

PEARC'19 Half Day Tutorial

Floating-Point Analysis Tools



Ignacio Laguna, Harshitha Menon
Lawrence Livermore National Laboratory

Michael Bentley, Ian Briggs, Ganesh Gopalakrishnan
University of Utah

Cindy Rubio González
University of California at Davis

The Floating-Point number system is not new

Then



Zuse Z1 (~1938)

Now

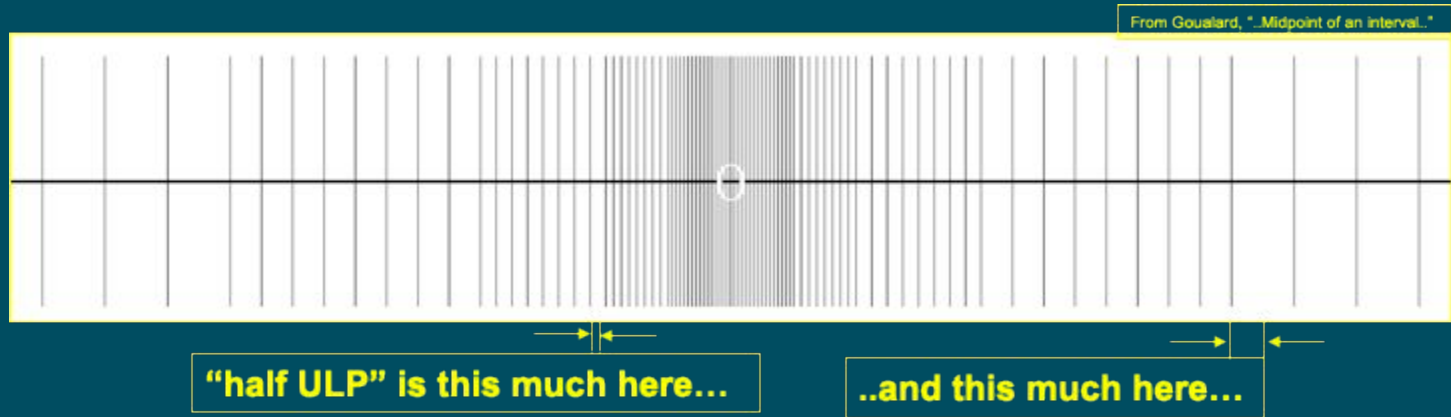
**IEEE Standard for Floating-Point
Arithmetic**



Floating-Point approximates Reals

- Because of rounding, $(x+y)+z \neq x+(y+z)$
 - And many more such identities are violated
- Compilers can change your math
 - $x/y \rightarrow x * (1/y)$
- Rounding errors are non-intuitive
 - Because of the uneven FP number scale

The Floating-Point Rounding is Non-Intuitive



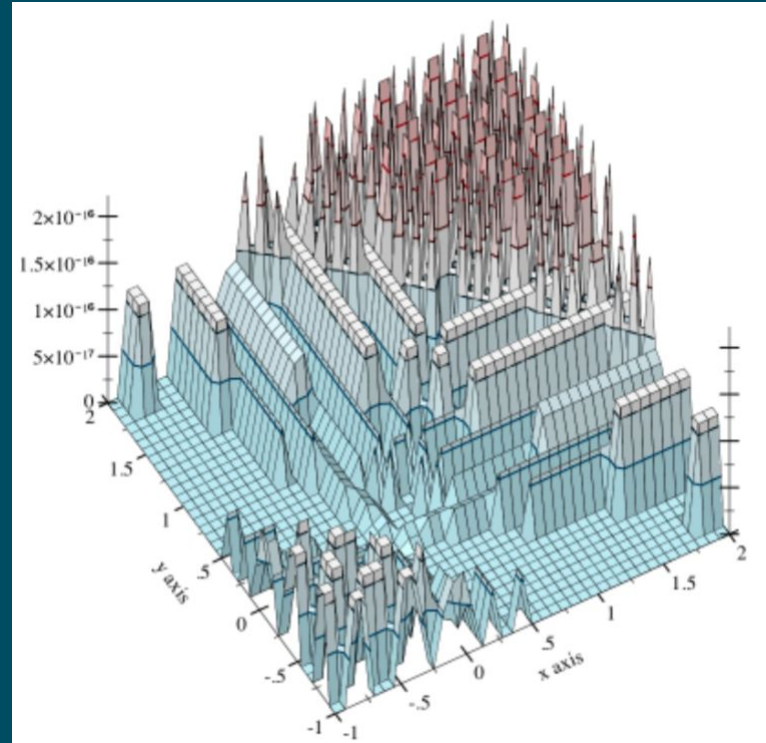
The FP number system tries to span a large range using an “insufficient number of bits”



Reason: The FP error function is highly non-intuitive

E.g.

Rounding error of $(x+y)$ as
a function of x and y





Kahan's observation

Numerical errors are rare, rare enough
not to care about them all the time,
but yet not rare enough to ignore them.
— William M. Kahan



Floating-Point Analysis is Suddenly “Front and Center” in HPC + many other areas

- Allocating needlessly high precision increases data movement
 - Multiple precision types are on the rise
 - Often driven by ML
- The variety of hardware is increasing
 - GPUs and other accelerators
 - Their normal behaviors as well as EXCEPTIONS are on the rise
- Compilers exploit floating-point in an increasing number of ways
 - Compiler flags mean different things
 - Compilers may heed your flags selectively



Frenetic pace of FP research now

- Multiple conferences
- Many sessions per conference
- Many different issues

Very little that is tangible for a practitioner to try some of these out

Some good resources do exist (will put it on our website)

E.g., fpbench.org

Michael O. Lam, Floating Point Analysis Research



Goals of this Tutorial

- **Introduce FOUR mileposts in your repertoire of knowledge**
 - Four tools you can practice during the tutorials
 - You can apply them in your own projects!
- **We are a resource you can count on during your future work**
 - We are invested in multiple research projects in this area
 - We know many more researchers and practitioners whose work we can refer

We hope to build a community of researchers and practitioners

See us (if you like) at SC'19 for a full-day tutorial on this + more topics!



Specifics of this tutorial

- **FPChecker**
 - Helps detect FP Exceptions on GPUs
 - **Outcome:** You can use it on your Clang-based GPU projects today!
- **FLiT**
 - Helps diagnose why your compiler optimization produces unacceptable answers!
 - **Outcome:** You can apply it in the context of your CPU projects today!
 - No Clang or Intel dependency!
- **Precimonious**
 - Learn the benefit of precision tuning on actual code
 - **Outcome:** You may apply it in the context of your Clang-based CPU codes today!
- **Adapt**
 - Learn what Automatic Differentiation is, plus how it helps tune precision
 - **Outcome:** You may apply it in the context of your CPU codes today!
 - No Clang, Intel, or CPU specificity



Access to AWS Instances

- You will be given access to AWS instances
 - User, password, and IP address will be provided

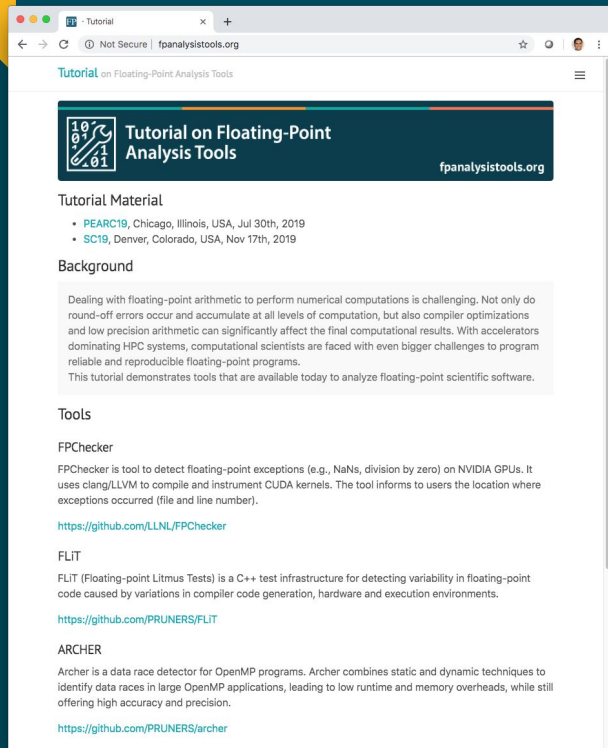
- How to access your instance:

```
ssh user@1.2.3.4
```

- Exercises for each module located in user's /home directory

```
/home/user1/  
|---Module-T00L1  
|   |--exercise-1  
|   |--exercise-2  
|   |--exercise-3  
|---Module-T00L2  
|   |--exercise-1  
|   |--exercise-2  
|   |--exercise-3  
...
```

Website & Schedule



<http://fpanalysistools.org/>

The screenshot shows the website fpanalysistools.org in a browser. The page title is "Tutorial on Floating-Point Analysis Tools". Below the title, there is a section for "Tutorial Material" listing two events: "PEARC19, Chicago, Illinois, USA, Jul 30th, 2019" and "SC19, Denver, Colorado, USA, Nov 17th, 2019". A "Background" section follows, explaining the challenges of floating-point arithmetic and the purpose of the tutorial. The "Tools" section lists three tools: "FPChecker" (a tool to detect floating-point exceptions), "FLIT" (Floating-point Litmus Tests), and "ARCHER" (a data race detector for OpenMP programs). Each tool has a brief description and a link to its GitHub repository.

Time	Module	Presenter	Slides
1:30pm - 1:40pm	Introduction	Ganesh, Ignacio	slides
1:40pm - 2:20pm	FPChecker	Ignacio	slides , source
	Key Topics:		
	- Floating-point exceptions, GPUs, CUDA		
2:20pm - 3:00pm	FLIT	Ganesh, Mike, Ian	slides , source
	Key Topics:		
	- Compiler optimizations, floating-point variability		
3:00pm - 3:30pm	Break		
3:30pm - 4:10pm	Precimonious	Cindy	slides , source
	Key Topics:		
	- Floating-point mixed-precision, tuning		
4:10pm - 4:50pm	ADAPT	Harshitha	slides , source
	Key Topics:		
	- Algorithmic differentiation, input sensitivity		