CONFERENCE ON NEXT GENERATION ARITHMETIC · 2026

A Motif-Indexed Dataset of HPC Numerical Kernel Value Distributions for Porting HPC Applications to Posits

Shree Bhargav Komanabelli · Theodore Omtzigt · Paul Dylan Sherman

MOTIVATION

Why this project exists

Posits are a strong fit for HPC	Tapered precision, and the quire gives exact accumulation with no intermediate rounding.
But applications aren’t uniform	Different kernels behave differently, numerically — gains have to be measured, not assumed.
So we built a reference	A dataset that numerically characterizes HPC applications ahead of porting.

BACKGROUND

Three systems

SPICE

Modified Nodal Analysis equations, solved with Newton–Raphson. Each iteration needs a sparse direct solve — KLU.

GROMACS

A molecular dynamics engine. Advances atomic positions under a force field, one timestep at a time.

CHARMM36

A force field GROMACS can run. Same engine, different parameterization — adds the CMAP correction kernel.

THE DATASET

What “numerically characterize” means

Value-distribution histograms

Recorded live from values flowing through each kernel at runtime — not static analysis — in both fp32 and fp64.

Format-fit recommendation

A closed-form calculation that sizes a posit format for a kernel’s measured range.

Berkeley motif tag

Classifies each kernel by computational pattern, so results transfer to other codes sharing that shape.

THE DATASET — FORMAT FITTING

A closed-form sizing method

STEP 1

Regime length and minimum word size at the measured range's worst-case end

→

STEP 2

Smallest posit⟨n, es⟩ meeting the target, searched over es = 0…3

f = 10	fp16-class precision, about 3.3 decimal digits
f = 23	fp32-class precision, about 7.2 decimal digits

Not a search over candidate formats — the format is sized for whichever end of the range has the longest regime, the worst case for precision.

RELATED WORK

Downstream of Quinlan & Omtzigt

Their algorithm	Factor once at low precision, solve, form a residual, correct, and repeat — Wilkinson's refinement, adapted to mixed precision.
Where the quire fits	Round once after an exact quire accumulation instead of after every term, and a low-precision factorization still converges.
Their result	Ten SuiteSparse matrices, n ≤ 240 — a 16-bit posit with equilibration reaches accuracy comparable to IEEE half precision.

We sit downstream of this — the same accumulator mechanism, applied inside KLU's sparse factorization at scales their study didn't test.

Starting hypothesis

“A 32-bit posit with quire accumulation would converge in SPICE where float64 wouldn’t.”

— Professor Gustafson’s proposed topic

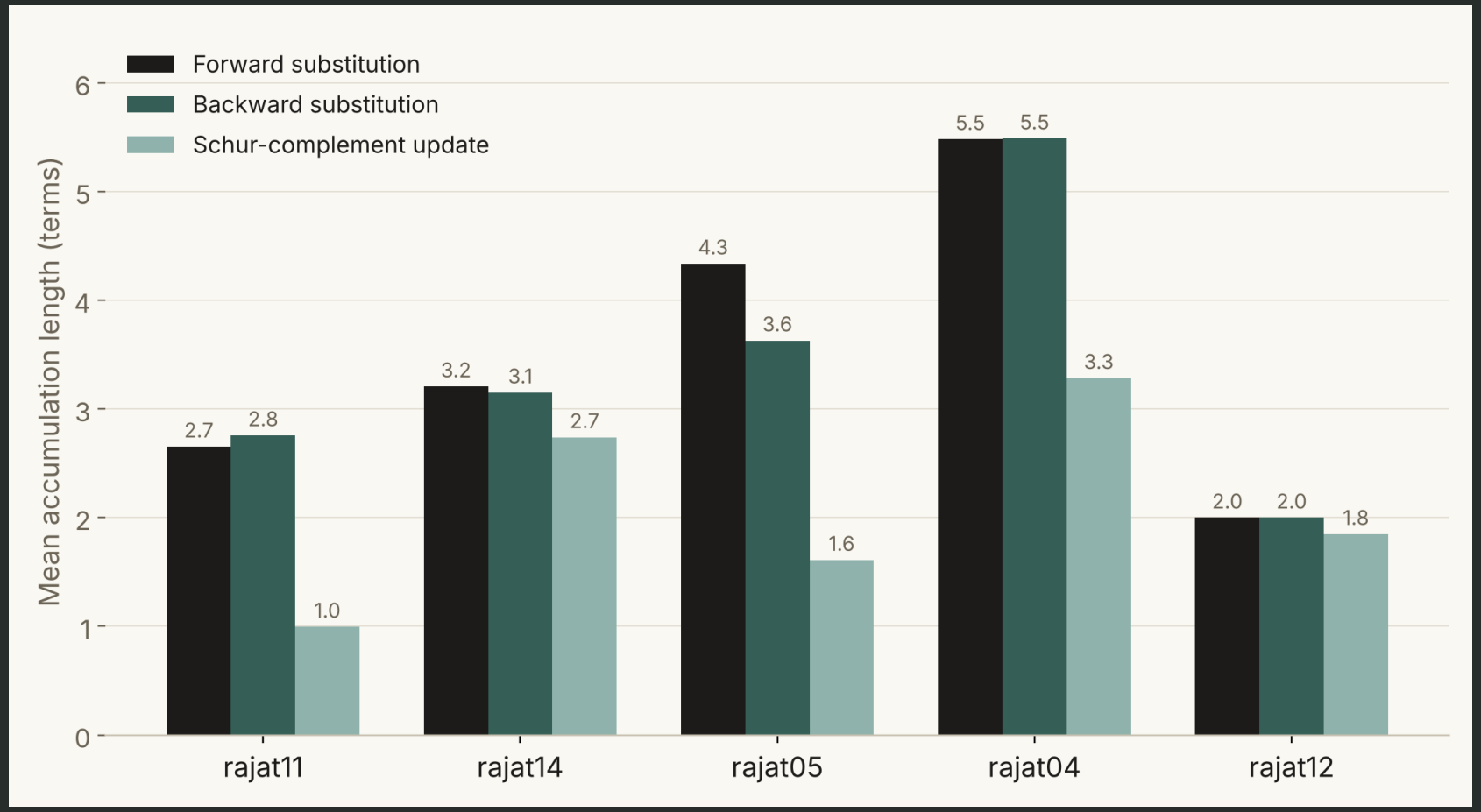
MTL5 + mp-spice — A policy-based accumulator seam swaps plain and quire accumulation on the same solver code path — every comparison ahead is apples-to-apples.

Wiring quire into LU factorization

Three phases	Forward substitution, backward substitution, and the Schur-complement update — the three phases of Gilbert-Peierls sparse LU.
The MTL5 policy seam	Every inner product in all three phases shares one accumulator — swap it for quire and each becomes exact, with one final rounding.
mp-spice as the integration layer	mp-spice wires that seam through KLU's own factorization and solve routines — the solver never references posits or quire directly.

Touching more of the pipeline, on the theory that more chances meant a better shot at a measurable gain.

Minimal gain — and why



Accumulation length stays short in every factorization stage — nothing for the quire to correct.

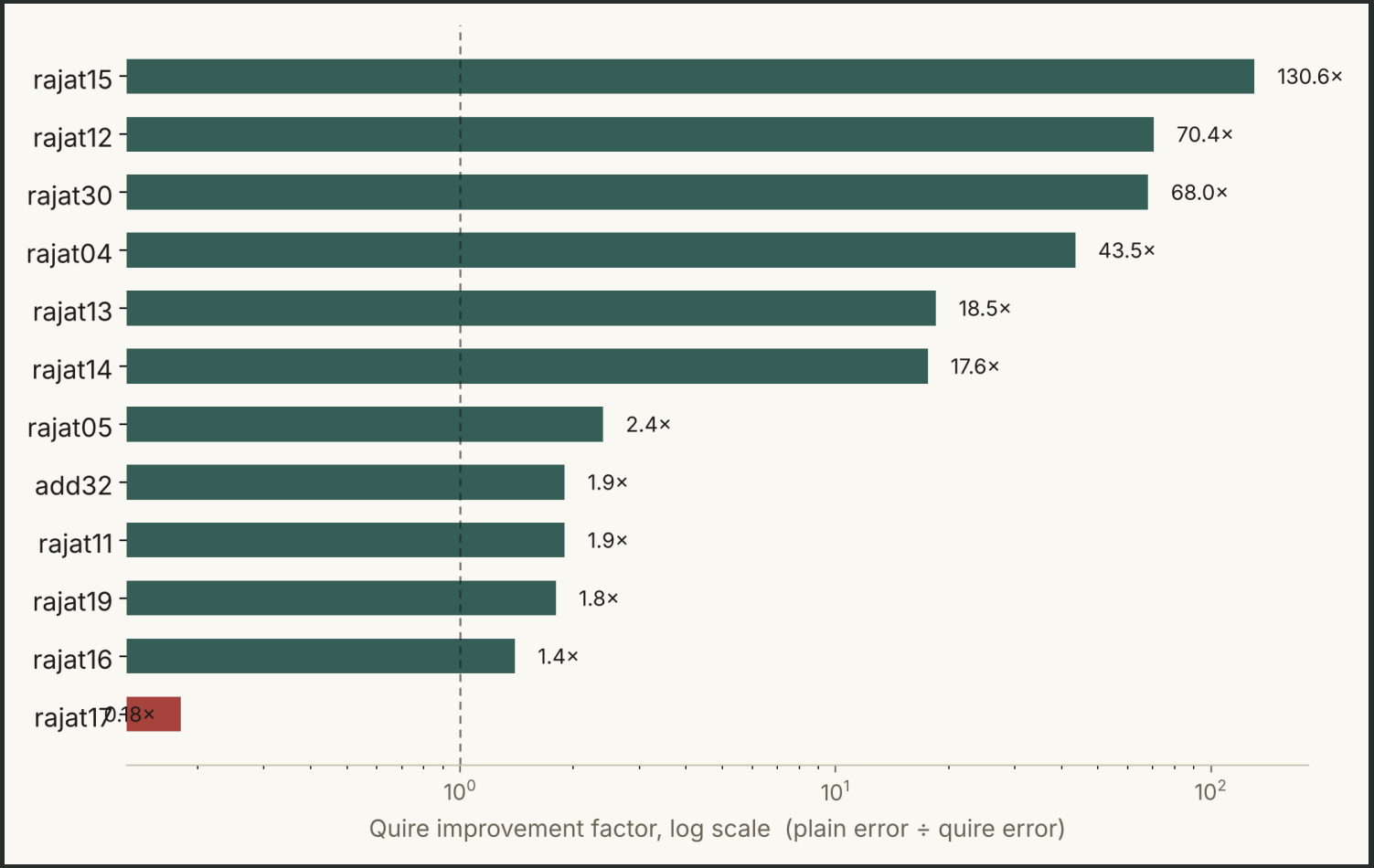
JOURNEY OF PORTING TO SPICE — RESIDUAL ACCUMULATION

Wiring quire into the residual

mp-spice's own work	MTL5's generic refinement core can't vary the residual's accumulator on its own — that capability was built here.
The mechanism	The residual r = b – Ax forms one row at a time, through the same policy seam used in factorization, rounded once at the end.
One choice, same solve	Factorization always runs in plain arithmetic — only the residual's accumulator changes, plain sum or exact quire register.

That single choice at one seam is what the next two results come down to.

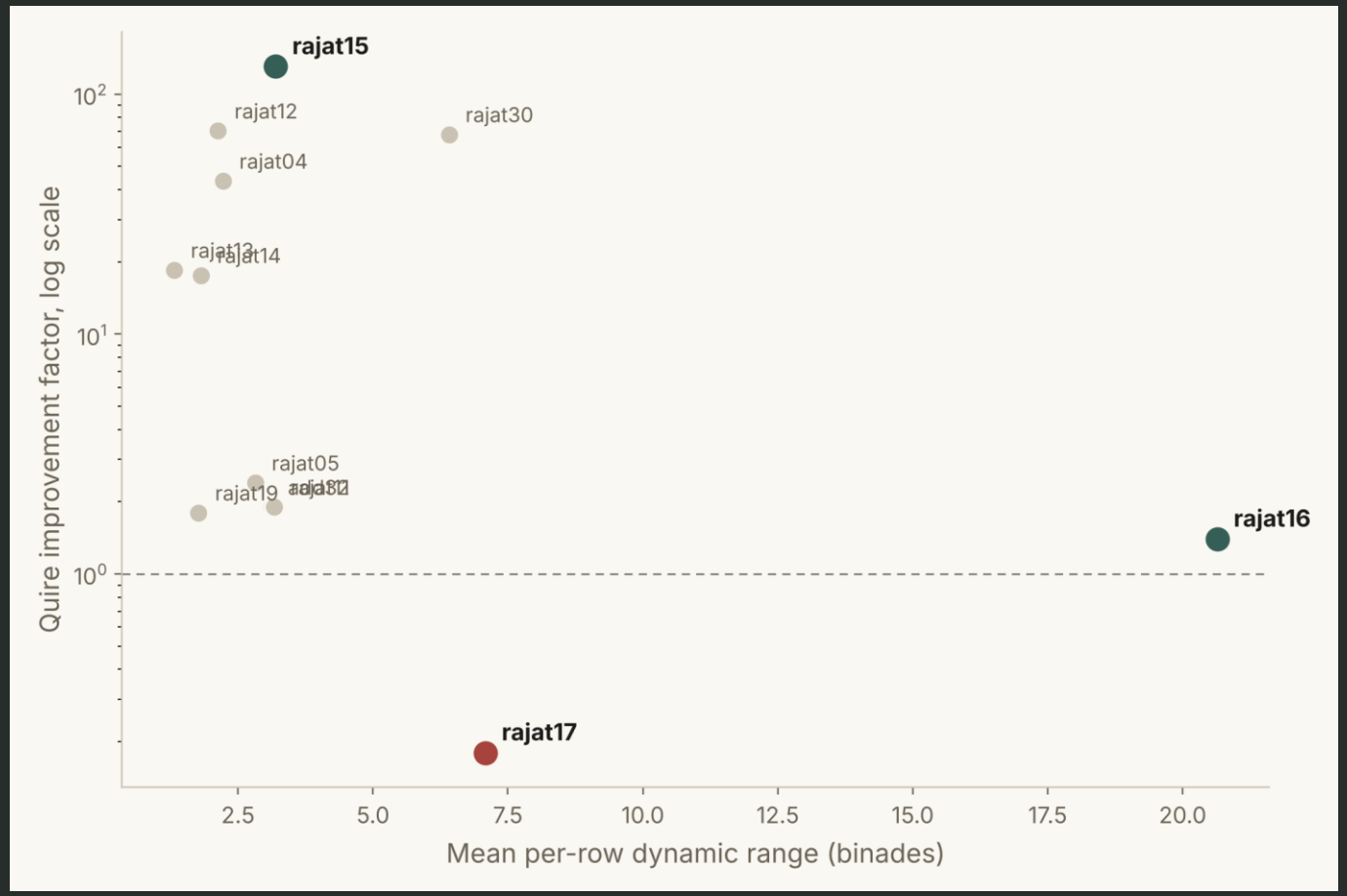
A real gain — and a counterexample



11 of 12 matrices improve, up to **130×**

rajat17 gets worse under quire — about **5.7×**

Dynamic range doesn't explain rajat17



rajat16 — widest range in the dataset, one of the weakest gains

rajat15 — ordinary range, the single best result

A representational problem, not a range problem.

INSTRUMENTING GROMACS AND CHARMM36

Six kernels, one new one

01 Coulomb (nonbonded)

02 Lennard-Jones

03 Force Accumulator

04 Bonded forces

05 PME Reciprocal

06 LINCS

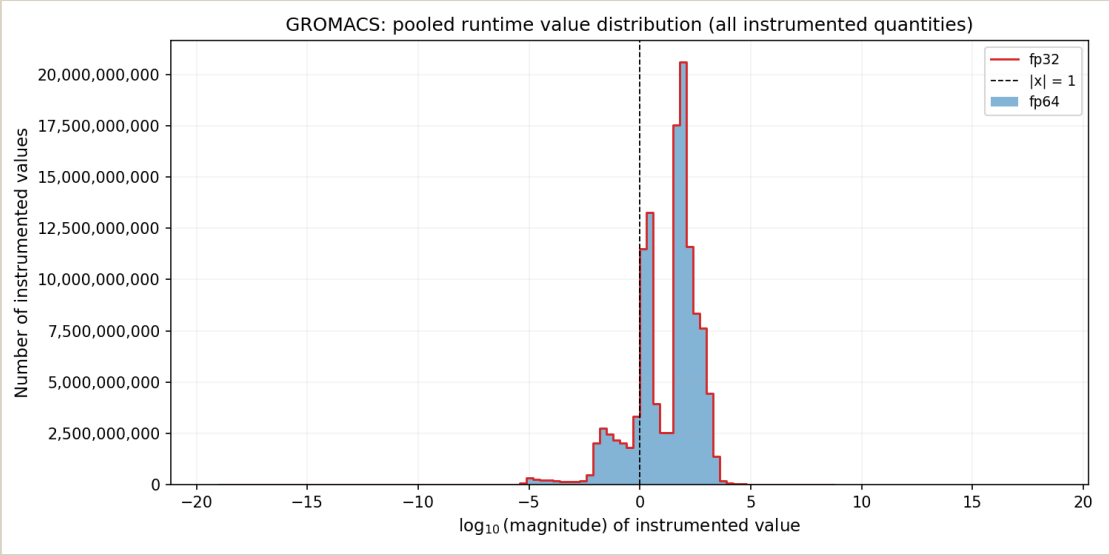
07 Leap-frog integrator

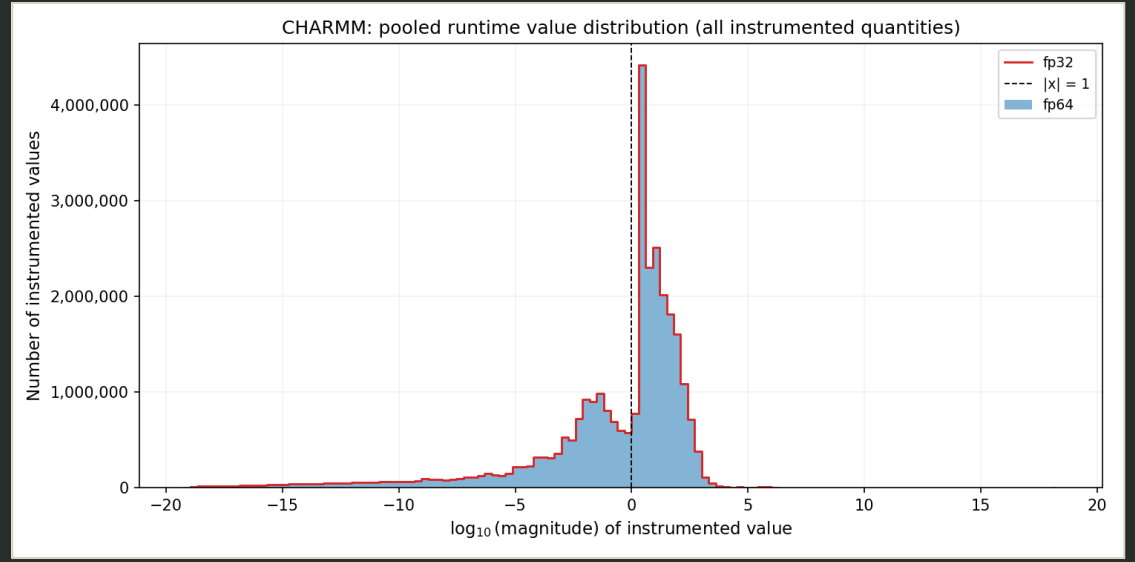
08 CMAP – CHARMM36 only

Scalar reference path GMX_SIMD=NONE forces one force evaluation per interaction, sampled exactly once.

Same source, two precisions Built identically for fp32 and fp64 via GMX_DOUBLE — a true apples-to-apples comparison.

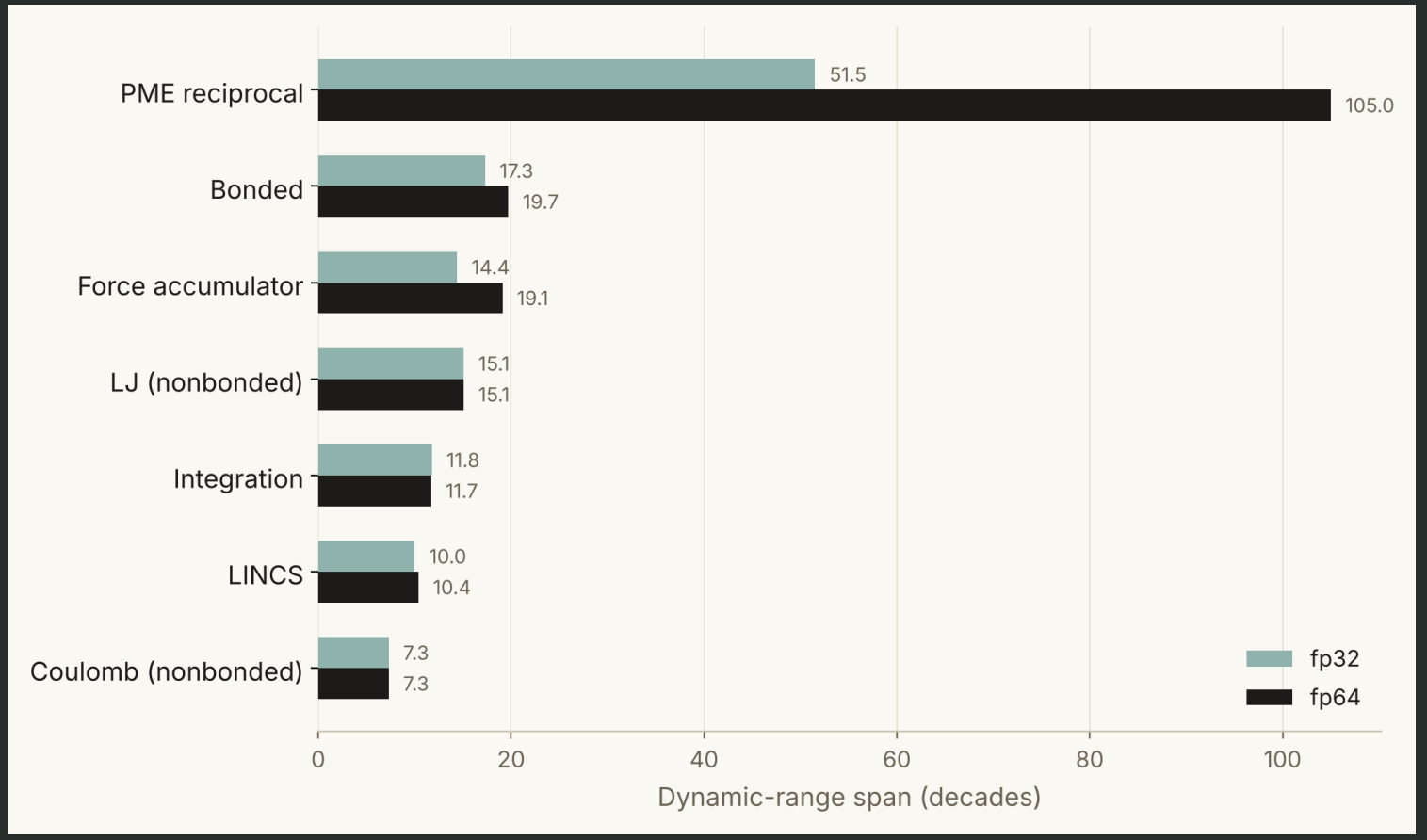
Where the values actually sit





Every instrumented quantity, pooled across all kernels, $\log_{10}$ of magnitude — GROMACS (left) and CHARMM36 (right).

Dynamic range by kernel



PME reciprocal is the outlier;
fp64 spans at least as many decades as fp32 throughout.

Most kernels cluster tightly

Kernel	GROMACS	CHARMM36
Coulomb (nonbonded)	posit⟨30,3⟩	posit⟨30,2⟩
Lennard-Jones	posit⟨34,3⟩	posit⟨35,3⟩
Force accumulator	posit⟨34,3⟩	posit⟨32,3⟩
Bonded forces	posit⟨34,3⟩	posit⟨33,3⟩
PME reciprocal	posit⟨69,3⟩	posit⟨42,3⟩
LINCS	posit⟨33,3⟩	posit⟨32,3⟩
Leap-frog integration	posit⟨32,3⟩	posit⟨31,3⟩
CMAP	–	posit⟨30,2⟩

Same algorithm, same engine — the narrower CHARMM36 parameterization alone roughly halves PME's required width.

Classifying by Berkeley motif

Kernel	Berkeley motif
Coulomb (nonbonded)	N-body methods
Lennard-Jones	N-body methods
Force accumulator	N-body methods
Bonded (+Urey–Bradley)	N-body methods
PME reciprocal	Spectral methods
LINCS	Sparse linear algebra
Leap-frog integration	Dense linear algebra
CMAP	Structured grid

Integration is intentionally “dense” — the motif describes how the kernel computes, a single array-wide update, not how physically coupled its inputs are.

CONCLUSION & FUTURE WORK

A starting point, not an endpoint

Broaden the SPICE side	More test matrices, and more posit-family formats beyond the single configuration tested here — such as b-posits.
Row-by-row diagnostic	Explain why rajat17 fails, by pairing each row's own dynamic range against its residual contribution.
One-pass format fitting	A whole-application recommendation system — then actually port GROMACS, CHARMM, and others to posits.

Thank you.

THANK YOU

Any Questions?