MAY 2018

Improving Mathematical functions in Unum

STUDENT: SIDONG FENG
SUPERVISOR: JOSH MILTHORPE

The Australian National University

Outline

- Background
- Motivation & Project Goal
- Approach
- Experimental Results
- Future Work

The Universal Number: UNUM

- Superset of IEEE types, both 754 and 1788
- Integers->floats->unums
- No rounding(accurate), no overflow to ∞, no underflow to zero
- Safe to parallelize
- They obey algebraic laws!
- Fewer bits than floats
- But… they're new

"YOU CAN'T BOIL THE OCEAN."

—Former Intel exec, when shown the unum idea

UNUM FORMAT

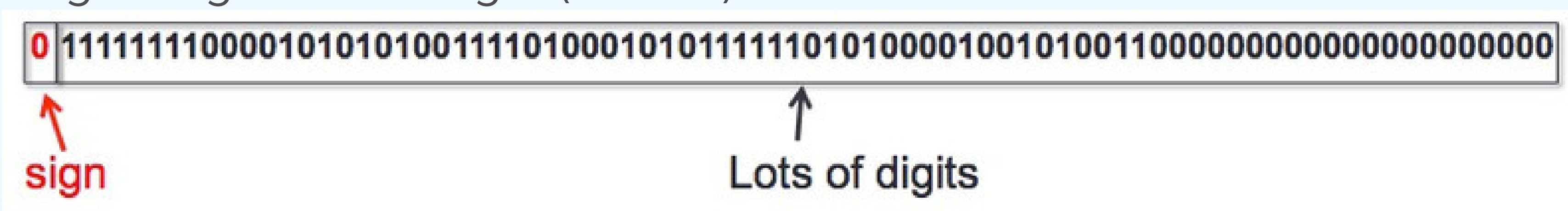
	es bits	fs bits			
s	e	f	0	$es-1$	$fs-1$
sign	exponent	fraction	ubit	exponent size	fraction size

$$x = (-1)^s \times \begin{cases} 2^{2-2^{es-1}} \times \left(\frac{f}{2^{fs}}\right) & \text{if } e = \text{all } \mathbf{0} \text{ bits,} \\ \infty & \text{if } e,\ f,\ es, \text{ and } fs \text{ have all their bits set to } \mathtt{1}, \\ 2^{1+e-2^{es-1}} \times \left(1 + \frac{f}{2^{fs}}\right) & \text{otherwise.} \end{cases}$$

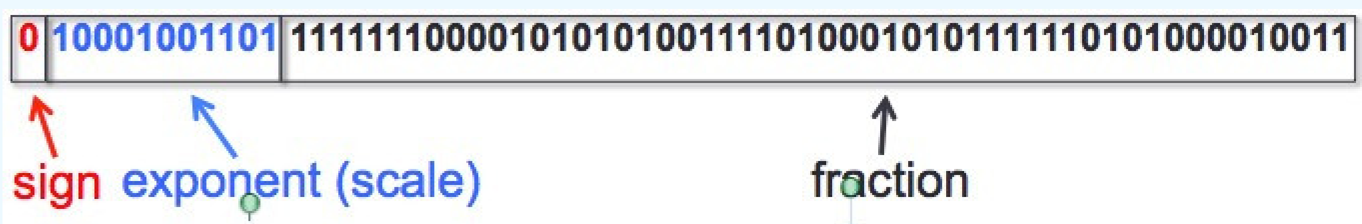
Three format to express a big number

Avogadro's number: ~6.022×10^{23} atoms

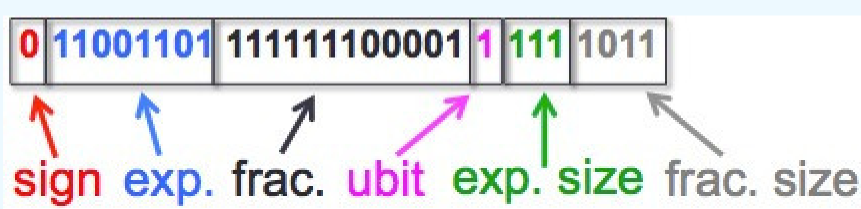
- Sign-Magnitude Integer (80 bits):



- IEEE Standard Float (64 bits):



- Unum (29 bits):



Unum Library by LLNL (Lawrence Livermore National Laboratory)

- hlayer
- ulayer
- glayer

GNU Multiple Precision Arithmetic Library(GMP)

There are several categories of functions in GMP:

- High-level signed integer arithmetic functions (mpz).
- High-level rational arithmetic functions (mpq).
- High-level floating-point arithmetic functions (mpf).

GMP
«Arithmetic without limitations»

Big problems facing computing

- Rounding leads to inaccurate answer
- Too much energy and power needed per calculation
- Not enough bandwidth (the “memory wall”)
- Rounding errors prevent use of parallel methods
- IEEE floats give different answers on different platforms

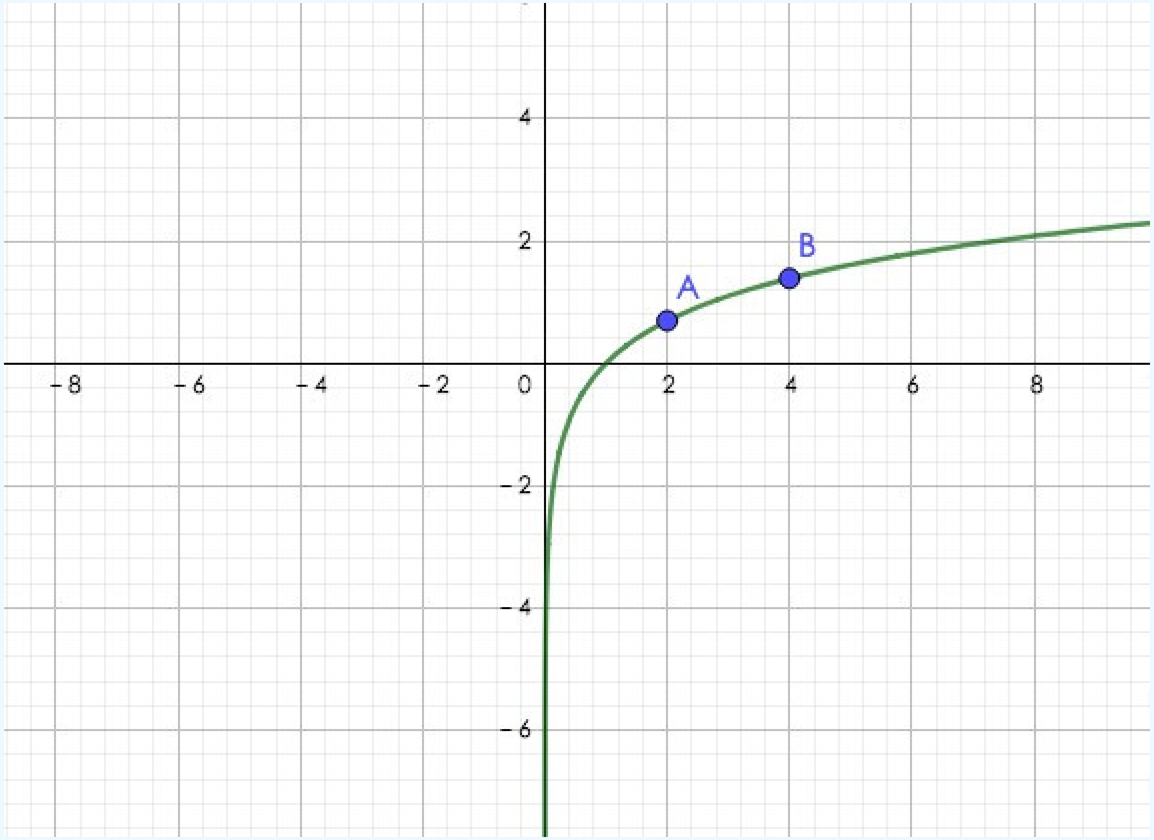
Project Goal

Improve the LLNL library,

Implement Arithmetic operation, Power,

and Transcendental functions, Exponential, Logarithmic.

Implementing the log function for unums



log(A,B) = (log(A) , log(B))

for all positive a and b

Implementing the log function for unums

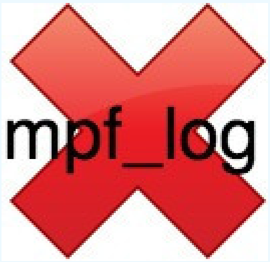
x = | NaN if x = NaN

| NaN if x < 0

| -∞ if x = 0

| +∞ if x = +∞

| mpf_log(x)



mpf_log: Taylor Expansion

$$\log(x) = \begin{cases} 0 & \text{if } x = 1 \\ n \cdot \log(2) + \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\alpha-1)^k}{k} & \text{where} \end{cases}$$

$$x = 2^n \cdot a \text{ and } a \in (0,2]$$

Implementing the exp function for unums

exp(a,b) = (exp(a),exp(b))

x = | NaN if x = NaN

| 0 if x = -∞

| 1 if x = 0

| +∞ if x = +∞

| mpf_exp(x)

mpf_exp: Taylor Series

$$\exp(x) = \begin{cases} 1 & \text{if } x = 0 \\ e^{\text{real}} \cdot e^{\text{Taylor\_Series(decimal)}} & \text{where} \end{cases}$$

$$\text{Taylor\_series} = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

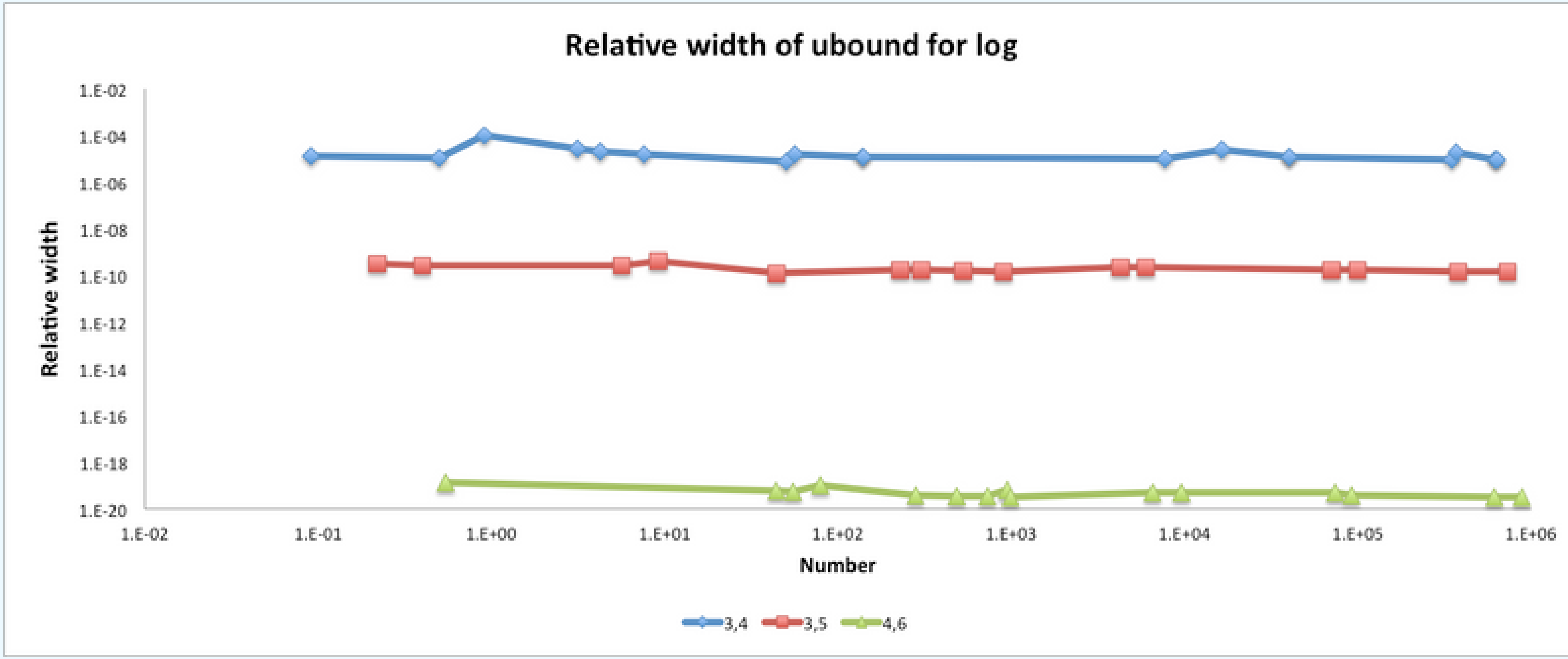
Implementing the pow function for unums

$$x^y = e^{y \log(x)}$$

Experiment

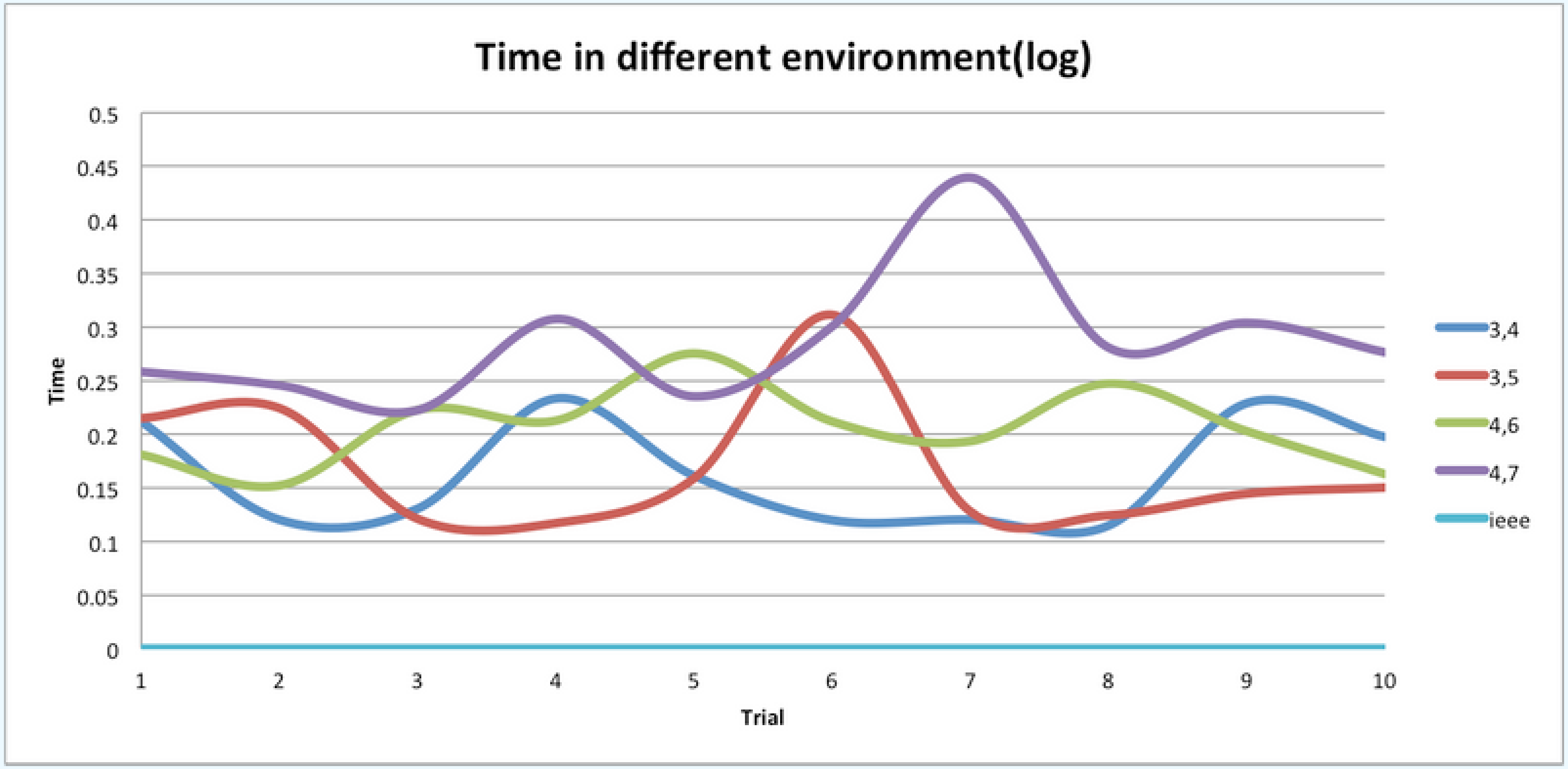
Running Environment: 2.5GHz Intel core i5

Experimental Results for log



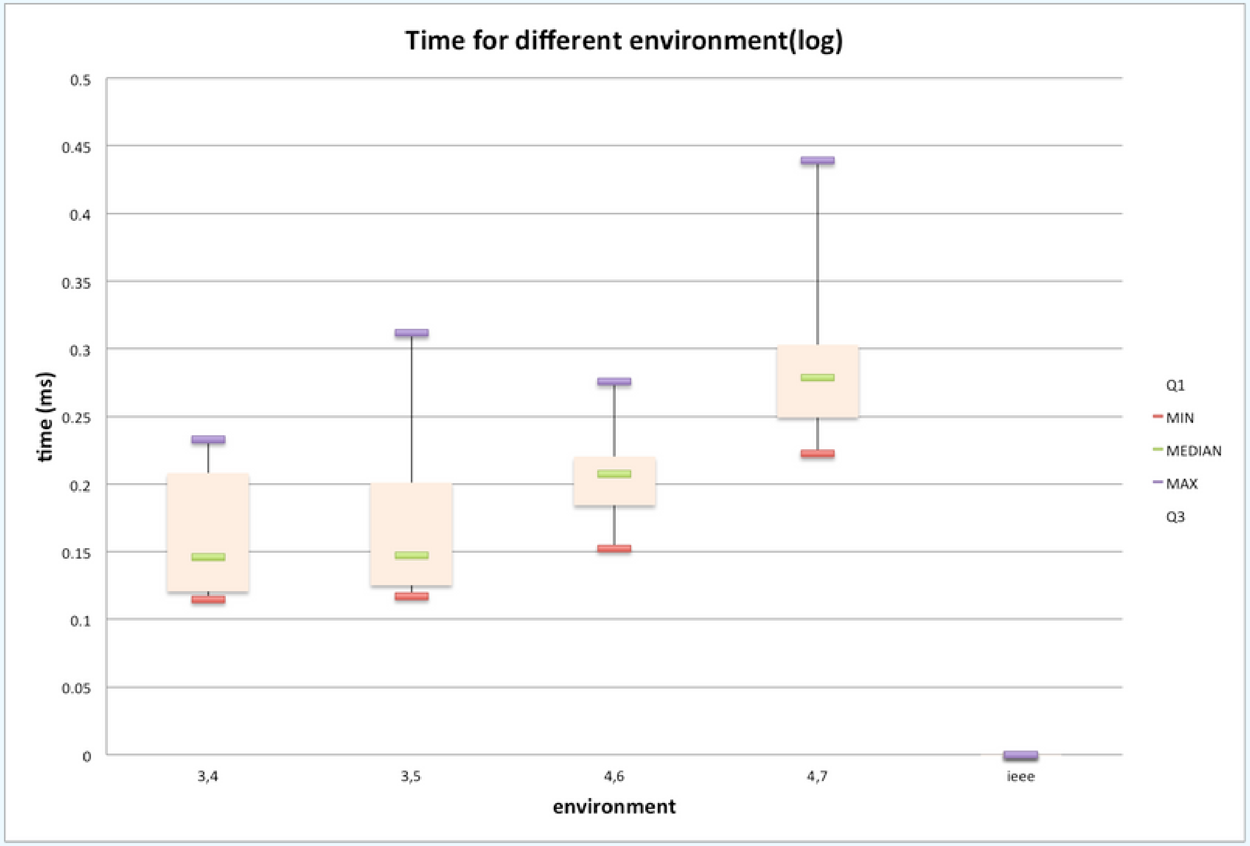
Experimental Results for log

Perform 10000 runs on different Unumenvironments and IEEE respectively.

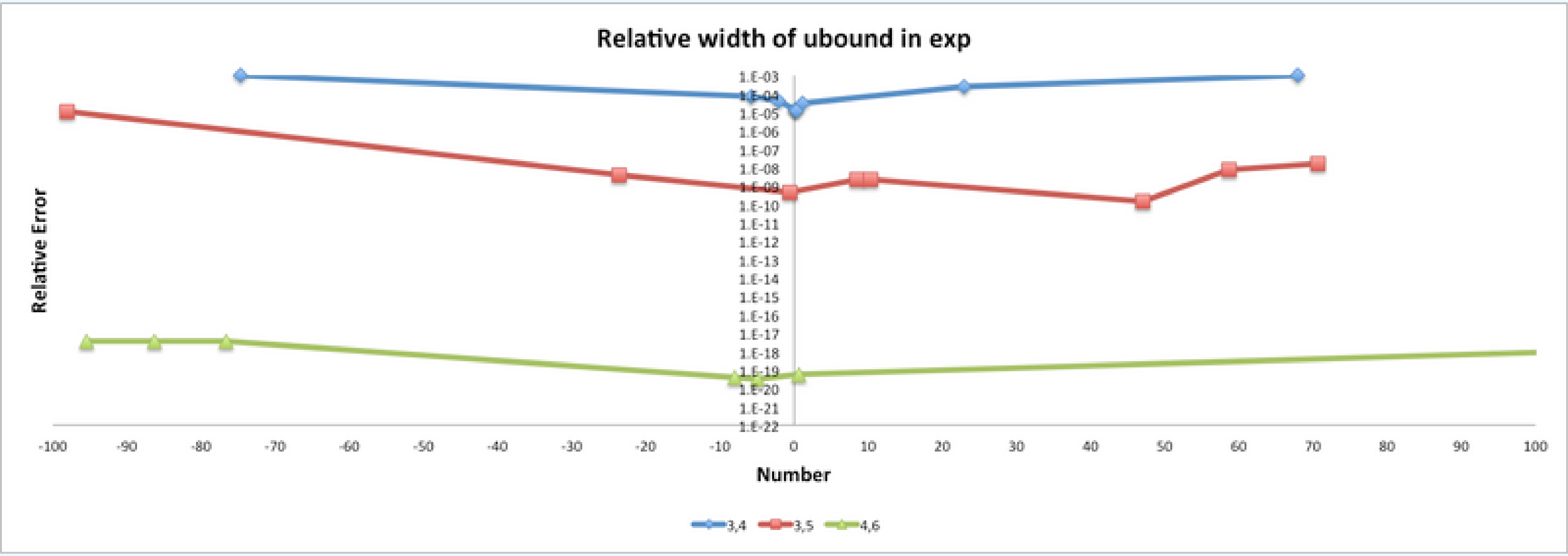


Experimental Results for log

Perform 10000 runs on different Unumenvironments and IEEE respectively.

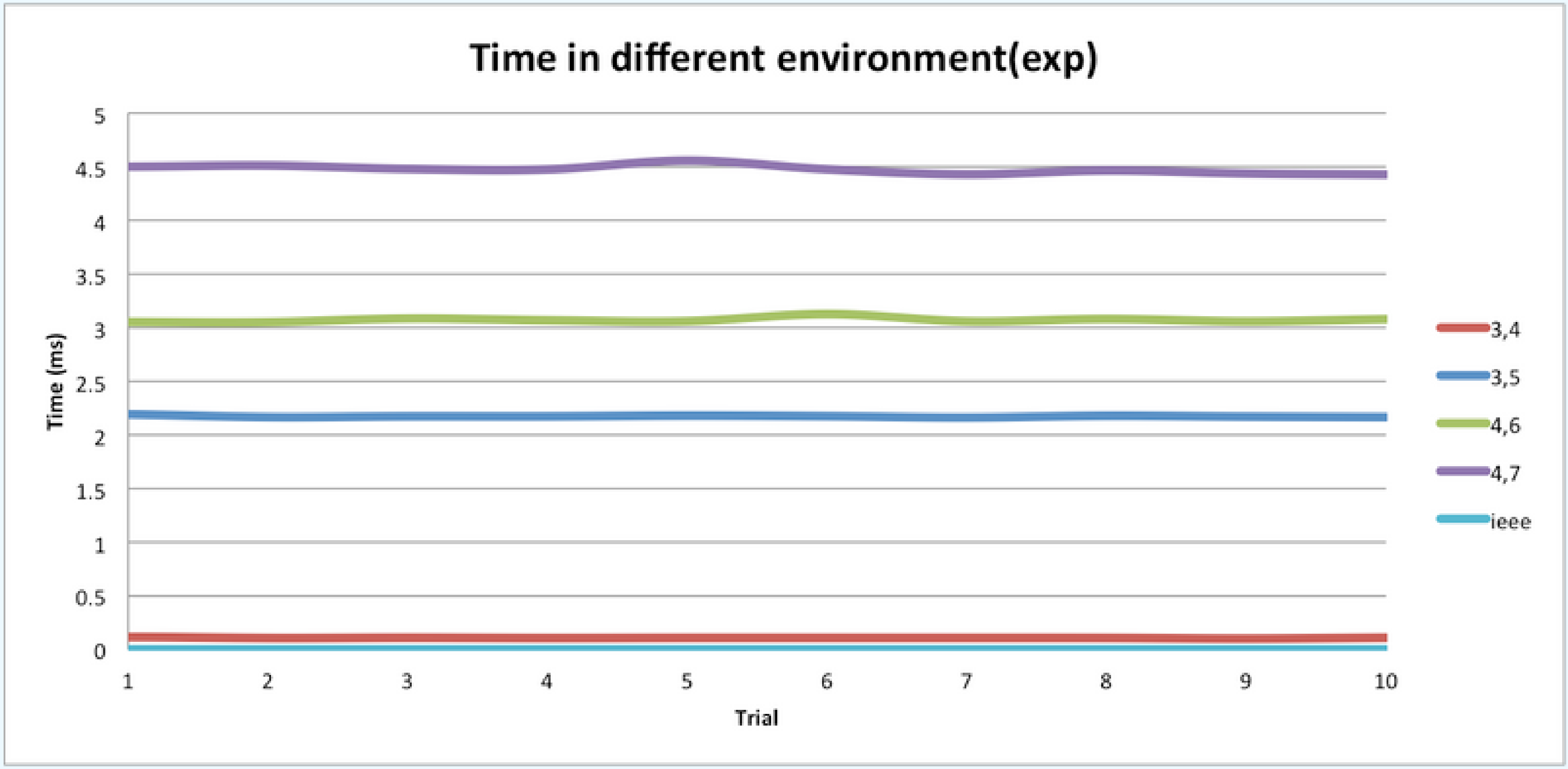


Experimental Results for exp



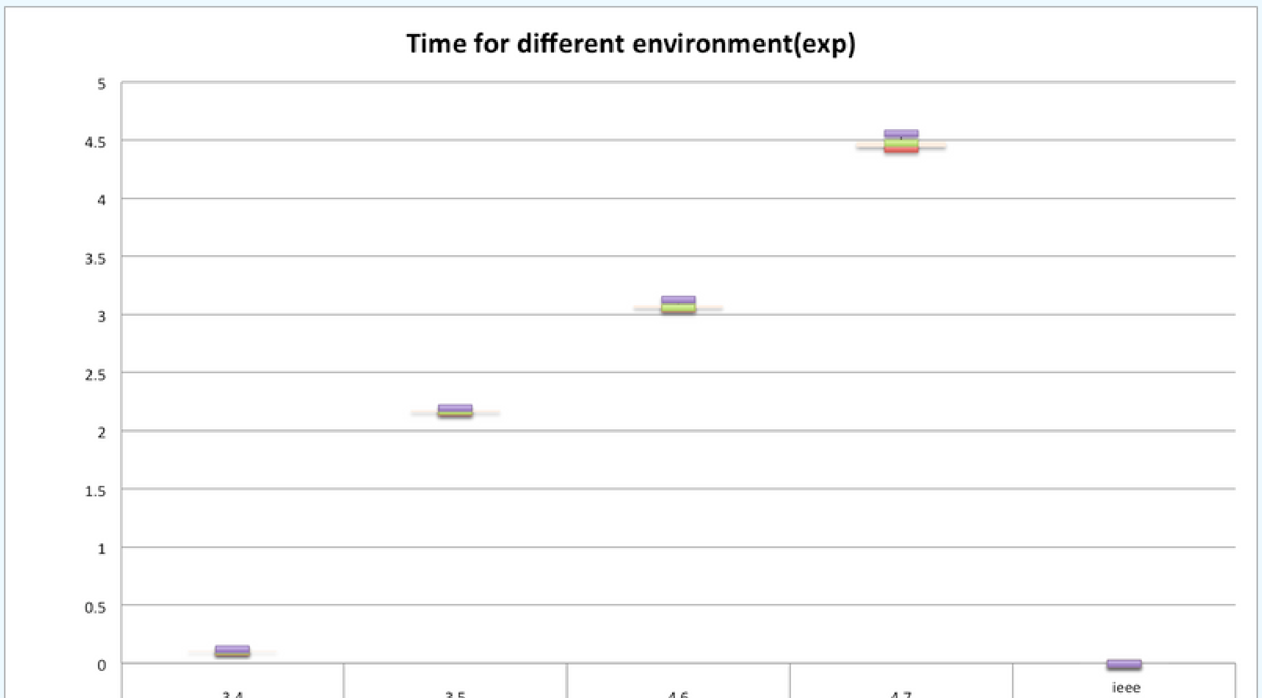
Experimental Results for exp

Perform 10000 runs on different Unumenvironments and IEEE respectively.



Experimental Results for exp

Perform 10000 runs on different Unumenvironments and IEEE respectively.



	3,4	3,5	4,6	4,7	ieee
Q1	0.10611625	2.1660525	3.0577025	4.442435	0.000008745
MIN	0.100924	2.16097	3.05082	4.42549	0.00000874
MEDIAN	0.1076215	2.1716	3.063505	4.47399	8.7475E-06
MAX	0.120828	2.19428	3.12628	4.55678	0.00000899
Q3	0.108771	2.176475	3.0796925	4.49137	0.00000875

Future work

Improving efficiency:

Golden Ratio (log)

nth shifting algorithm (exp, pow)

Conclusion

Unum is much slower than IEEE (so far), however it produces an accurate result. (Gain accuracy, loss performance). If you do want a right result, use unum. More operations, more available testing.

References:

[1] Wikipedia. Unum (number format), 2017.

[2] Gustafson, John L. The end of numerical error, 06 2016.

[3] Kulisch, Ulrich W. Up-to-date Interval Arithmetic from closed intervals to connected sets of real numbers, 07 2016.

[4] Free Software Foundation. What is GNU? September 4, 2009