Tools for modern scientific computing

Piotr Gawron

Nicolaus Copernicus Astronomical Center Polish Academy of Sciences
AGH Artificial Intelligence Center of Excellence

The 6th Young Astronomers Meeting at CAMK-PAN
Warsaw, 6 Mar 2024

Section 1

Computing in science

Introduction

Excerpt from scientific production code

```
if str(some_list)!="[]":
  # perform some computation
```

Golden maxim for today

When an engineer is wrong — the people suffer, when a scientist is wrong — the truth suffers.

Calculations in high energy physics

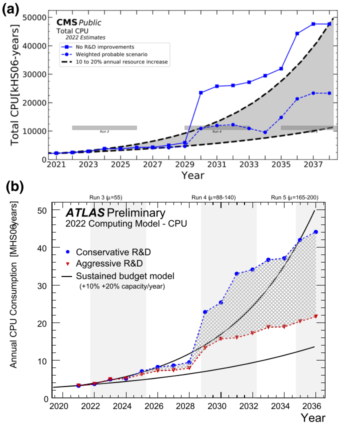
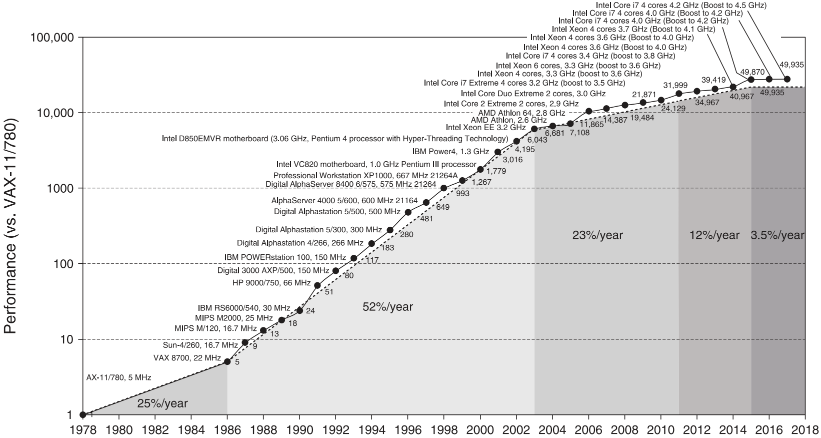
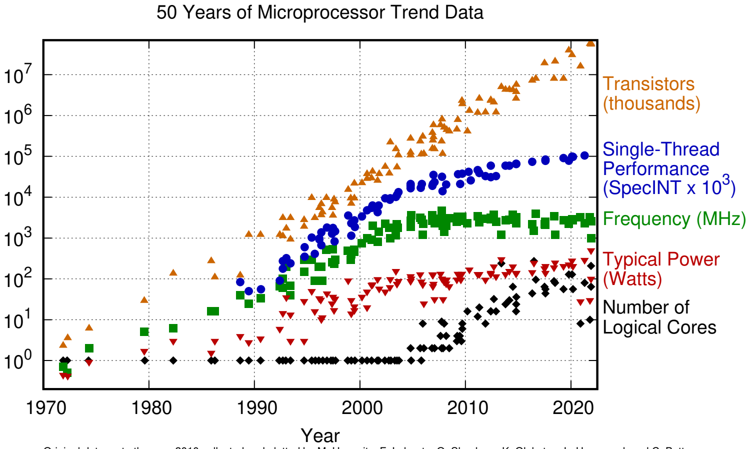


Fig. 1 Estimated CPU required by the CMS (top) and ATLAS (bottom) experiments for LHC and HL-LHC [6, 7]

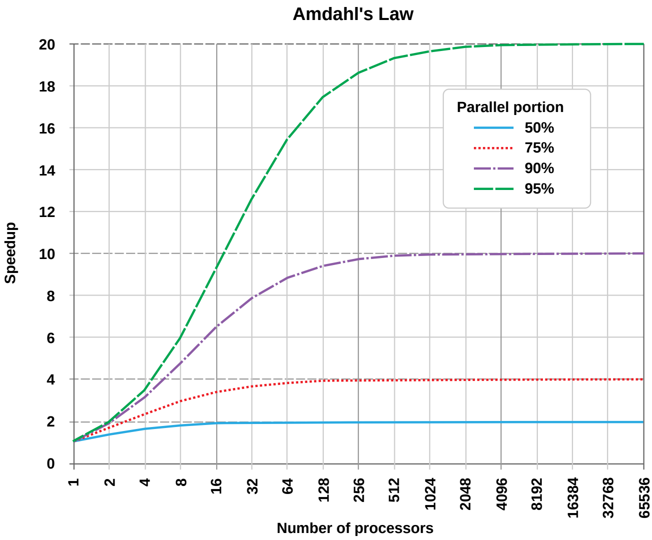
Moore's Law



Source: Hennessy, John L., and David A. Patterson. Computer architecture: a quantitative approach. Elsevier, 2011. 6th edition.



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp



Amdahl's Law
Parallel portion
50%
75%
90%
95%
Speedup
0
2
4
6
8
10
12
14
16
18
20
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
Number of processors

ML4Science

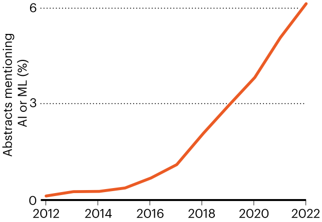
Artificial-intelligence models require the vast computing power of supercomputers, such as this one at the University of California, San Diego.

Garbage in, garbage out: mitigating risks and maximizing benefits of AI in research

Brooks Hanson, Shelley Stall, Joel Cutcher-Gershenfeld, Kristina Vrouwenvelder, Christopher Wirz, Yuhan (Douglas) Rao & Ge Peng

GROWING AI USE IN EARTH AND SPACE SCIENCE

A rising proportion of abstracts for the annual meeting of the American Geophysical Union mention artificial intelligence (AI) or machine learning (ML) — a trend seen across all areas of geoscience.



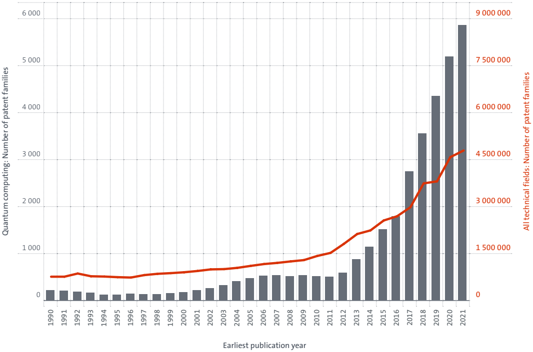
EUROPEAN SPACE AGENCY/COPERNICUS SENTINEL DATA (2017)/SPL

AI tools are being used to assess environmental observations, such as this satellite image of agricultural land in Bolivia that was once a forest.

Patents in quantum computing

Figure 2

Number of DOCDB patent families per earliest publication year in the field of quantum computing

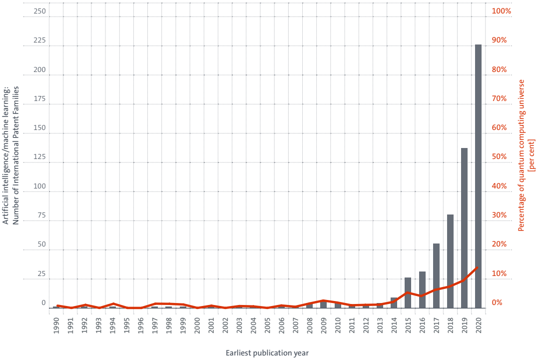


Source: authors' calculations

Quantum computing versus machine learning

Figure 16

Number of inventions per earliest publication year related to quantum computing and artificial intelligence/machine learning



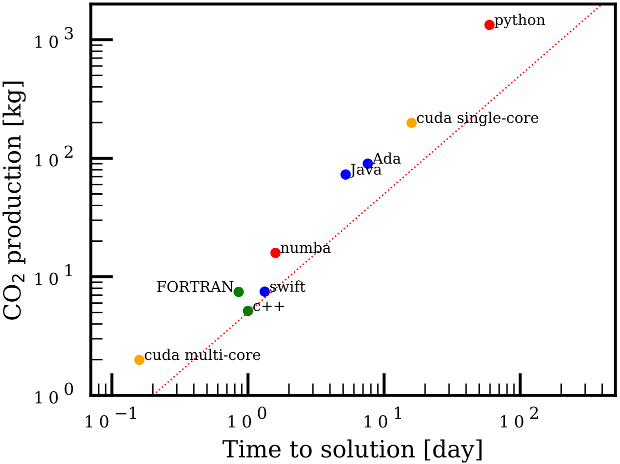
Source: authors' calculations

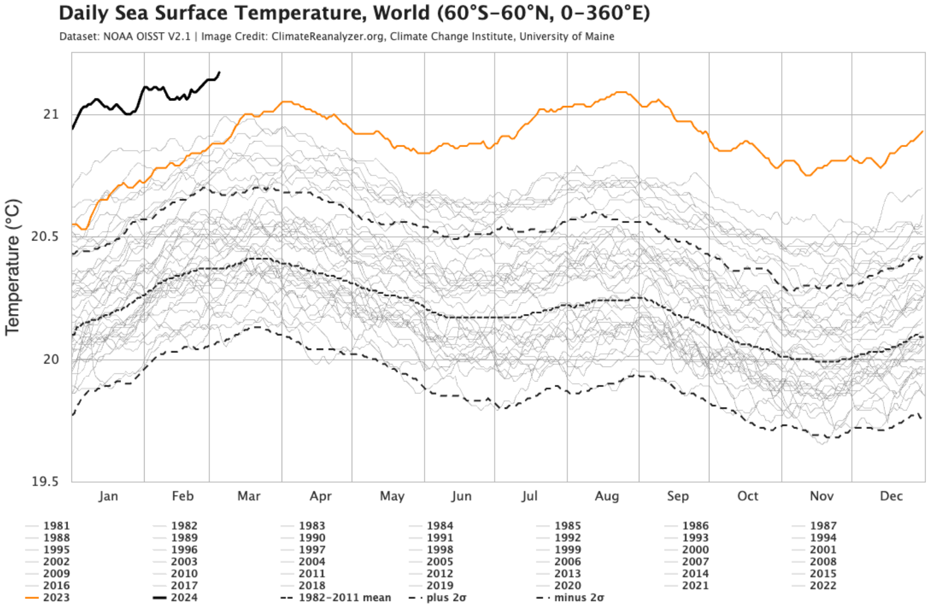
CO2 consumption by astrophysicists

The Ecological Impact of High-performance Computing in Astrophysics

Simon Portegies Zwart

[1] *Leiden Observatory, Leiden University, PO Box 9513, 2300 RA, Leiden, The Netherlands* [1]





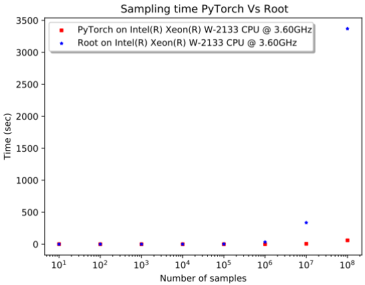
Daily Sea Surface Temperature, World (60°S–60°N, 0–360°E)
Dataset: NOAA OISST V2.1 | Image Credit: ClimateReanalyzer.org, Climate Change Institute, University of Maine
Temperature (°C)
21
20.5
20
19.5
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
1982–2011 mean
plus 2σ
minus 2σ

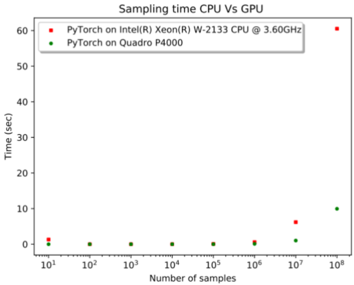
Are there slow and fast programming languages

- Yes: slow — Python, C++; fast — Python, C++
- No: there are slow and fast computer systems.

ROOT
Data Analysis Framework

PyTorch



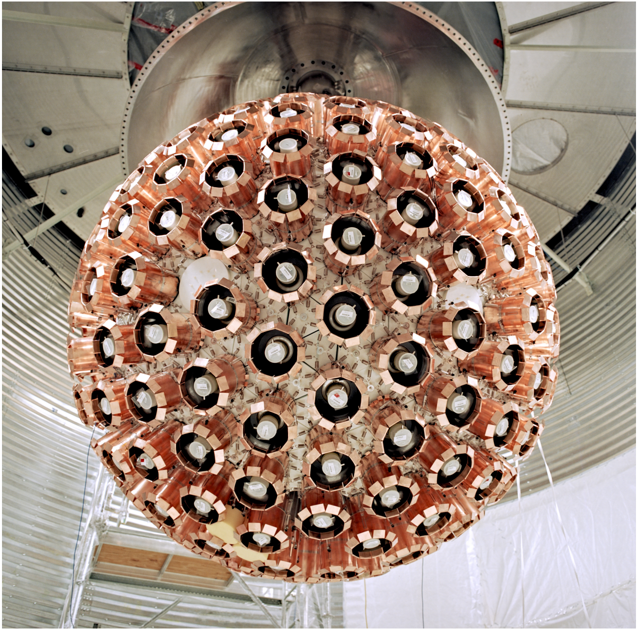


Scientific software engineering matters

Section 2

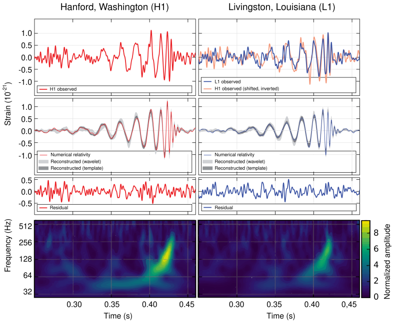
What do we do?

Dark matter detectorDEAP-3600



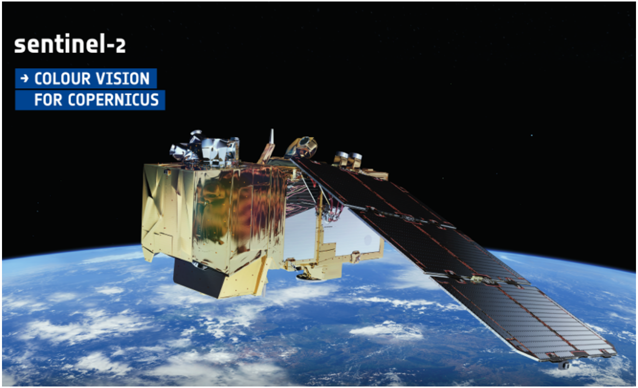
Gravitational waves

Virgo detector



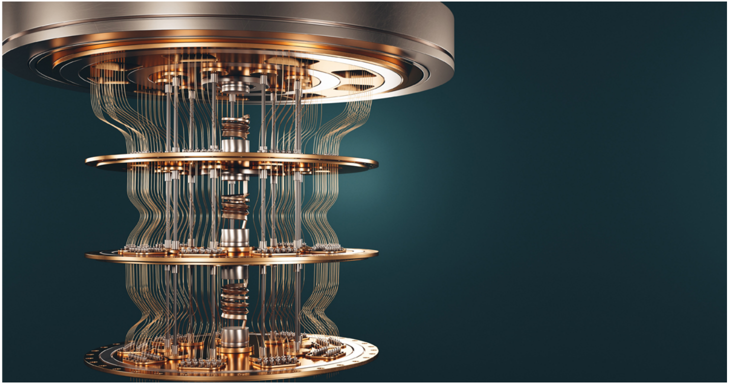
Measurement of first black hole coalescence with LIGO

Satellite imagery



Sentinel-2

Quantum computing



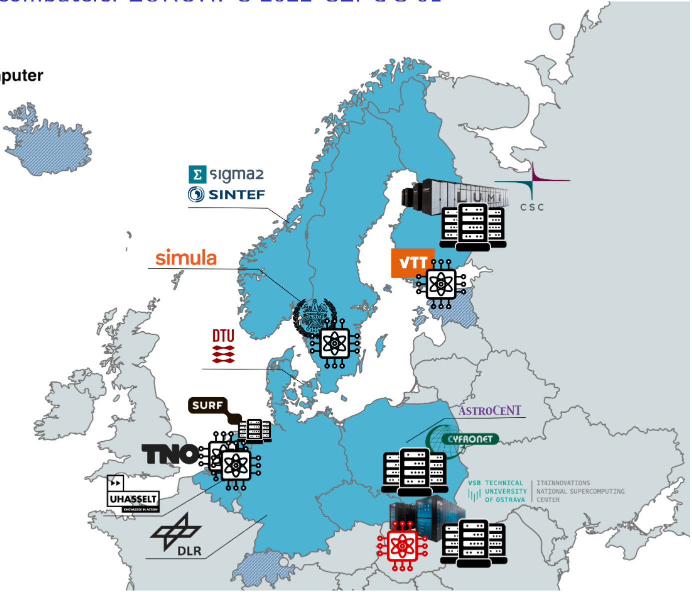
Section 3

Quantum computing

LUMI-Q

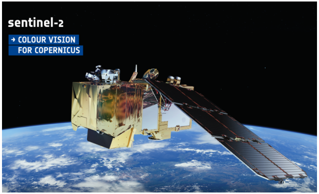
European quantum computers. EUROHPC-2022-CEI-QC-01

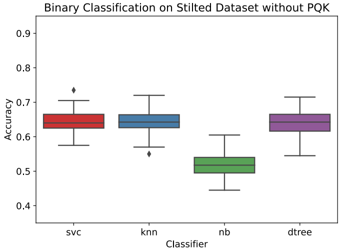
- LUMI-Q consortium
- LUMI consortium
- LUMI-Q quantum computer
- quantum computer
- supercomputer

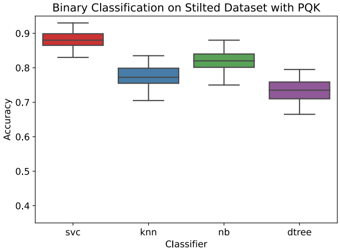


Spectral information processing with quantum neural networks

Manish Gupta, Piotr Gawron, Co-operation ESA's Φ-Lab — AstroCeNT







Unsupervised quantum machine learning for Earth observations

Piotr Gawron with IITiS PAN, CSGroup and CNES

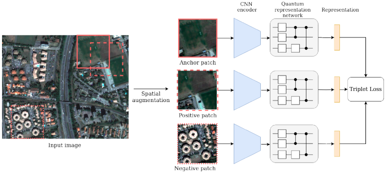
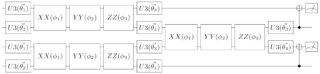
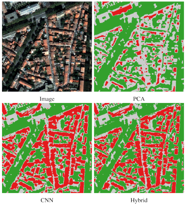


Fig. 1. Illustration of the proposed hybrid contrastive learning framework.





Section 4

Julia

Julia

Julia is

- fast, compiled on the fly,
- high-level,
- expressive
- programming language
- designed for scientific computing.

Julia

Example of an optimisation problem

$$
\begin{aligned}
\min \quad & 12x + 20y \\
\text{s.t.} \quad & 6x + 8y \geq 100 \\
& 7x + 12y \geq 120 \\
& x \geq 0 \\
& y \in [0, 3]
\end{aligned}
$$

Julia

Example of an optimisation problem

```julia
using JuMP
```

```julia
using HiGHS
```

```
model = A JuMP Model
        Feasibility problem with:
        Variables: 0
        Model mode: AUTOMATIC
        CachingOptimizer state: EMPTY_OPTIMIZER
        Solver name: HiGHS
```

```julia
model = Model(HiGHS.Optimizer)
```

$$x$$

```julia
@variable(model, x >= 0)
```

$$y$$

```julia
@variable(model, 0 <= y <= 3)
```

$$12x + 20y$$

```julia
@objective(model, Min, 12x + 20y)
```

$$6x + 8y \geq 100$$

Julia

Example of an optimisation problem

$$7x + 12y \geq 120$$

```
@constraint(model, c2, 7x + 12y >= 120)
```

```
print(model)
```

```
Min 12 x + 20 y
Subject to
 c1 : 6 x + 8 y ≥ 100
 c2 : 7 x + 12 y ≥ 120
 x ≥ 0
 y ≥ 0
 y ≤ 3
```

```
optimize!(model)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
2 rows, 2 cols, 4 nonzeros
2 rows, 2 cols, 4 nonzeros
Presolve : Reductions: rows 2(-0); columns 2(-0); elements 4(-0) - Not reduced
Problem not reduced by presolve: solving the LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 2(220) 0s
          2     2.0500000000e+02 Pr: 0(0) 0s
Model   status      : Optimal
Simplex   iterations: 2
Objective value     :  2.0500000000e+02
HiGHS run time      :          0.00
```

```
Enter cell code...
```

Julia

Lorenz system

```
using DifferentialEquations
```

```
using Plots
```

parameterized_lorenz! (generic function with 1 method)

```
function parameterized_lorenz!(du, u, p, t)
    x, y, z = u
    σ, ρ, β = p
    du[1] = dx = σ * (y - x)
    du[2] = dy = x * (ρ - z) - y
    du[3] = dz = x * y - β * z
end
```

u0 = [1.0, 0.0, 0.0]

```
u0 = [1.0, 0.0, 0.0]
```

tspan = (0.0, 100.0)

```
tspan = (0.0, 100.0)
```

p = [10.0, 28.0, 2.66667]

```
p = [10.0, 28.0, 8 / 3]
```

```
prob =
ODEProblem with uType Vector{Float64} and tType Float64
timespan: (0.0, 100.0)
u0: 3-element Vector{Float64}:
 1.0
 0.0
 0.0
```

```
prob = ODEProblem(parameterized_lorenz!, u0,
tspan, p)
```

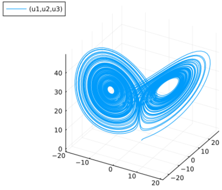
Julia

Lorenz system

```
sol =
```

	timestamp	value1	value2	value3
1	0.0	1.0	0.0	0.0
2	3.56786e-5	0.999643	0.000998805	1.78143e-8
3	0.000392465	0.996105	0.0109654	2.14696e-6
4	0.00326241	0.969359	0.0897706	0.000143802
5	0.00905808	0.924204	0.242289	0.00104616
6	0.0169565	0.880046	0.438736	0.00342426
7	0.02769	0.848331	0.691563	0.00848762
8	0.0418564	0.849504	1.01454	0.0182121
9	0.0602404	0.913907	1.44256	0.0366938
10	0.0836854	1.08886	2.05233	0.0740257
	more			

```
sol = solve(prob)
```




```
plot(sol, idxs = (1, 2, 3))
```

Calculations in high energy physics

Computing and Software for Big Science (2023) 7:10
https://doi.org/10.1007/s41781-023-00104-x

RESEARCH

Check for updates

Potential of the Julia Programming Language for High Energy Physics Computing

Jonas Eschle[1] · Tamás Gál[2] · Mosè Giordano[3] · Philippe Gras[4] · Benedikt Hegner[5] · Lukas Heinrich[6] · Uwe Hernandez Acosta[7,8] · Stefan Kluth[6] · Jerry Ling[9] · Pere Mato[5] · Mikhail Mikhasenko[10,11] · Alexander Moreno Briceño[12] · Jim Pivarski[13] · Konstantinos Samaras-Tsakiris[5] · Oliver Schulz[6] · Graeme Andrew Stewart[5] · Jan Strube[14,15] · Vassil Vassilev[13]

Received: 6 June 2023 / Accepted: 11 August 2023
© The Author(s) 2023

Calculations in high energy physics

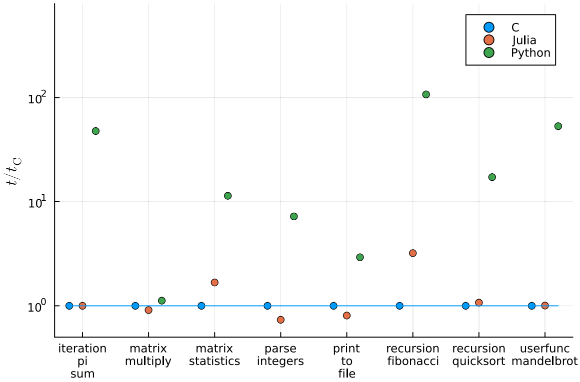
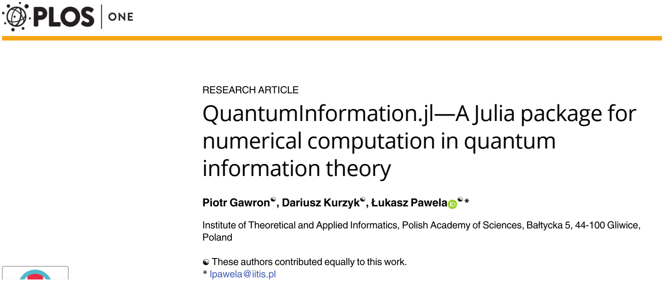


Fig. 3 Comparison of C/C++, Python and Julia language performance for a set of short algorithms. OpenBLAS, together with `NumPy` in the Python case are used for matrix operation. The score is defined as the time to run the algorithm divided by the time to run the C version of the same algorithm

Julia

Quantum information


PLOS | ONE

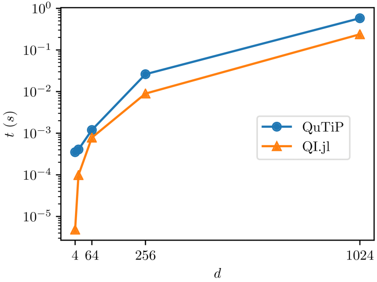
RESEARCH ARTICLE

QuantumInformation.jl—A Julia package for numerical computation in quantum information theory

Piotr Gawron[☯], **Dariusz Kurzyk**[☯], **Łukasz Pawela**[☯]*

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

☯ These authors contributed equally to this work.
* lpawela@iitis.pl



Julia

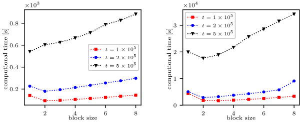
Calculation of cumulants

SIAM J. Sci. Comput.
Vol. 40, No. 3, pp. A1590–A1610

© 2018 Society for Industrial and Applied Mathematics

EFFICIENT COMPUTATION OF HIGHER-ORDER CUMULANT TENSORS*

KRZYSZTOF DOMINO†, PIOTR GAWRON†, AND ŁUKASZ PAWELA†



(a) 4th cumulant tensor

(b) 5th cumulant tensor

Fig. 2. *Computation time for cumulant tensors computed using the block structure and the proposed algorithm for different block sizes b, at $n = 60$.*

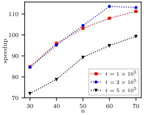
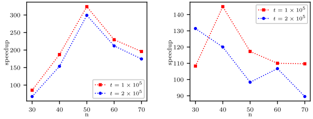
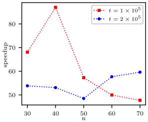


Fig. 3. *Computation time speedup of fourth cumulant tensor computed using the block structure and the proposed algorithm vs. the naive algorithm.*



(a) Julia implementation of general algorithm from [16].

(b) R implementation of general algorithm



(c) Specialized algorithm from [16] implemented in R.

Fig. 6. *Computation time speedup of fourth cumulant tensor calculation using algorithm employing the block structure vs. algorithms from* [16].

PhD opportunities

- Quantum computing for astronomy / astrophysics
- Neuromorphic computing
- Scientific software tools in Julia — studying new computation methods
- Large scale computation workflows

Thank you

CAMK PAN

CEAI

Center of Excellence in Artificial Intelligence

https://piotrgawron.eu

gawron@camk.edu.pl