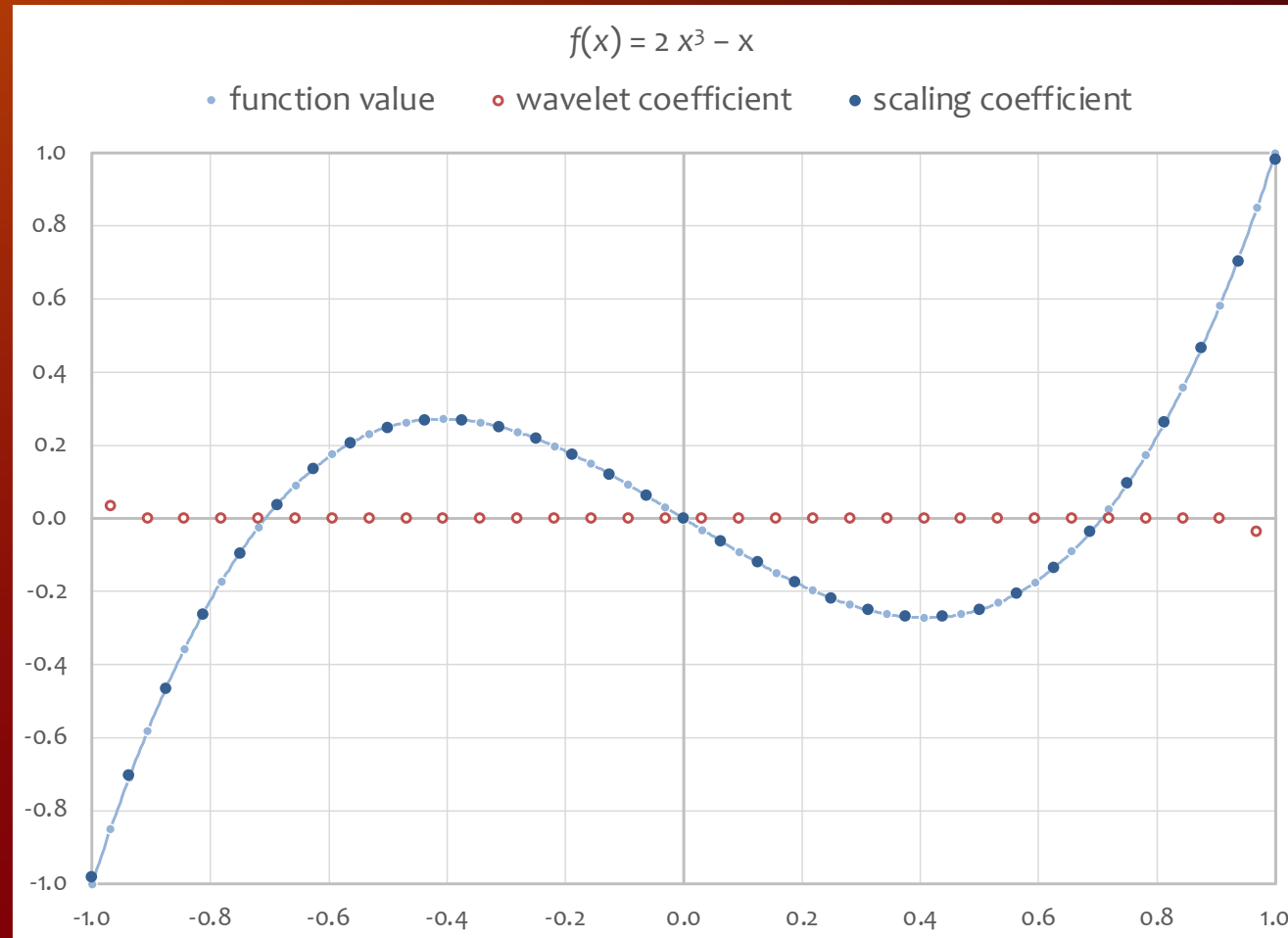


DECORRELATION—WAVELET TRANSFORMS

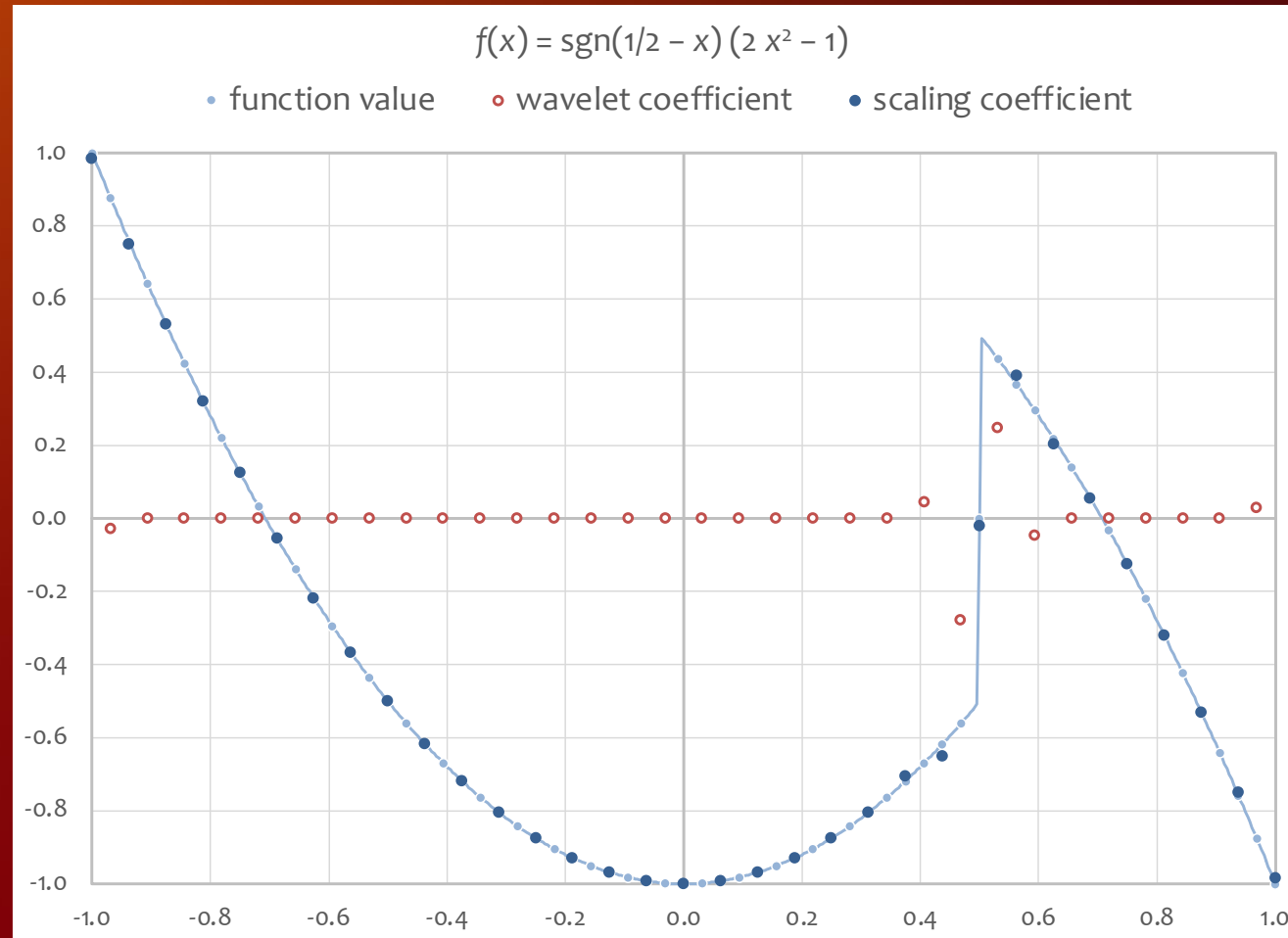
- Wavelet transform linearly maps N values to N coefficients: **reversible!**
- **Scaling coefficients:** low-pass filtered signal (“approximation”)
 - $\lfloor N/2 \rfloor$ “even” samples
 - Promoted to next level: input to transform in multi-resolution hierarchy
- **Wavelet coefficients:** high-pass filtered signal (“details” or “residuals”)
 - $\lfloor N/2 \rfloor$ “odd” samples
- CDF 9/7 wavelets are popular for image & numerical compression
 - Interpolate up to cubic polynomials
 - Perfect interpolation (prediction) implies **all-zero wavelet coefficients**
 - Approximate smooth functions very well with small wavelet coefficients



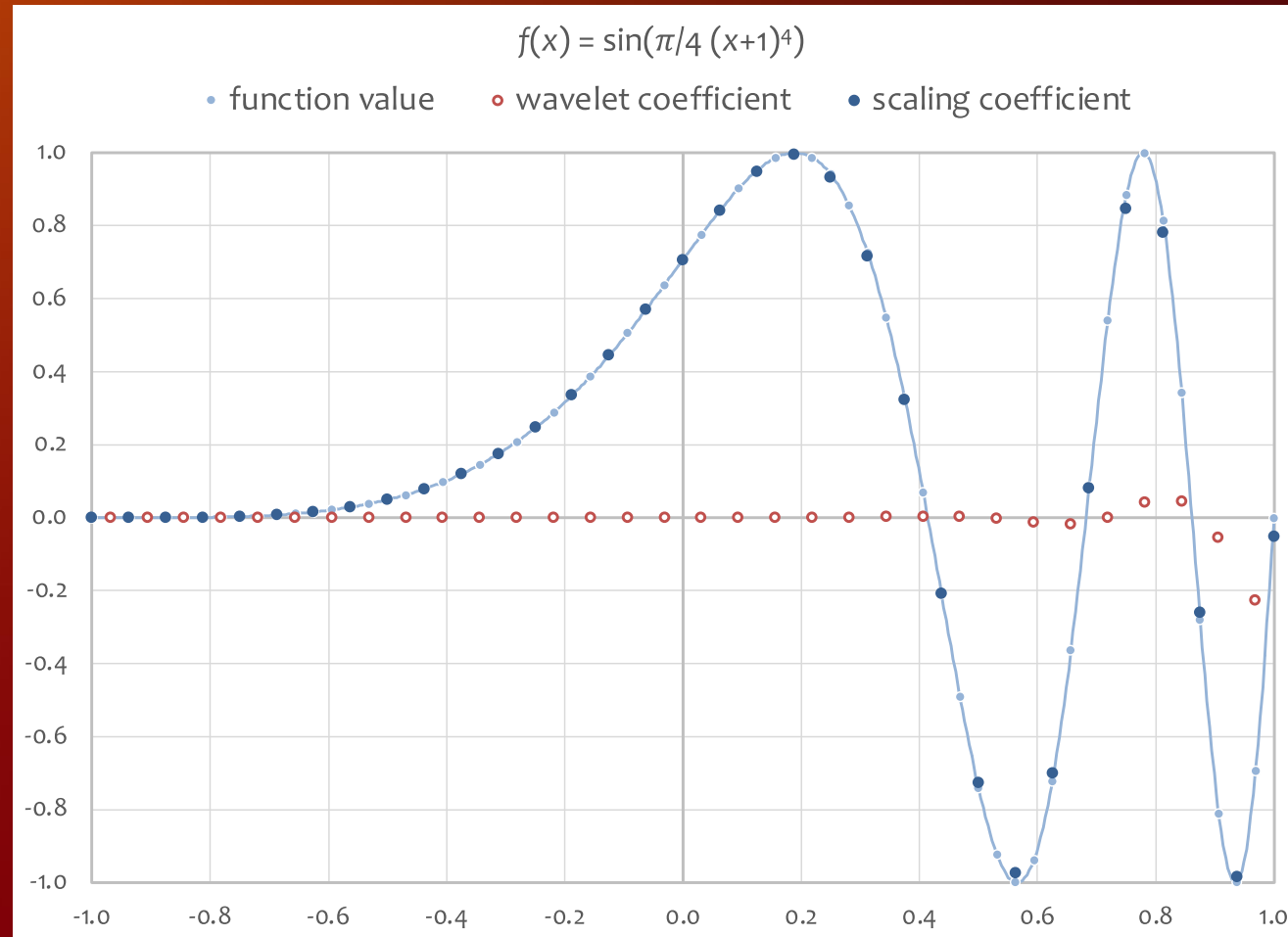
CDF 9/7 EXAMPLE: CUBIC POLYNOMIAL



CDF 9/7 EXAMPLE: DISCONTINUITY



CDF 9/7 EXAMPLE: BEYOND POLYNOMIALS



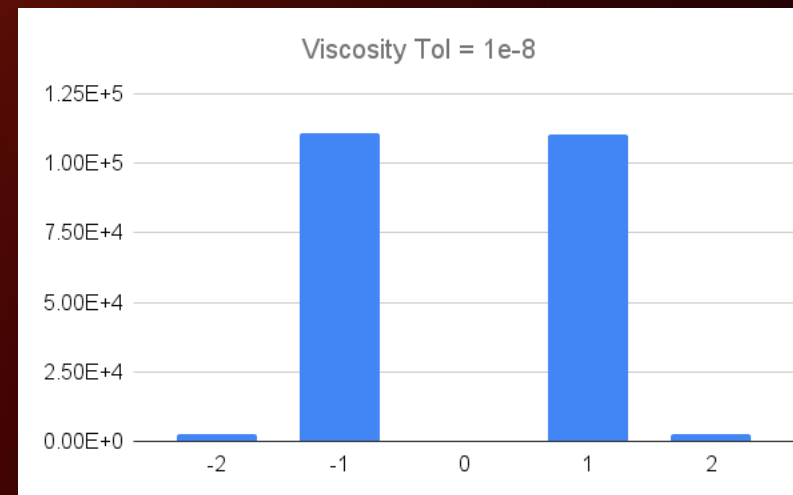
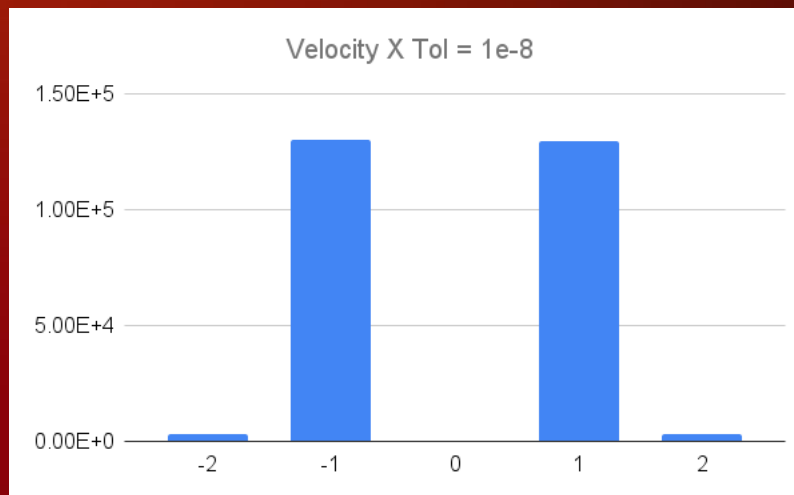
COEFFICIENT CODING

- SPECK [Pearlman et al. 2004] encodes coefficients one bit plane at a time
- Step 1: Locate “significant coefficients” w.r.t. the current bit plane k
 - **Significance test** determines if coeff. magnitude exceeds quantization step $q = 2^k$
 - Divide volume (octree style), perform significance test on each subtree, repeat refinement until individual significant coefficients are located
 - Output: **results of binary significance tests**
 - Spatially clustered significant coefficients share the cost of storing significance tests
- Step 2: Encode current bit plane of already significant coefficients
 - Insignificant coefficients are treated as zeros and are not coded \Rightarrow compression
- Iterate these two steps with the next (less significant) bit plane
 - Already significant coefficients are not tested for significance in next bit plane



OUTLIER CODING

- Apply **inverse wavelet transform** to the quantized coefficients
- Determine “outliers” (usually $< 2\%$) that violate the PWE tolerance
- Encode corrections using SPECK to force outliers into range
 - Integer multiples of $2 \times \text{tolerance}$: we never observed corrections outside $[-4, 4]$
 - Inliers result in zero correction and remain insignificant in this SPECK pass

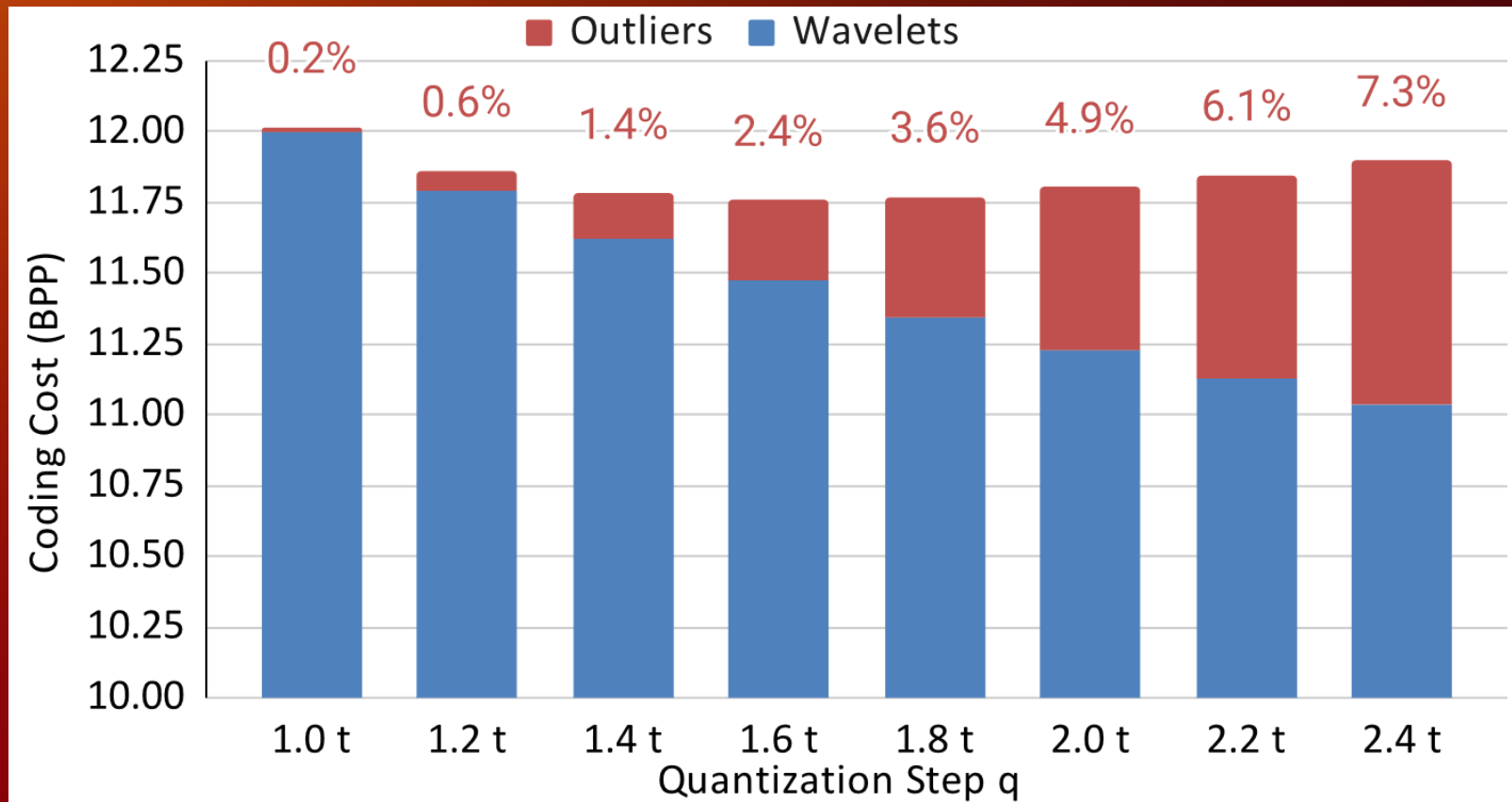


BALANCE: COEFFICIENT AND OUTLIER CODING

- Total storage = Coefficient Storage + Outlier Storage
 - Too much coefficient coding: reduce average error *unnecessarily* low
 - Too much outlier coding: miss out the high efficiency of coefficient coding
 - Goal: find a balance where the **total storage is minimal**
- This balance is governed by quantization step q of last coeff. bit plane
 - Smaller $q \Leftrightarrow$ higher coefficient storage; larger $q \Leftrightarrow$ higher outlier storage
 - Observation: good q values have magnitude similar to the tolerance



BALANCE: COEFFICIENT AND OUTLIER CODING



Empirical Formula: $q = 1.5 \times \text{tolerance}$



CHARACTERISTICS, PERFORMANCE

- SPERR is effective in a wide range of compression qualities:
 - Low quality, high compression ratio: visualization
 - High quality, low compression ratio: saving double-precision output with accuracy comparable to single precision or higher (beats casting to float)
- Parallelization:
 - On CPUs: domain decomposition (256^3 by default)
 - Each subdomain is processed independently
 - On GPUs: no implementation yet (SPECK encoding is hard to parallelize)
- Compared to ZFP and SZ, for a prescribed PWE tolerance...
 - SPERR likely compresses the data more
 - SPERR takes longer to compress



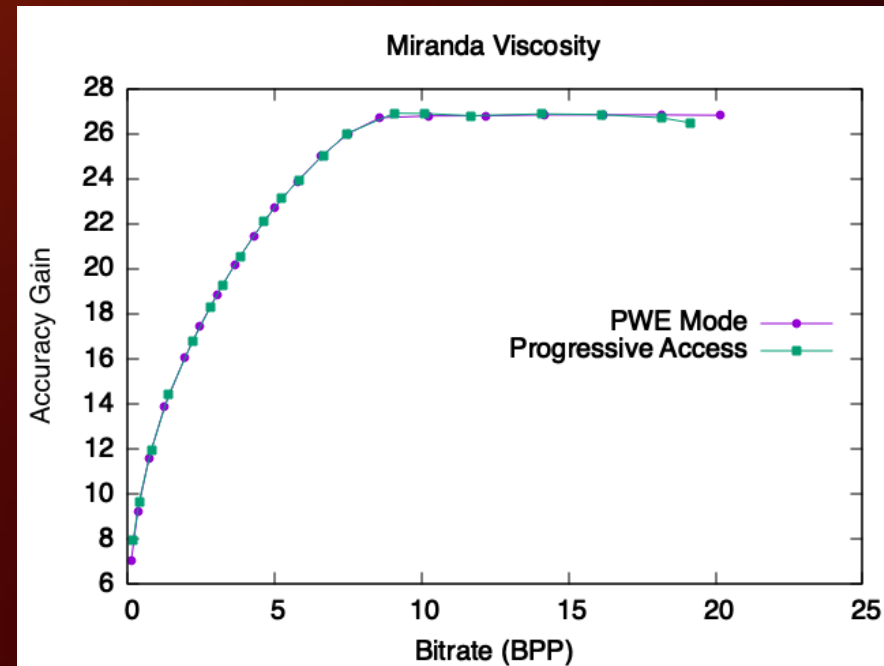
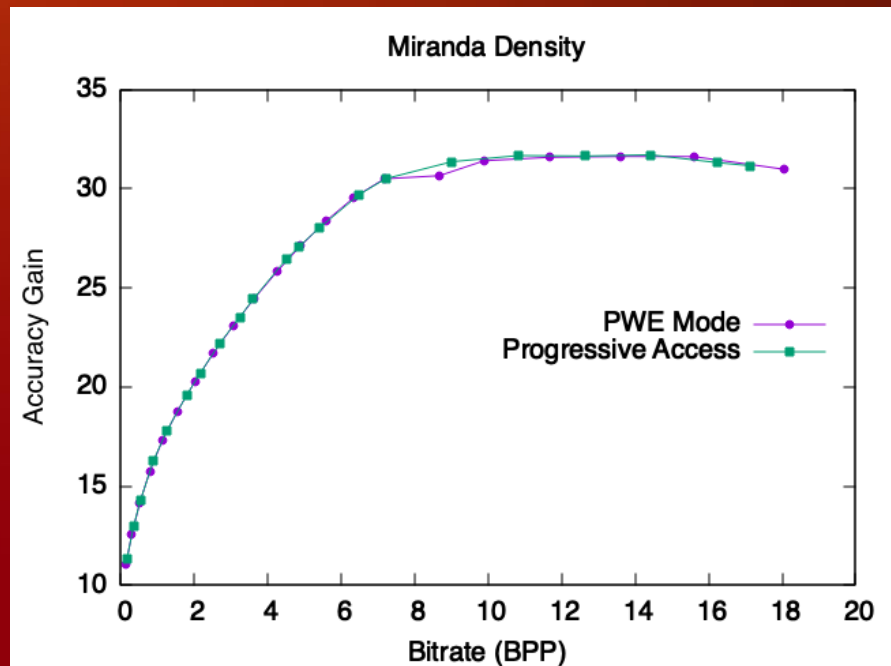
FLEXIBLE-RATE DECODING

- May decompress only a prefix of the compressed bitstream to a bitrate prescribed by the user
 - Essentially progressive-precision decoding (though see caveats later)
- Desirable property: the prefix is the most efficient compressed representation at that bitrate (in terms of average error)
 - There is **no storage overhead** associated with flexible-rate decoding!



FLEXIBLE-RATE DECODING

- Accuracy gain metric: combines bitrate and root-mean-square error
 - Measures amount of information inferred by the compressor (free lunch)
 - $gain = \log_2 \left(\frac{\sigma}{E} \right) - bitrate \approx \frac{SNR}{6.02} - bitrate$



FLEXIBLE-RATE DECODING: CAVEATS

- No built-in error guarantee when using only a prefix of the bitstream
- When domain decomposition is used, flexible-rate decoding must be applied to each subdomain independently
 - Each subdomain has a separate bitstream; no interleaving of streams is done
- No incremental updates
 - When more bits arrive, decompression must be restarted from beginning
 - Not progressive in the usual sense of incrementally refining approximation



INTEGRATIONS/APPLICATIONS

- I/O plugins: HDF5
- Applications: MURaM solar simulation
- Future: cloud-based data portals:
 - Egress costs are high (9 cents per GB)
 - Transmission may be slow
- Future: tiered storage:
 - A fraction (of the compressed bitstream) on hot storage and the bulk on cold storage

