IEEE P3329

Impact of software and algorithms in quantum computing energy efficiency

Adrien Suau (QraftWare)
Jean-Baptiste Latre (Qualitative Computing)

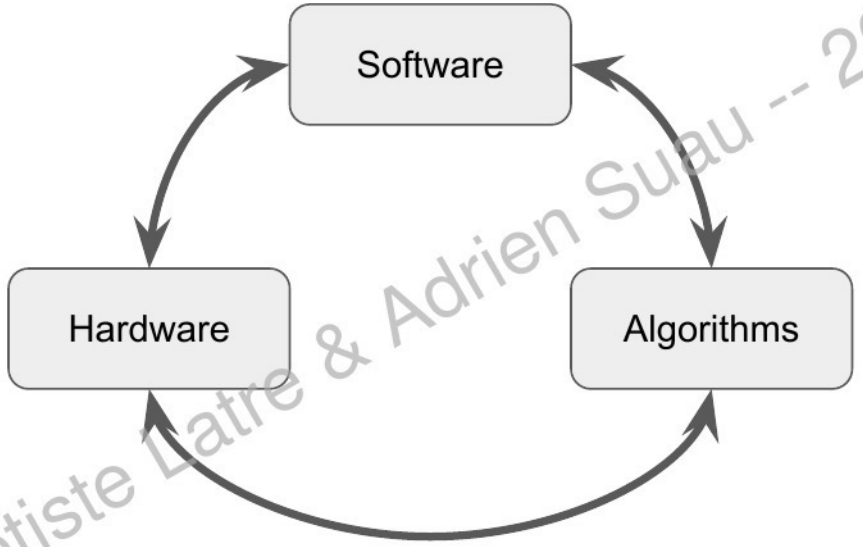
28th Nov 2023

Background

Qualitative Computing (2022)

QraftWare (2023)

Define common grounds for co-design:

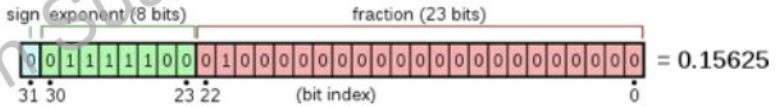


Virtuous or negative spiral

Feel free to interrupt for questions at any time

Practical impact of models

Theoretical Wonderland vs Finite precision computations IEEE 754



Archetypal phenomenon: **catastrophic cancellation**

Huge numerical errors due to subtraction of two nearby quantities

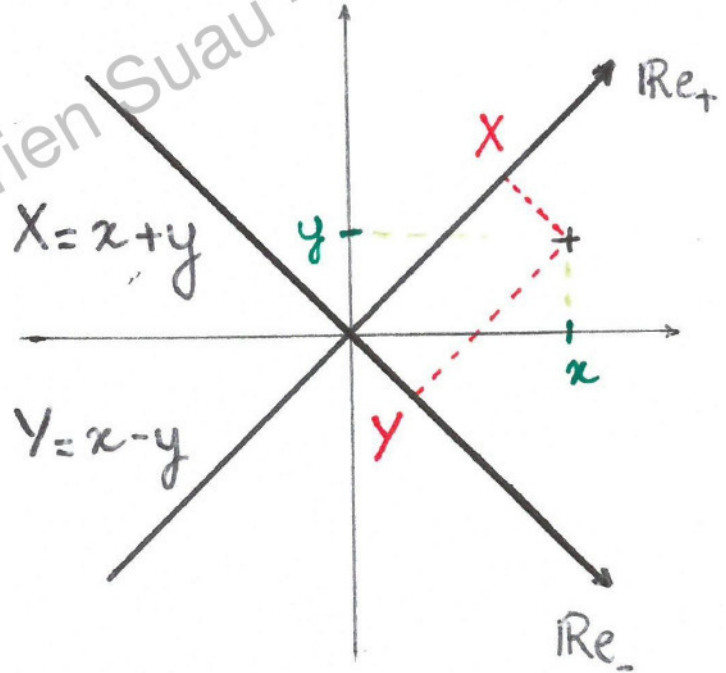
William Kahan

Change of basis

$$z = (x, y) \in \mathbb{R}^2 \quad M = \begin{bmatrix} x & y \\ y & x \end{bmatrix}$$

Hadamard matrix

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad P = \frac{1}{2}H$$



Change of basis

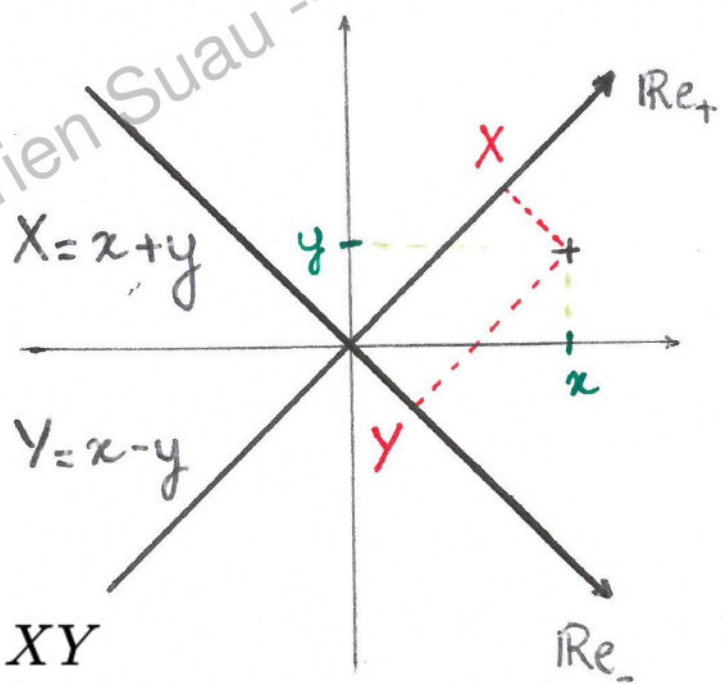
$$z = (x, y) \in \mathbb{R}^2 \quad M = \begin{bmatrix} x & y \\ y & x \end{bmatrix}$$

Hadamard matrix

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad P = \frac{1}{2}H$$

$$P^{-1}MP = \begin{bmatrix} x+y & 0 \\ 0 & x-y \end{bmatrix}$$

$$detM = x^2 - y^2 = (x+y)(x-y) = XY$$

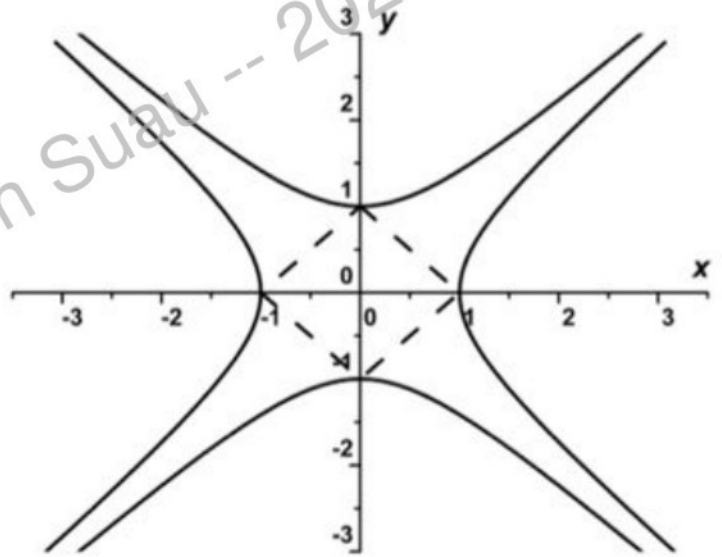


Difference between exact / finite precision computations

$$M = \begin{bmatrix} x & y \\ y & x \end{bmatrix}$$

Quadratic iteration $M_0 = M$

$$M_{n+1} = M_n^2$$



Check if determinant remains bounded

$$\mu(z) = x^2 - y^2$$

$$\mu(z) = XY = (x+y)(x-y)$$

Analytical solution $detM_n < \infty$

Difference between exact / finite precision computations

$$M = \begin{bmatrix} x & y \\ y & x \end{bmatrix}$$

Quadratic iteration $M_0 = M$

$$M_{n+1} = M_n^2$$

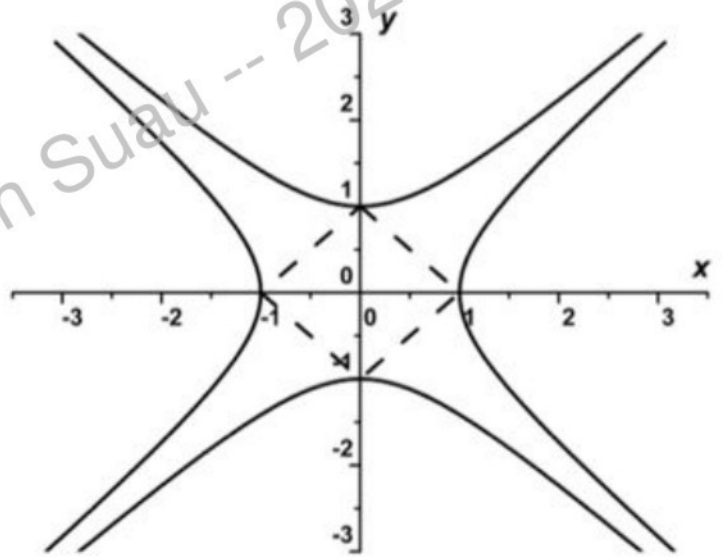
$$M_1 = M^2 =$$

$$\begin{bmatrix} x^2+y^2 & 2xy \\ 2xy & x^2+y^2 \end{bmatrix} \equiv \begin{bmatrix} (x+y)^2 & 0 \\ 0 & (x-y)^2 \end{bmatrix}$$

Check if determinant remains bounded

$$\mu(z) = x^2 - y^2$$

$$\mu(z) = XY = (x+y)(x-y)$$



Analytical solution $detM_n < \infty$

Difference between exact / finite precision computations

$$M = \begin{bmatrix} x & y \\ y & x \end{bmatrix}$$

Quadratic iteration $M_0 = M$

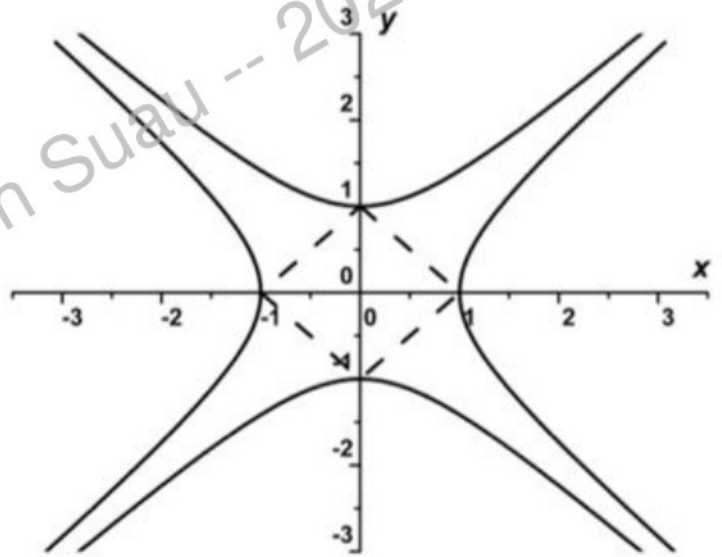
$$M_{n+1} = M_n^2$$

TWO DIFFERENT WAYS TO COMPUTE DETERMINANT

Check if determinant remains bounded

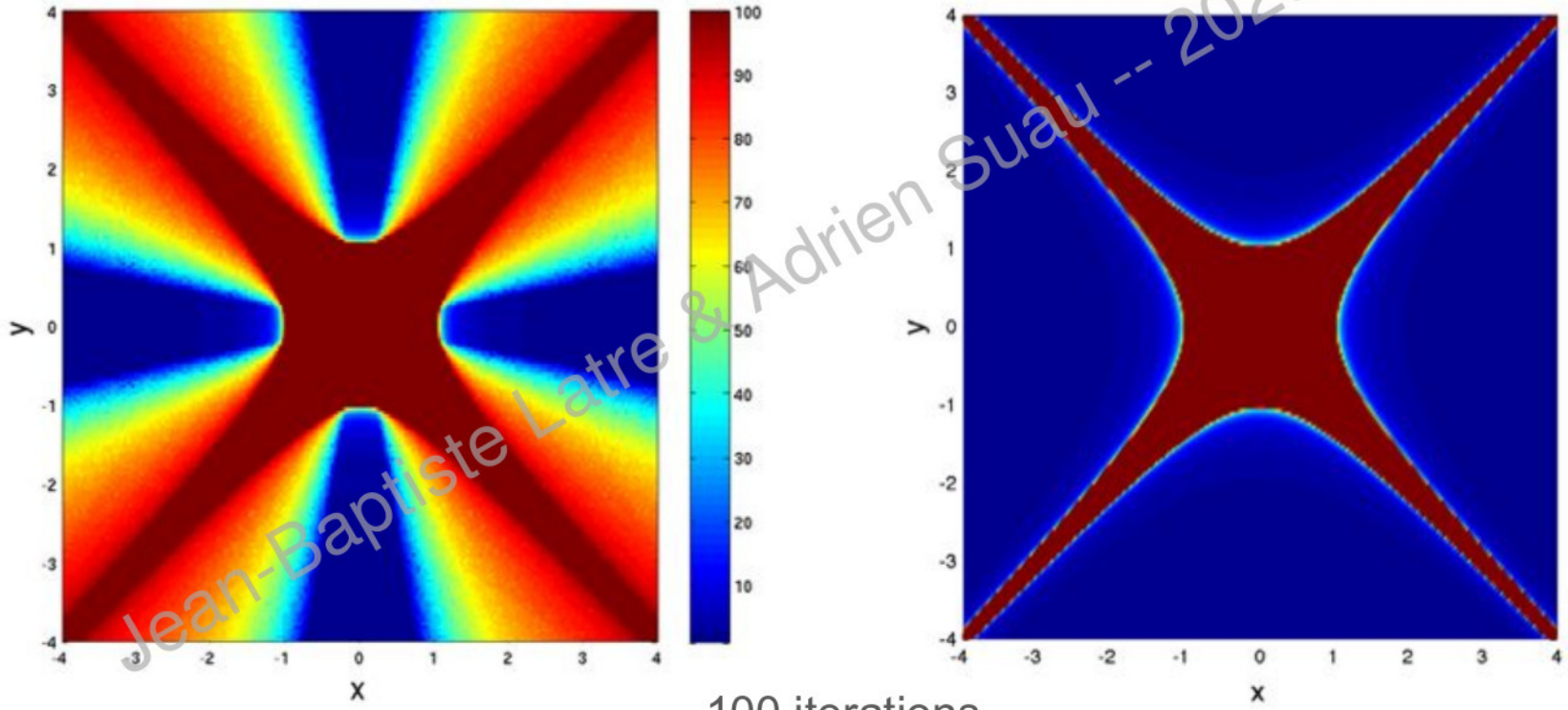
$$\mu(z) = x^2 - y^2$$

$$\mu(z) = XY = (x+y)(x-y)$$

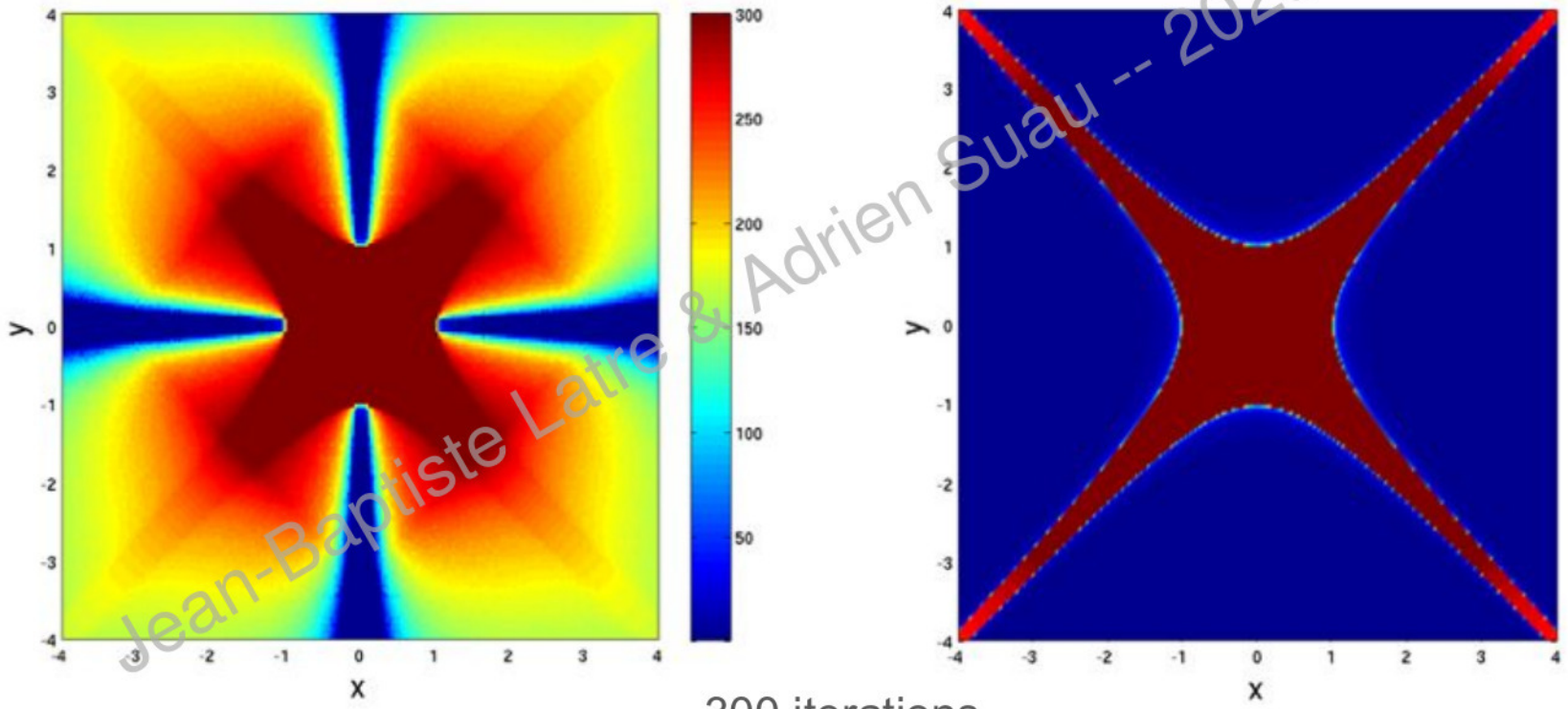


Analytical solution $det M_n < \infty$

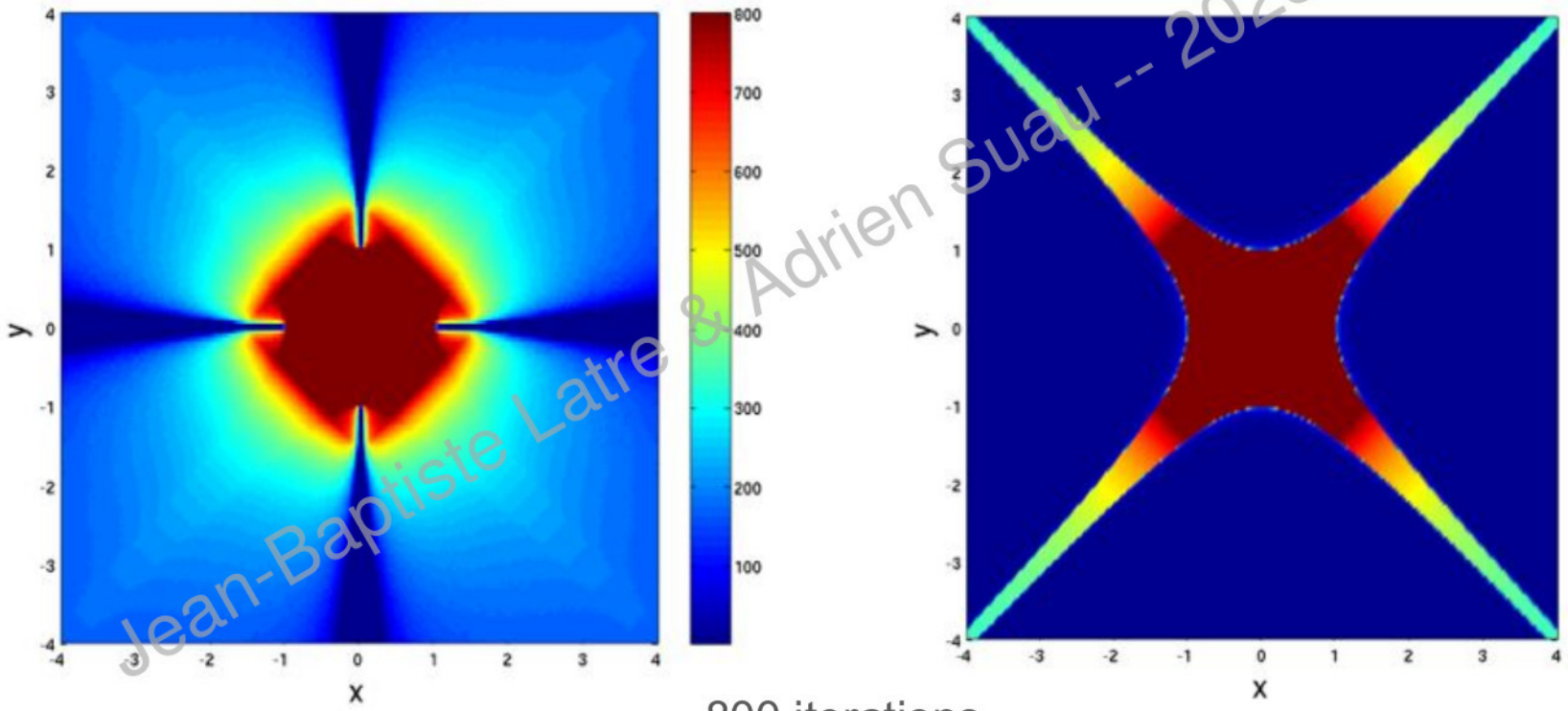
Colormap: number of iterations to trespass arbitrary threshold



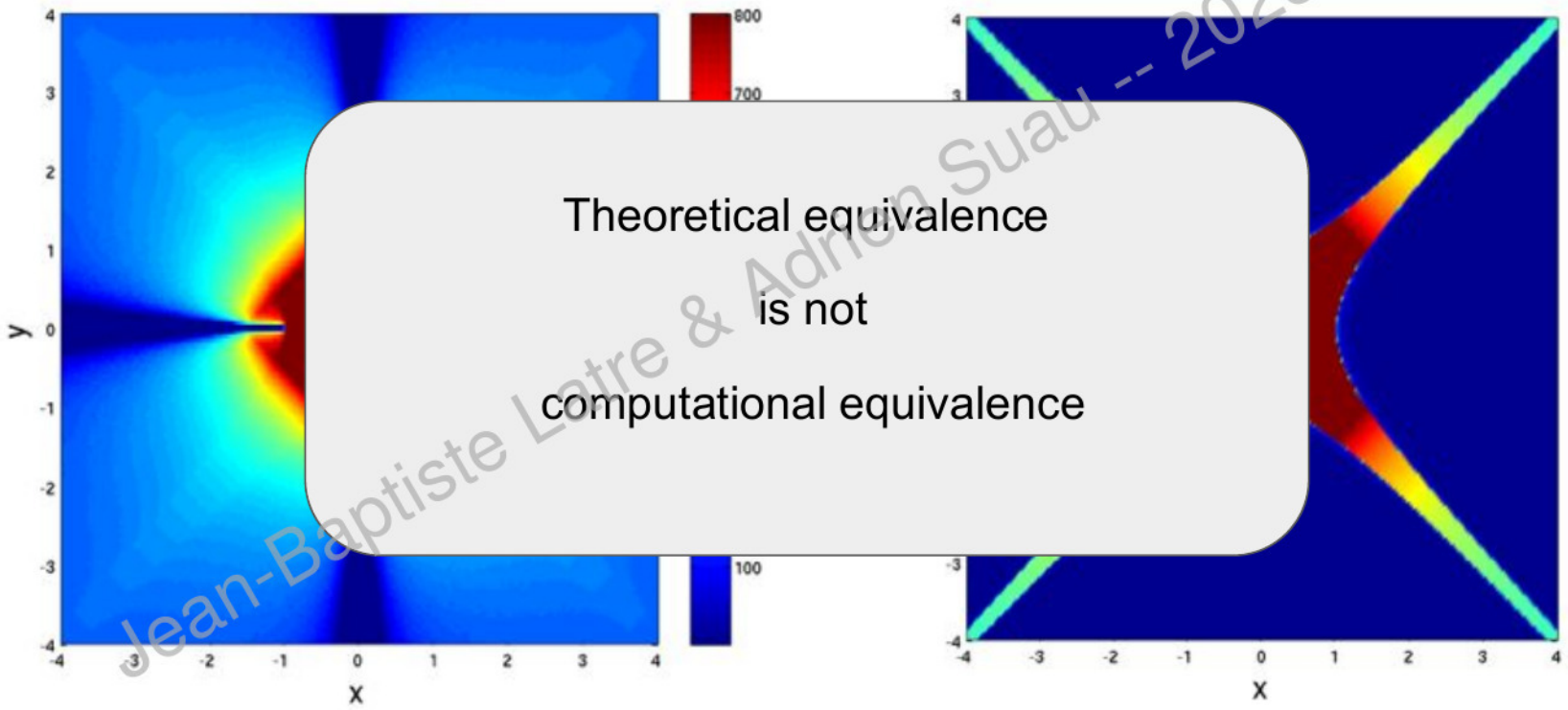
100 iterations



300 iterations



800 iterations

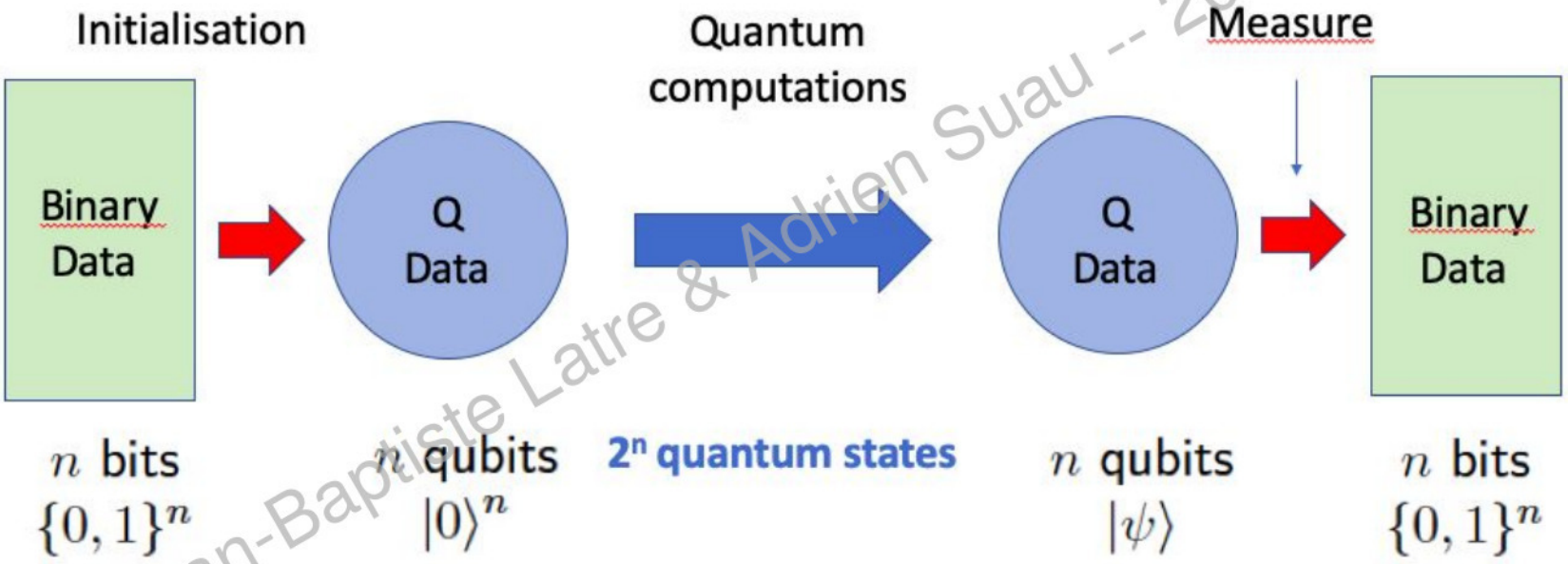


Theoretical equivalence

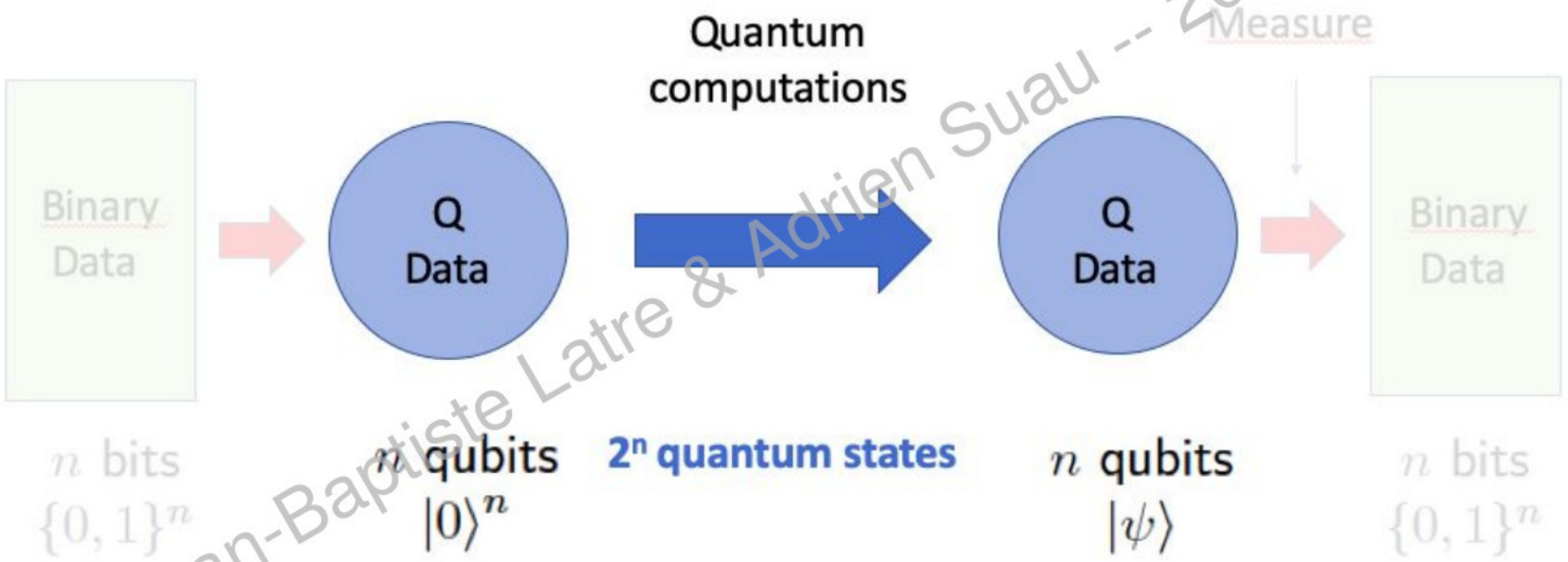
is not

computational equivalence

Global computation as noisy process



Range of QEC: only storage and transmission errors



Impact of algorithms and Quantum Error Correction (QEC)

NISQ is limited for practical purposes, mainly educational and onboarding

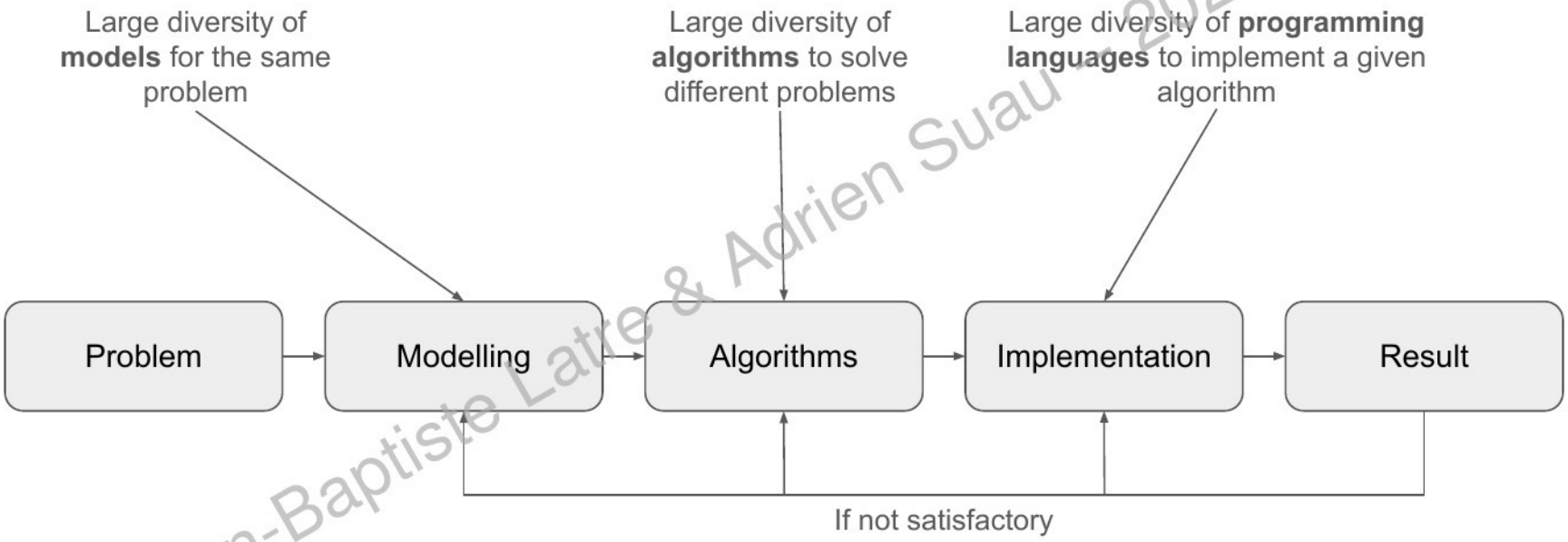
Cannot make any resource estimation on variational algorithms, only empirical

QEC may be the main cost for computation, decoding is classical (scaling ?)

Fault Tolerance (with initialisation and measure) does not exist yet

For now, algorithms should include **by design** numerical robustness

From modelling to implementation



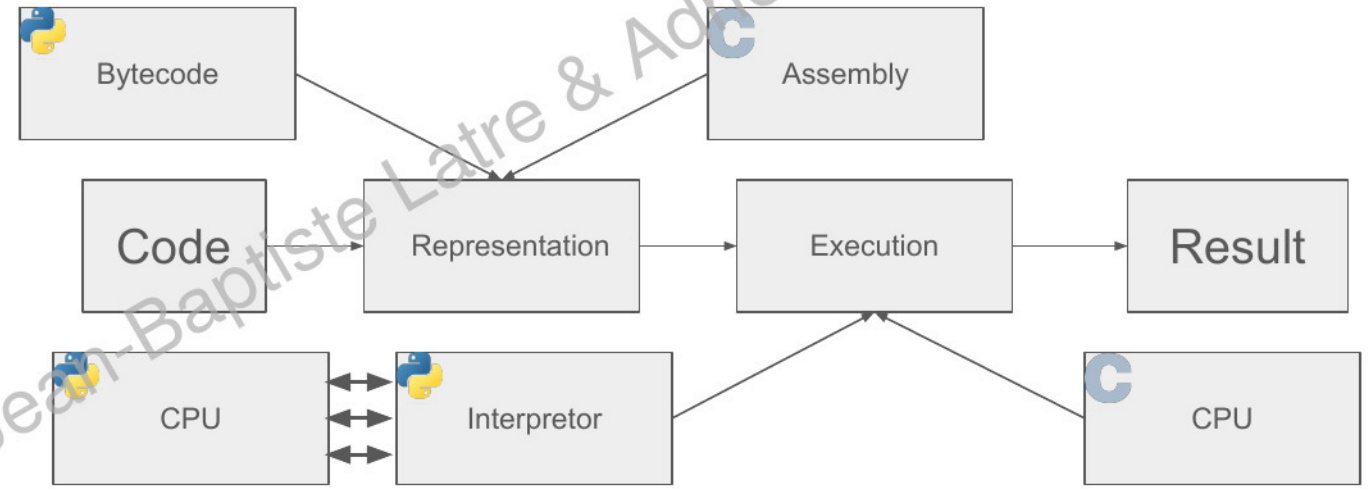
Programming languages

Python

- **First appearance**: 1991
- **Interpreted** language
- Garbage collected

C language

- **First appearance:** 1972
- **Compiled** language
- No garbage collection



Collatz (Syracuse) conjecture

Let

$$f(n) = \begin{cases} n/2 & \text{if } n \equiv 0 \pmod 2, \\ 3n+1 & \text{if } n \equiv 1 \pmod 2. \end{cases}$$

Then, for any integer n, iterating f eventually ends by 1.

Implemented using Python  and C  .

Difference #1: execution time

```
>> hyperfine --warmup 3 './collatz'
Benchmark 1: ./collatz
  Time  (mean ± σ):     287.2 ms ±   1.6 ms    [User: 286.0 ms, System: 1.0 ms]
  Range (min … max):   283.2 ms … 288.8 ms    10 runs

>> hyperfine --warmup 3 'python collatz.py'
Benchmark 1: python collatz.py
  Time  (mean ± σ):     5.479 s  ± 0.068 s    [User: 5.473 s, System: 0.003 s]
  Range (min … max):   5.411 s  … 5.617 s    10 runs
```

Python  is ~20 times slower than C  on **this specific example**

Difference #2: Energy consumption

	Energy (J)
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(i) Lua	45.98
(i) Jruby	46.54
(i) Ruby	69.91
(i) Python	75.88
(i) Perl	79.58

	Time (ms)
(c) C	1.00
(c) Rust	1.04
(c) C++	1.56
(c) Ada	1.85
(v) Java	1.89
(i) TypeScript	46.20
(i) Ruby	59.34
(i) Perl	65.79
(i) Python	71.90
(i) Lua	82.91

Pereira, R., et al. "Ranking Programming Languages by Energy Efficiency," in Science of Computer Programming, vol. 205, pp. 102609, 2021.

The impacts of abstractions

Low abstraction
Closer to physics
Harder to program
Efficient

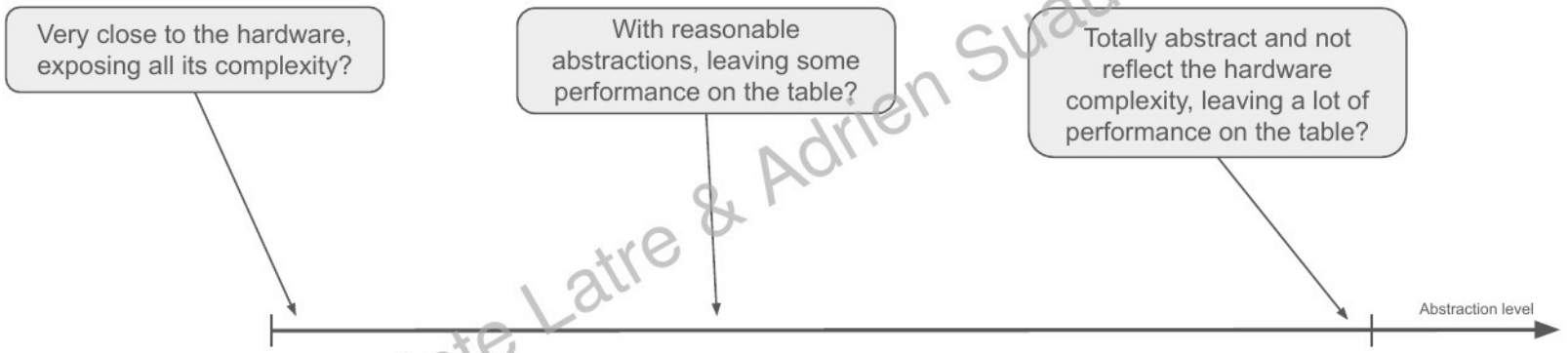
High abstraction
Far from physics
Easier to program
Less efficient

Implications for QEI: benchmark

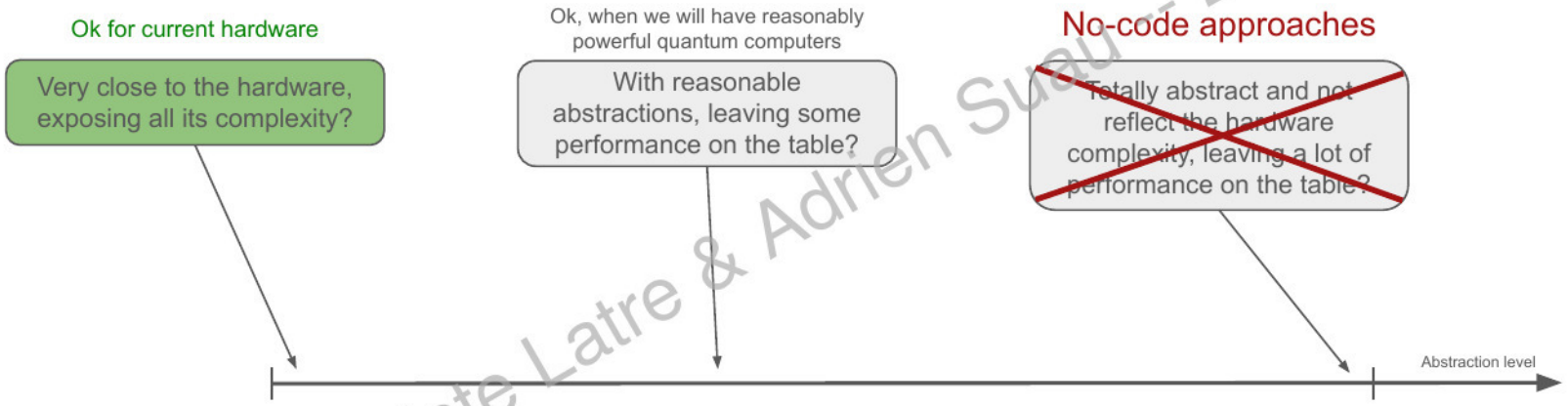
Benchmarks should be implemented with the less abstractions possible.

Implications for physicists: hardware design

When designing hardware, the programmer interface should be



Implications for physicists: personal opinion



Requirements for explicit implementation

Cannot rely on black box algorithms and problem dependency (e.g., Grover)

Balance theoretical speedup algorithm with global cost

- Data initialisation
- Cost of HPC / QPU interaction
- Input outputs HW/SW (algorithms with n to n connectivity)

Reasonable programming interface

Experience feedback Classical Computation

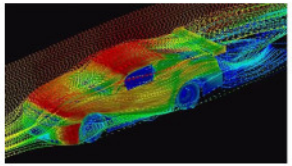
Low level is difficult but efficient (e.g. embedded systems in spatial design)

High level is a consequence of: decades of research (libraries)

large availability (waste) of resource

Is high level QC exact the opposite of QEI ?

- Computational Fluid Dynamics (wasted memory resources >80%)
- AI, works because of extreme usage of resource (brute force methods)



Conclusions

Both algorithms and software can impact the performance by several orders of magnitudes

Mathematical models: look for appropriate mathematical description of quantum physics (Better to be efficient from the beginning than paying the price latter)

Software solutions: stay away from abstractions and redefine basic routines at the lowest reasonable level.

Collaboration with other standardisation groups ?