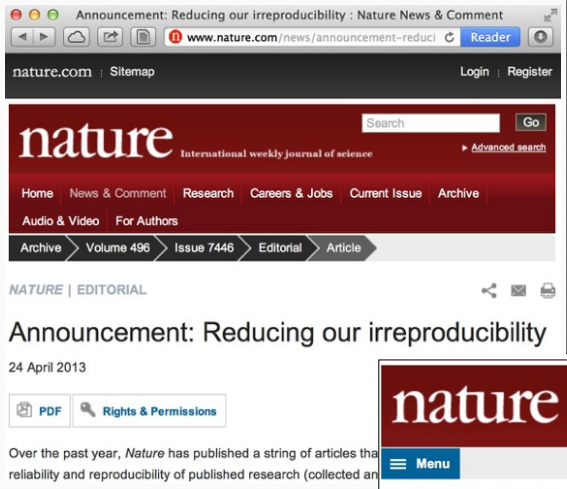
Reproducibility Crisis, Open Science, and Computer Science

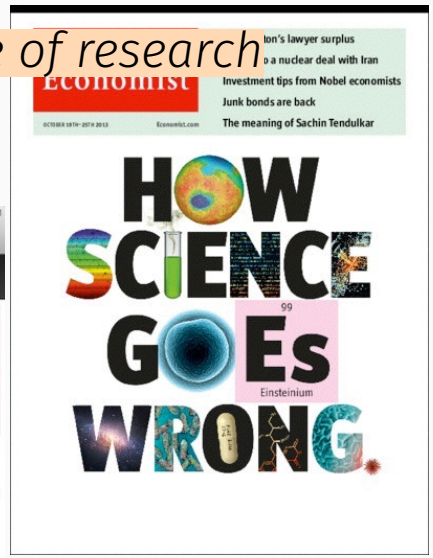
Arnaud Legrand

December 2020

Public Evidence for a Lack of Reproducibility

- J.P. Ioannidis. *Why Most Published Research Findings Are False* PLoS Med. 2005.
- *Lies, Damned Lies, and Medical Science*, The Atlantic. Nov, 2010
- *Reproducibility: A tragedy of errors*, Nature, Feb 2016.
- Steen RG, *Retractions in the scientific literature: is the incidence of research fraud increasing?*, J. Med. Ethics 37, 2011





Courtesy V. Stodden, SC, 2015

Newsworthy Stories about Scientific Misconduct

Dong-Pyou Han Assistant professor, Biomedical sciences, Iowa State University, 2013

Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity

- Han and his team received ≈ $19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials He was also fined US $7.2 million!

Newsworthy Stories about Scientific Misconduct

Dong-Pyou Han Assistant professor, Biomedical sciences, Iowa State University, 2013

Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity

- Han and his team received ≈ $19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials He was also fined US $7.2 million!

Dieterik Stapel Professor, Social Psychology, Univ. Amsterdam, 2011

I failed as a scientist. I adapted research data and fabricated research. Not once, but several times, not for a short period, but over a longer period of time. [..] I am aware of the suffering and sorrow that I caused to my colleagues... I did not withstand the pressure to score, to publish, the pressure to get better in time. I wanted too much, too fast. In a system where there are few checks and balances, where people work alone, I took the wrong turn.

58 retracted publications

Newsworthy Stories about Scientific Misconduct

Dong-Pyou Han Assistant professor, Biomedical sciences, Iowa State University, 2013

Falsified blood results to make it appear as though a vaccine exhibited anti-HIV activity

- Han and his team received ≈ $19 million from NIH
- 1 retracted publication and resignation of university. Sentenced in 2015 to 57 months imprisonment for fabricating and falsifying data in HIV vaccine trials He was also fined US $7.2 million!

Dieterik Stapel Professor, Social Psychology, Univ. Amsterdam, 2011

I failed as a scientist. I adapted research data and fabricated research. Not once, but several times, not for a short period, but over a longer period of time. [..] I am aware of the suffering and sorrow that I caused to my colleagues... I did not withstand the pressure to score, to publish, the pressure to get better in time. I wanted too much, too fast. In a system where there are few checks and balances, where people work alone, I took the wrong turn.

58 retracted publications

Brian Wansink Professor, Psychological Nutrition, Cornell, 2016

I gave her a data set of a self-funded, failed study which had null results. I said "This cost us a lot of time and our own money to collect. There's got to be something here we can salvage because it's a cool (rich & unique) data set." I told her what the analyses should be. [..] Every day she came back with puzzling new results, and every day we would scratch our heads, ask "Why," and come up with another way to reanalyze the data with yet another set of plausible hypotheses

17 retracted publications

Scientific Misconduct? What are the consequences ?

Bad science is **deleterious**

- It is used to backup stupid politics, it affects people's life, ...
- It blurs the frontier between scientists and crooks

Media attention inflates **conspiracy** opinions ☹

- *Scientific result are worthless.*
- *Scientists cannot even agree with each others on economy/climate/vaccine/5G/...*
- *Stop the scientific dictatorship/lobby!*

A Credibility Crisis?

Scientific misconduct is obviously wrong but it's not new!

- Every domain has its black sheep
- The publish or perish pressure is a huge pain

The Battle against Scientific Fraud

CNRS International Magazine




A Credibility Crisis?

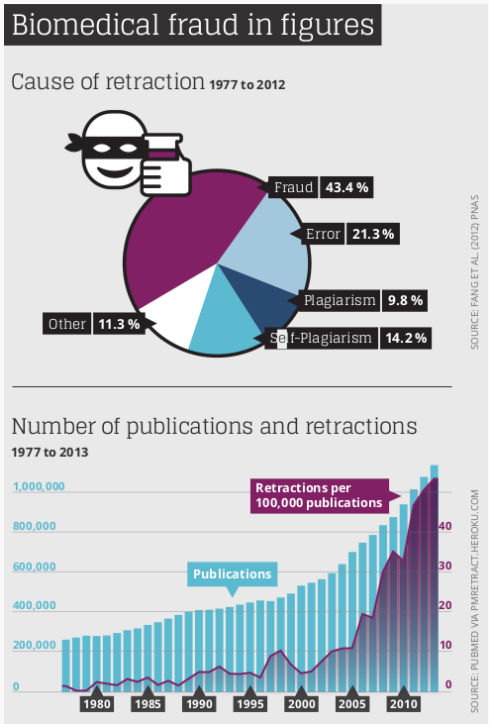
Scientific misconduct is obviously wrong but it's not new!

- Every domain has its black sheep
- The publish or perish pressure is a huge pain

The Battle against Scientific Fraud

CNRS International Magazine

Fraud is the (uninteresting) visible part of the iceberg

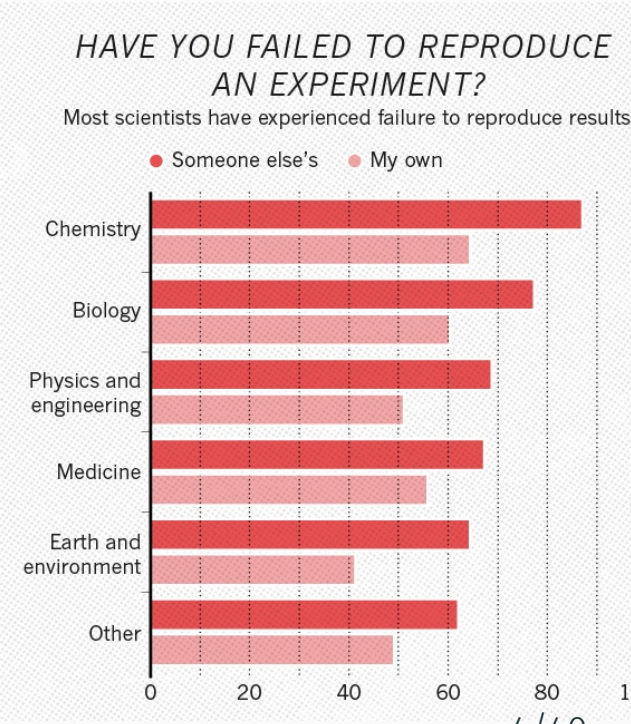


- Failing to reproduce the results of others is commor

  1,500 scientists lift the lid on reproc

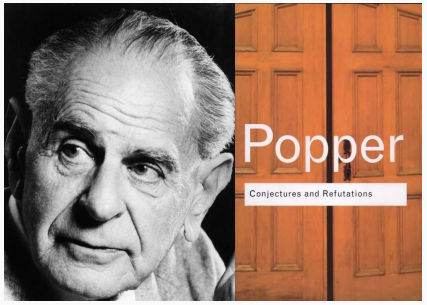
  Nature, May 2016

- How so? Why now? Why is this important? What can we do about it?



Reproducibility of experimental results: the *hallmark of science*

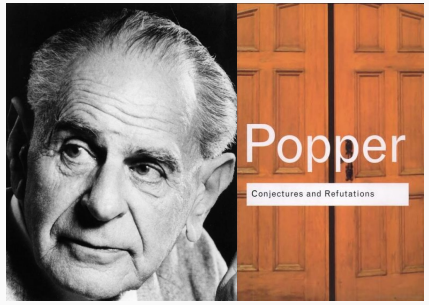
1934: **Karl Popper** puts the notions of falsifiability and crucial experiment as the hallmark of science



- If no experiment can be set up to disprove your theory, it is not science
- Good experiments discriminate good theories from bad ones
- Non-reproducible single occurrences are of no significance to science

Reproducibility of experimental results: the *hallmark of science*

1934: **Karl Popper** puts the notions of falsifiability and crucial experiment as the hallmark of science



- If no experiment can be set up to disprove your theory, it is not science
- Good experiments discriminate good theories from bad ones
- Non-reproducible single occurrences are of no significance to science

An ideal rather than the norm

Popper's proposal works well for Physics from the 18th century but is not so simple for many other domains:

- Theory of evolution
- Spotting a SuperNova
- Particle Physics (a single LHC)
- Biology (every animal does not behave in the same way)
- Anthropology (impact on people from a remote culture)

Reproducibility: a *core value of science*

1. Universality: Science aims for objective findings, accessible to anyone

   Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

   Methods, biases, ... How to discriminate sound ~~theories~~ experiments from bad ones? 😉

   Reproducibility acts as a Quality control

Reproducibility: a *core value of science*

1. Universality: Science aims for objective findings, accessible to anyone

   Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

   Methods, biases, ... How to discriminate sound ~~theories~~ experiments from bad ones? 😉

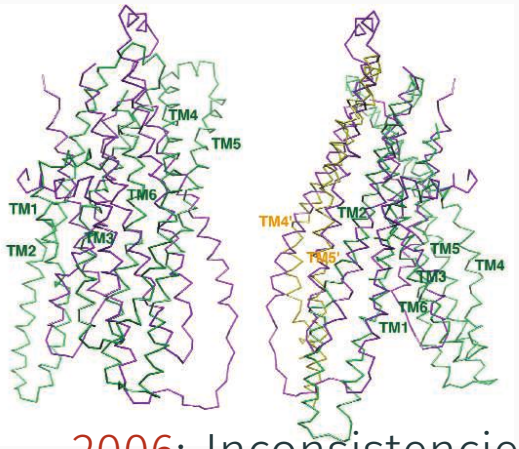
   Reproducibility acts as a Quality control

But, scientific practices have greatly evolved, in particular since we rely on computers

How computers broke science – and what we can do about it

– Ben Marwick, The conversation, 2015

How computers broke science



Geoffrey Chang (Scripps, UCSD) works on crystalography and studies the structure of cell membrane proteins.

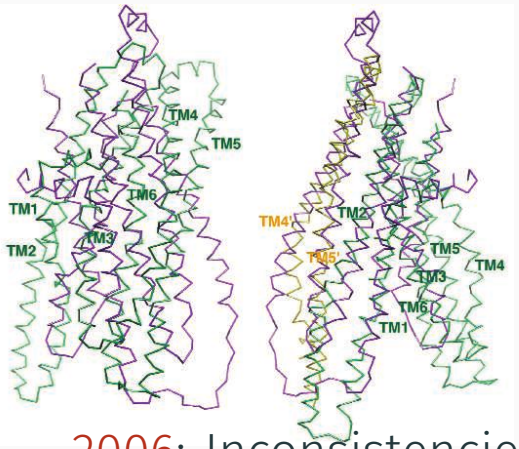
He specialized in structures of multidrug resistant transporter proteins in bacteria: MsbA de Escheria Choli (Science, 2001), Vibrio cholera (Mol. Biology, 2003), Salmonella typhimurium (Science, 2005)

2006: Inconsistencies reveal a programming mistake

> *A homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.*

5 retractations that motivate improved software engineering practices in comp. biology

How computers broke science



Geoffrey Chang (Scripps, UCSD) works on crystalography and studies the structure of cell membrane proteins.

He specialized in structures of multidrug resistant transporter proteins in bacteria: MsbA de Escheria Choli (Science, 2001), Vibrio cholera (Mol. Biology, 2003), Salmonella typhimurium (Science, 2005)

2006: Inconsistencies reveal a programming mistake

> *A homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.*

5 retractations that motivate improved software engineering practices in comp. biology

There is worse!

- The generalized and intensive use of spreadsheets (COVID tracing)
- Relying on black box statistical methods is infinitely easier than understanding them (Learning and Data Analytics frameworks are nuclear weapons)
- Numerical errors and software environment unawareness

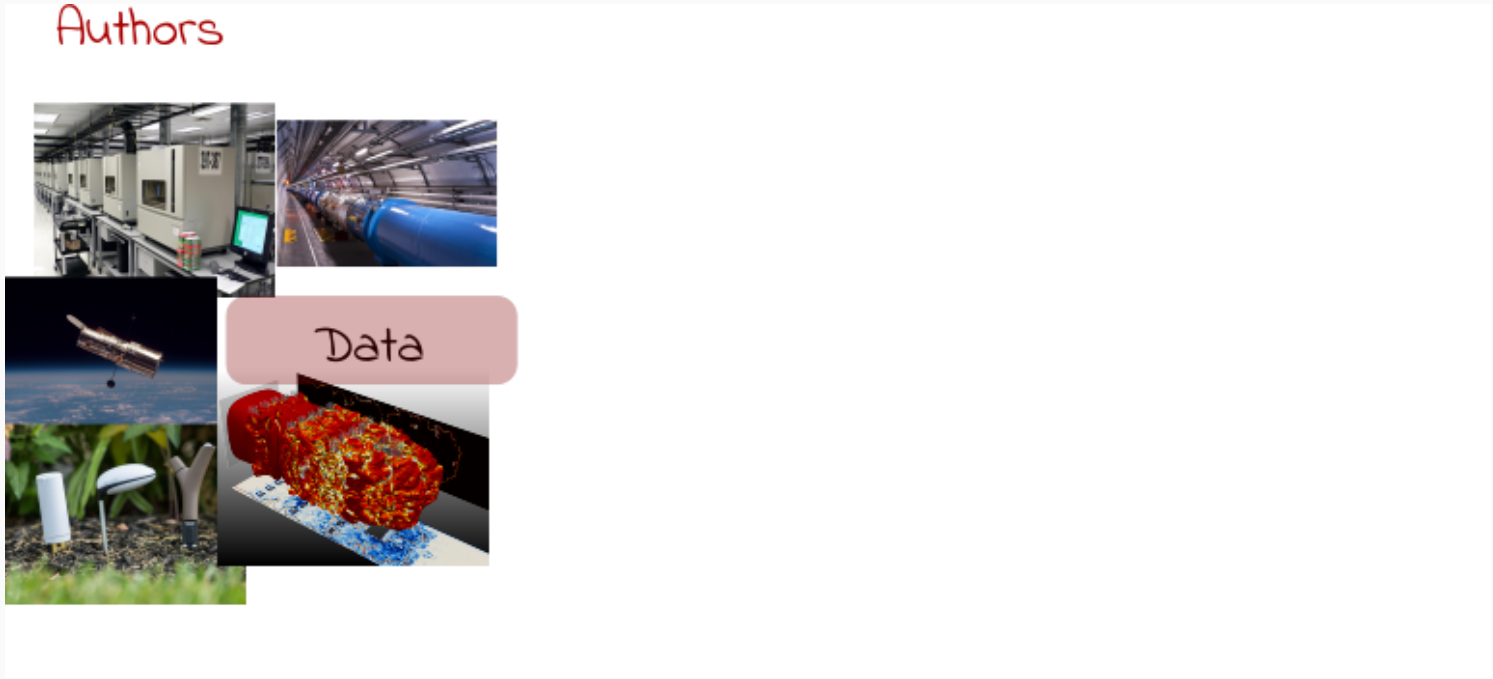
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



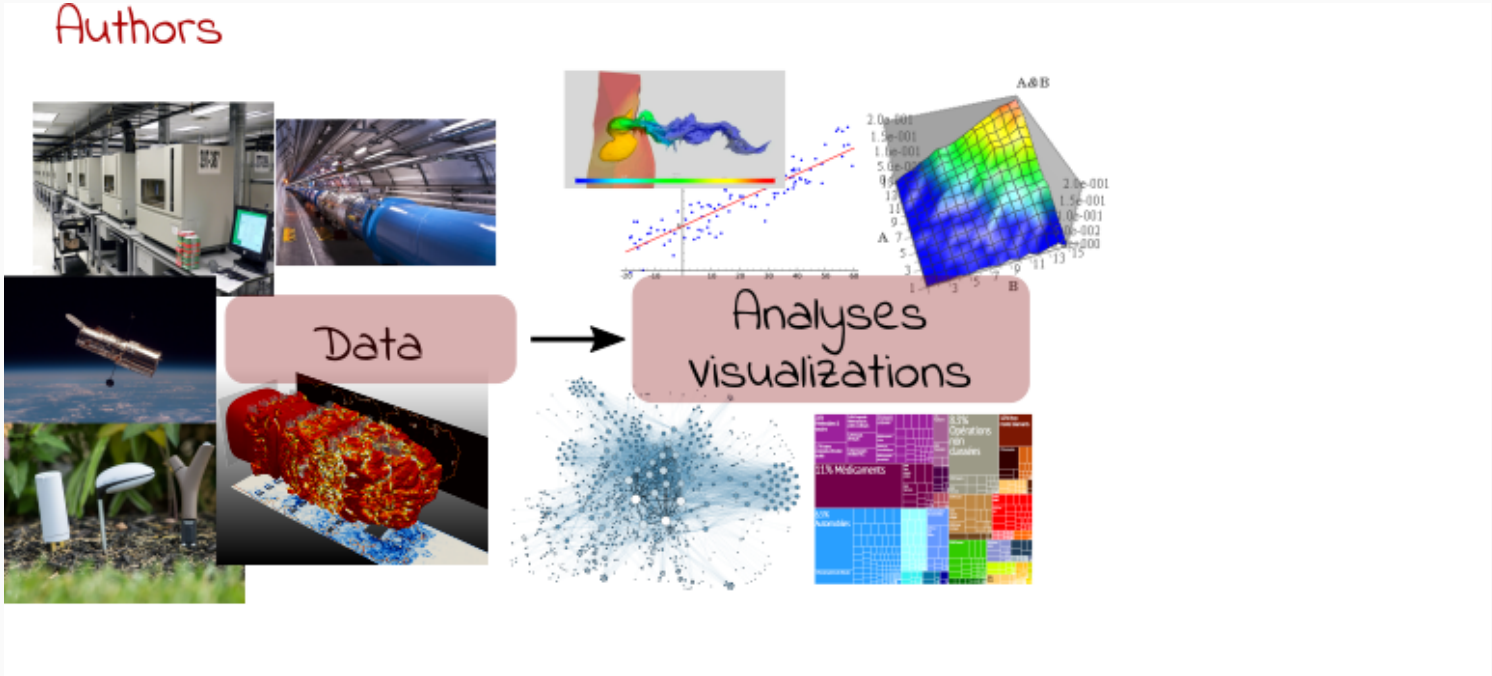
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



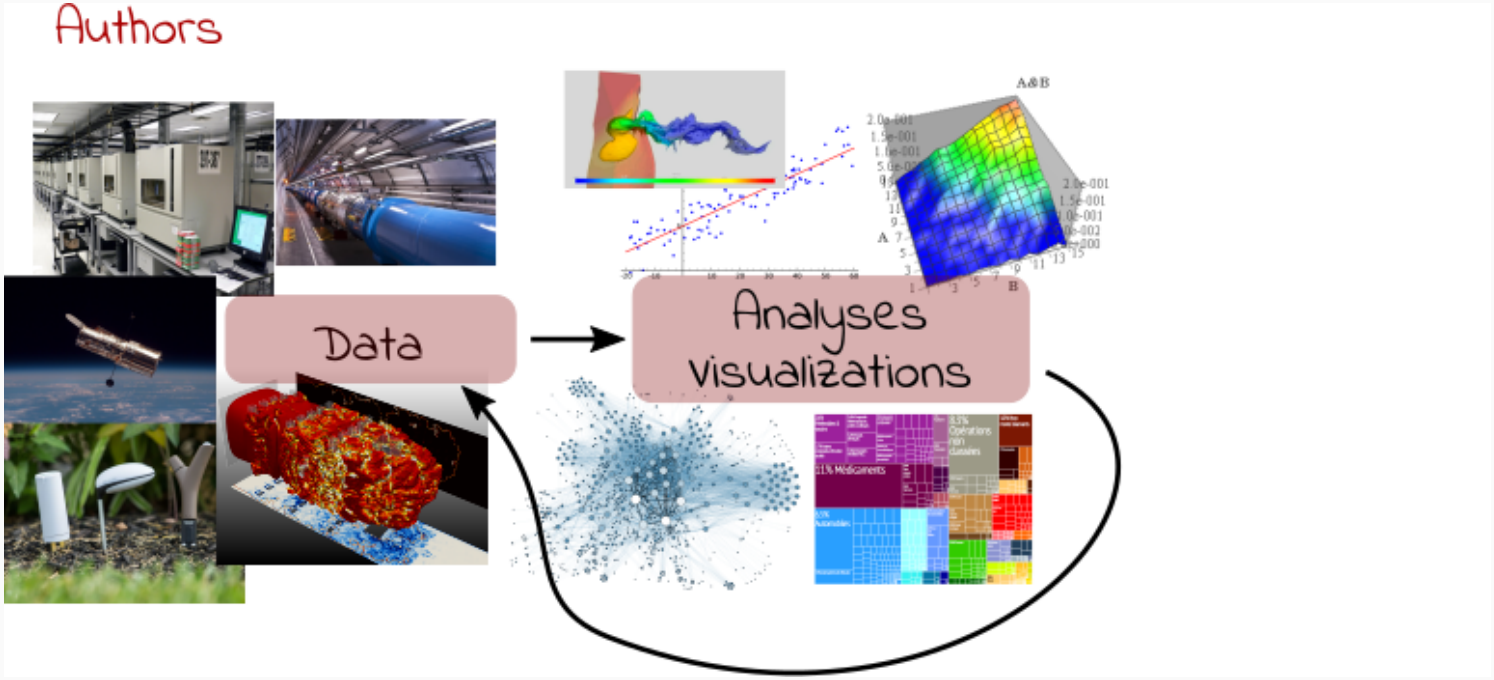
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



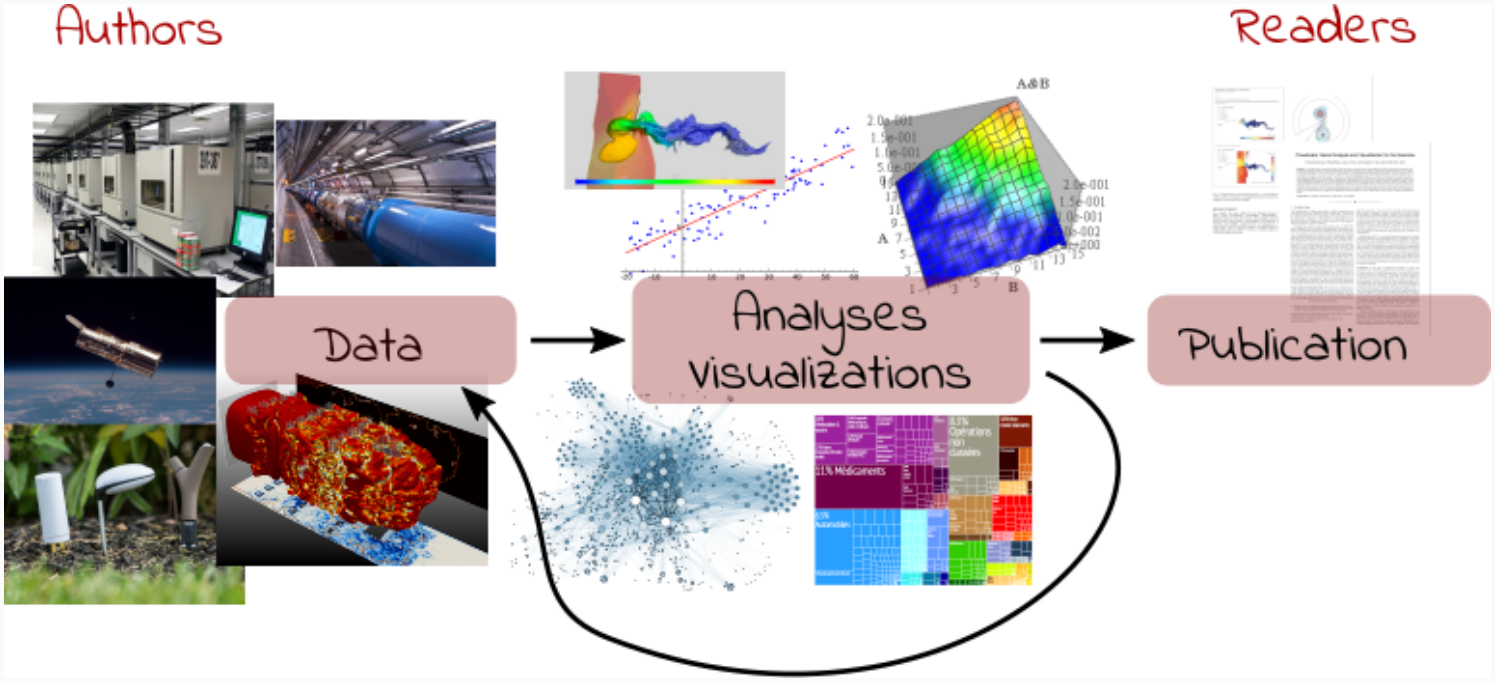
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



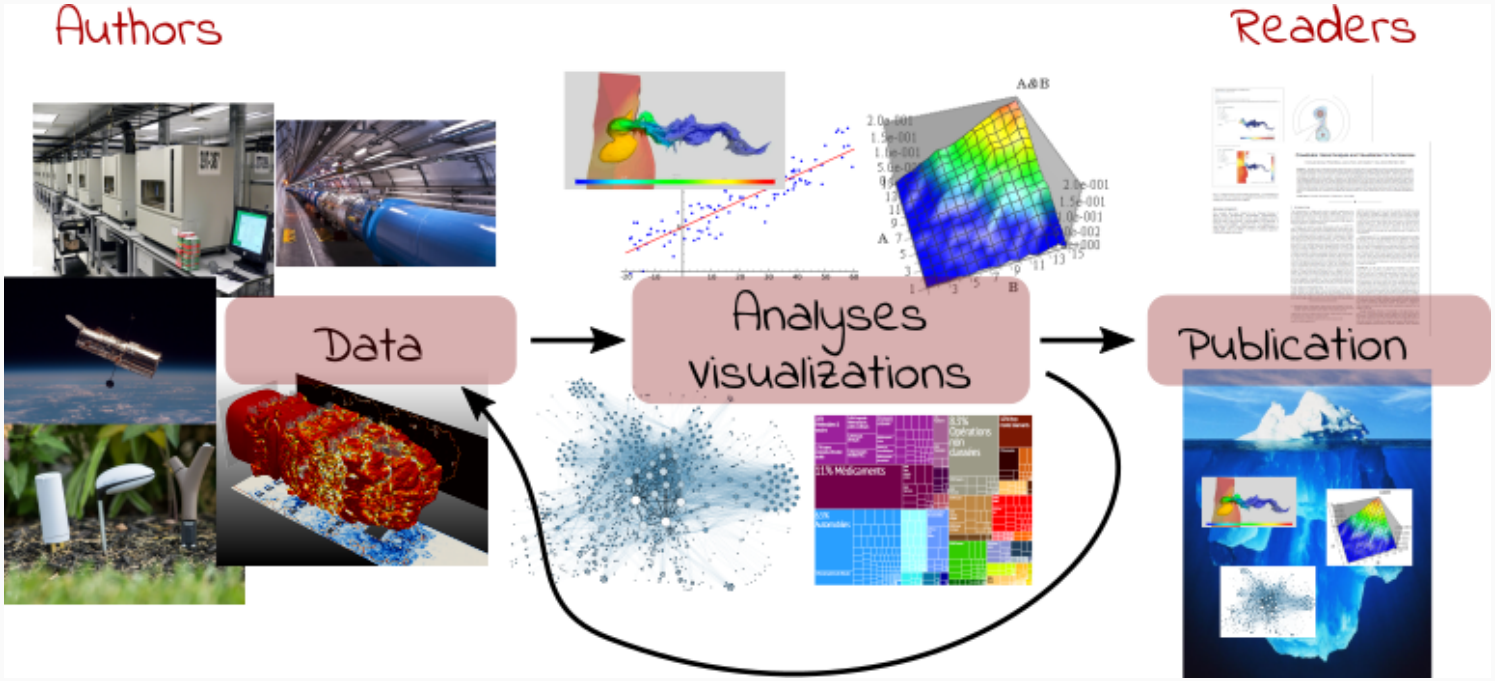
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



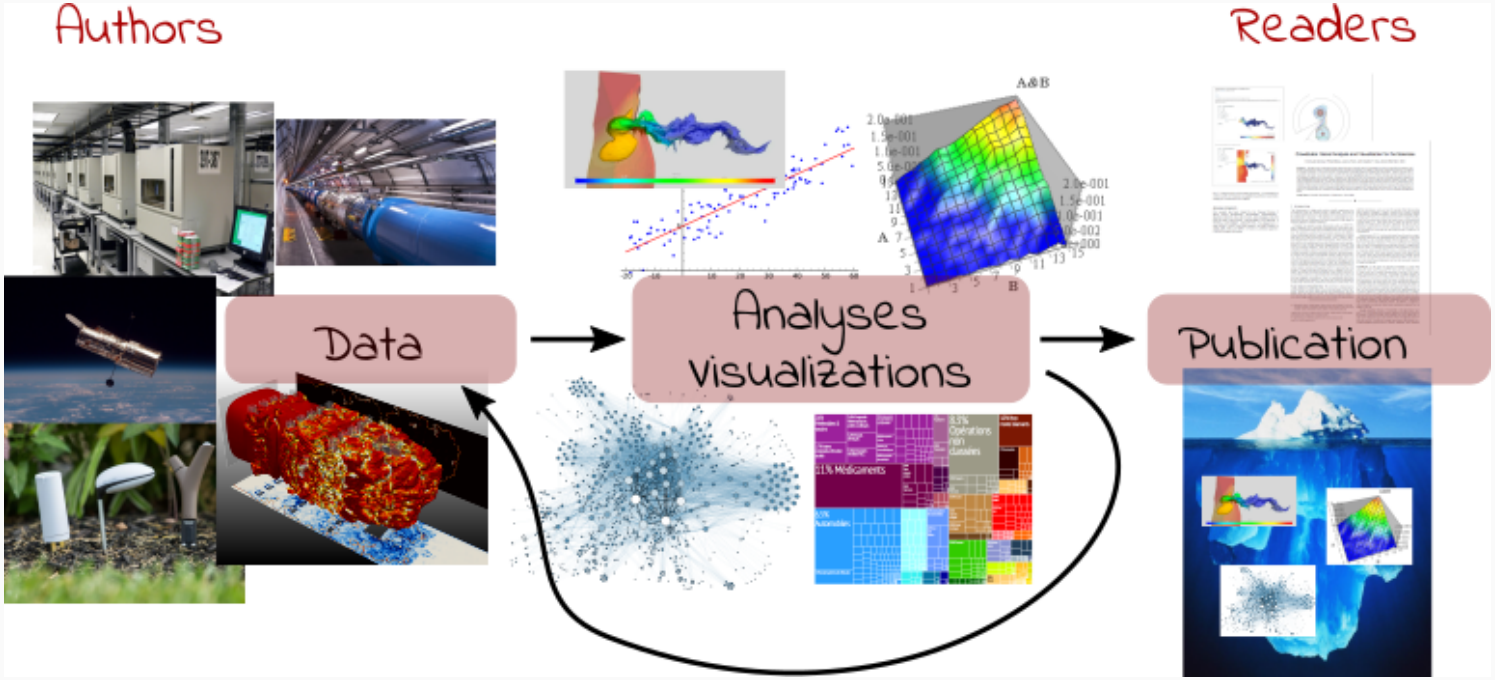
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



Reproducible Research = Bridging the Gap by working Transparently

Reproducible Research Practices

"Reproducible Research": First Appearance

Claerbout & Karrenbach, meeting of the Society of Exploration Geophysics, 1992

Electronic Documents Give Reproducible Research a New Meaning **RE1.3**

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provid[illegible] of reproducibility in compute[illegible]

In 1990, we set this sequence of goals:

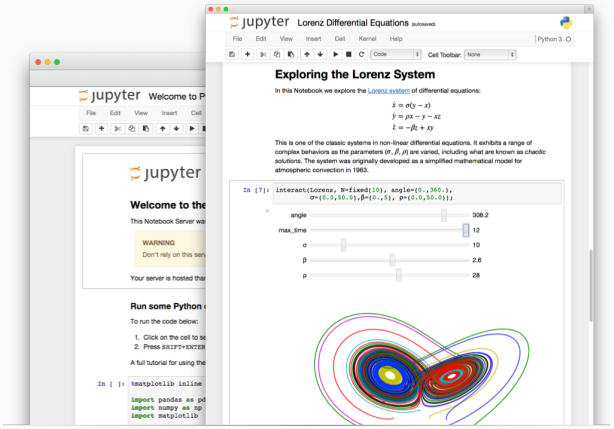
- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).
- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

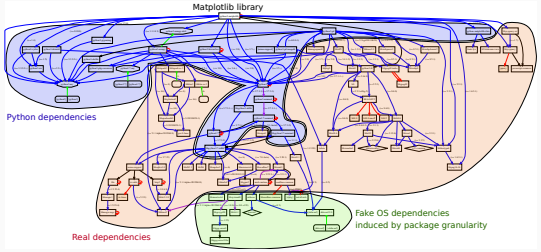
Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's interior. but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commerical television. We include all our plot software as well as freely available software from many sources, including compilers and the LaTeX word processing system. Naturally

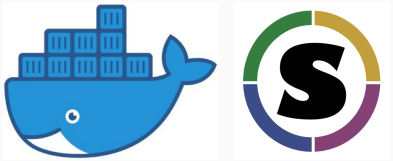
Existing Tools, Emerging Standards

Notebooks and workflows

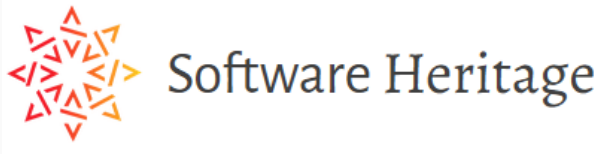


Software environments





Sharing platforms



Tool 1: Computational Notebooks/Litterate Programming

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

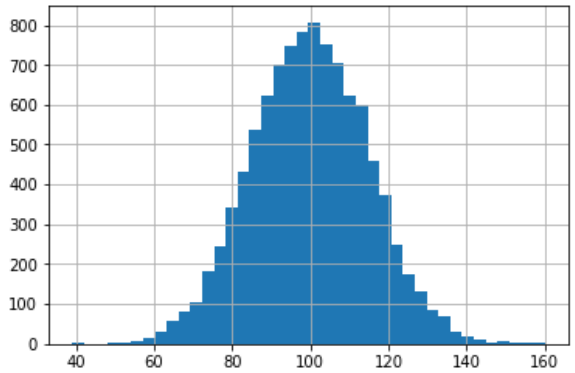
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement

jupyter example_pi (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Python 3

Code CellToolbar

Hide Prompts Hide Code Hide Outputs

```
# Un document computationnel
```

Hide Prompts Hide Code Hide Outputs

```
Mon ordinateur m'indique que $\pi$ vaut *approximativement*
```

In [1]:

Hide Prompts Hide Code Hide Outputs

```
from math import *
print(pi)
```

```
3.141592653589793
```

Hide Prompts Hide Code Hide Outputs

```
Mais calculé avec la __méthode__ des [aiguilles de Buffon]
(https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme
__approximation__ :
```

In [2]:

Hide Prompts Hide Code Hide Outputs

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

Out[2]: 3.1437198694098765

Hide Prompts Hide Code Hide Outputs

```
On peut inclure des formules mathématiques comme $\frac{1}{\sigma
\sqrt{2\pi}} \exp \left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des
_dessins qui n'ont rien à voir_ avec $\pi$ (si ce n'est une constante de
normalisation... ☺).
```

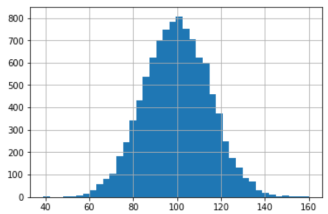
In [3]:

Hide Prompts Hide Code Hide Outputs

```
%matplotlib inline
import matplotlib.pyplot as plt

mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)

plt.hist(x,40)
plt.grid(True)
plt.show()
```



Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

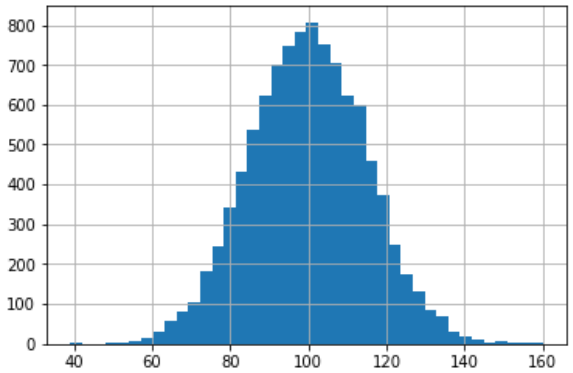
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

```
3.1437198694098765
```

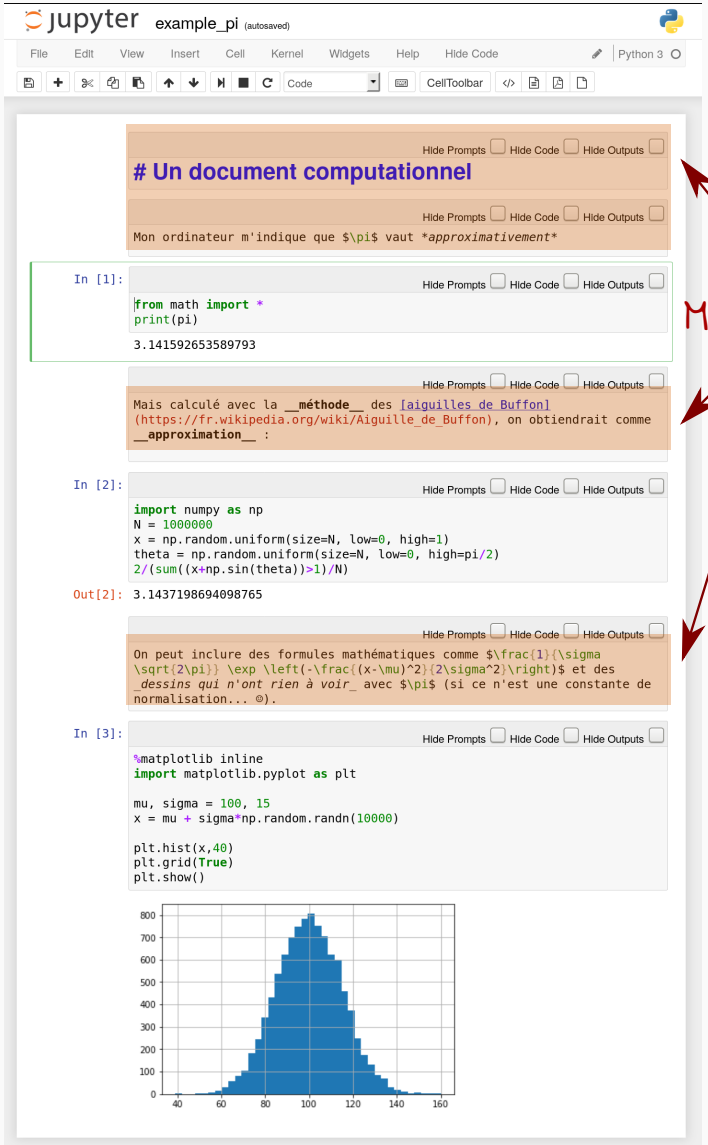
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement

Document final



Mark Down

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

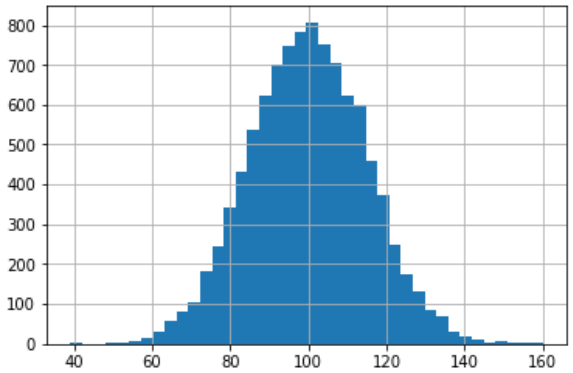
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

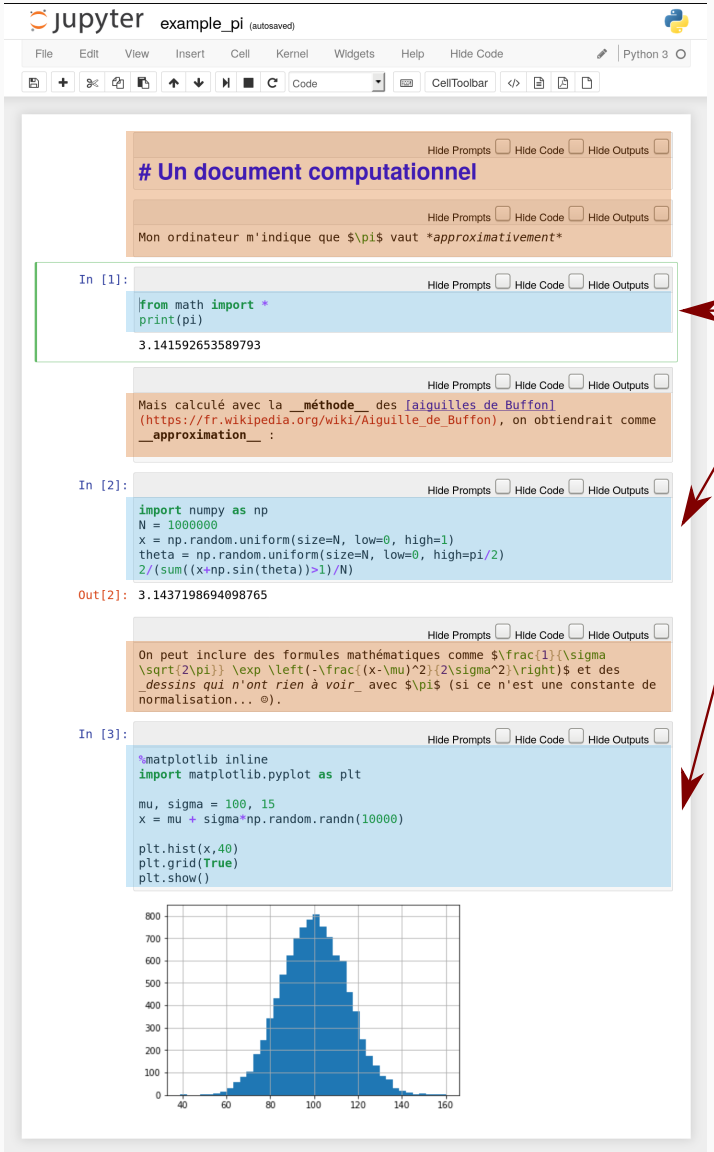
```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Code

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

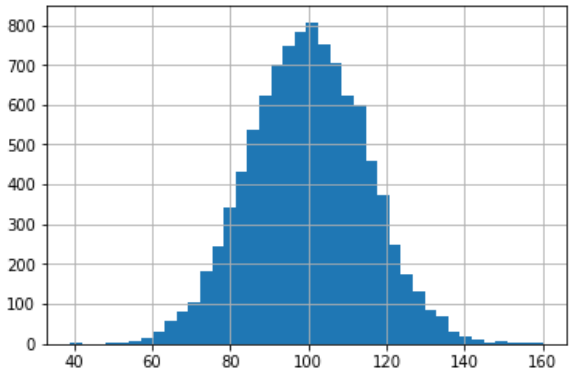
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

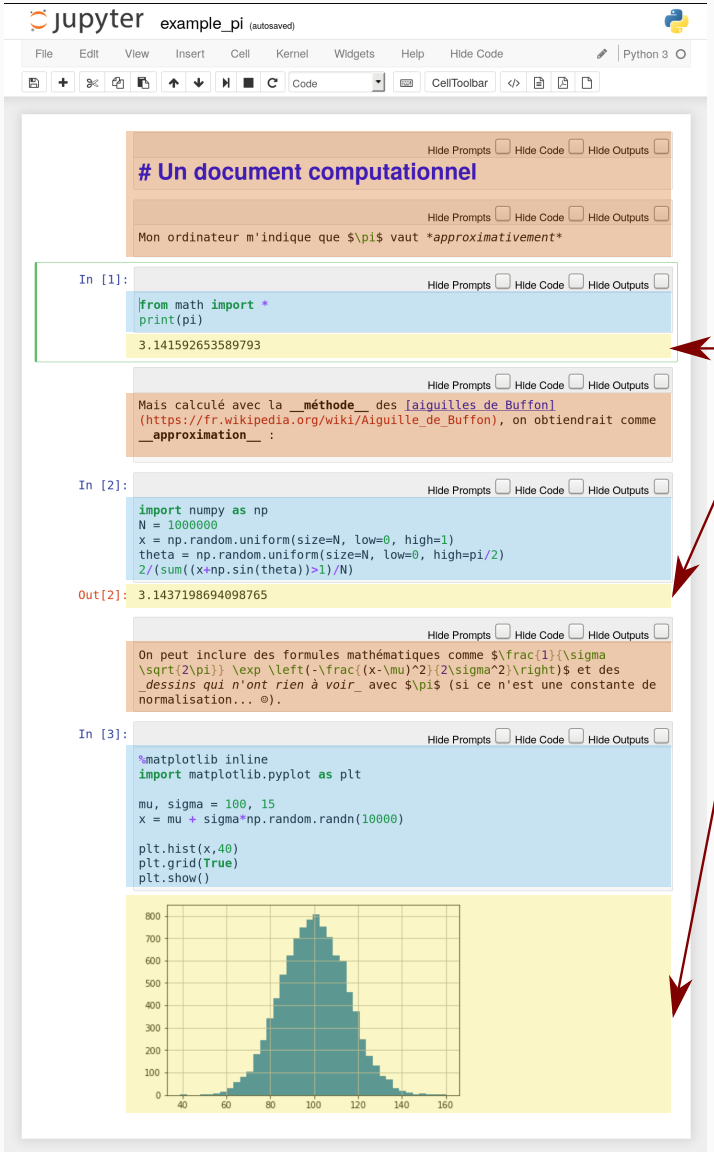
```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Résultats

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

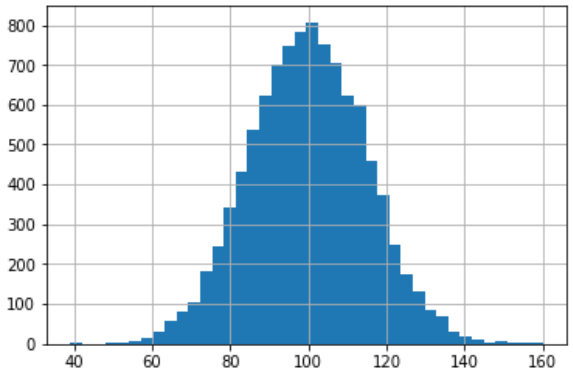
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

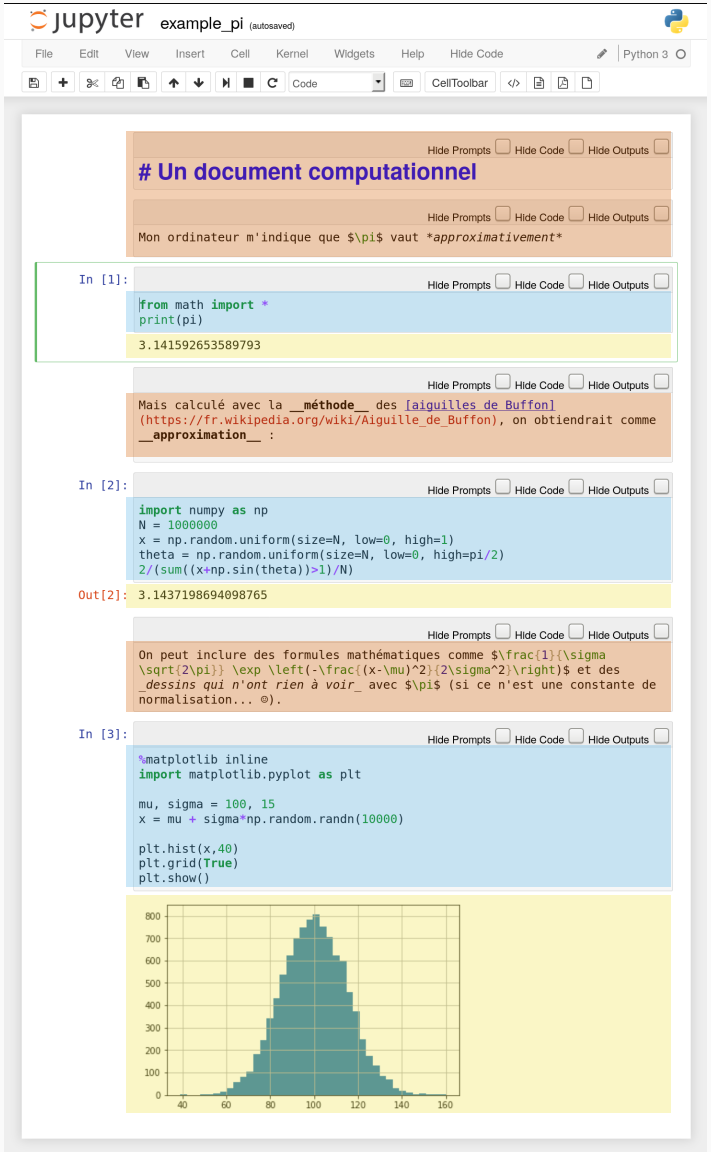
```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Export

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

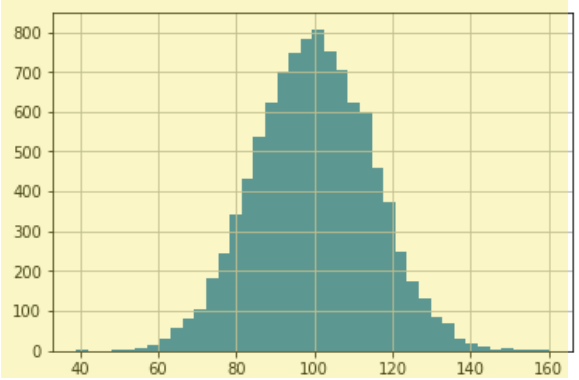
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

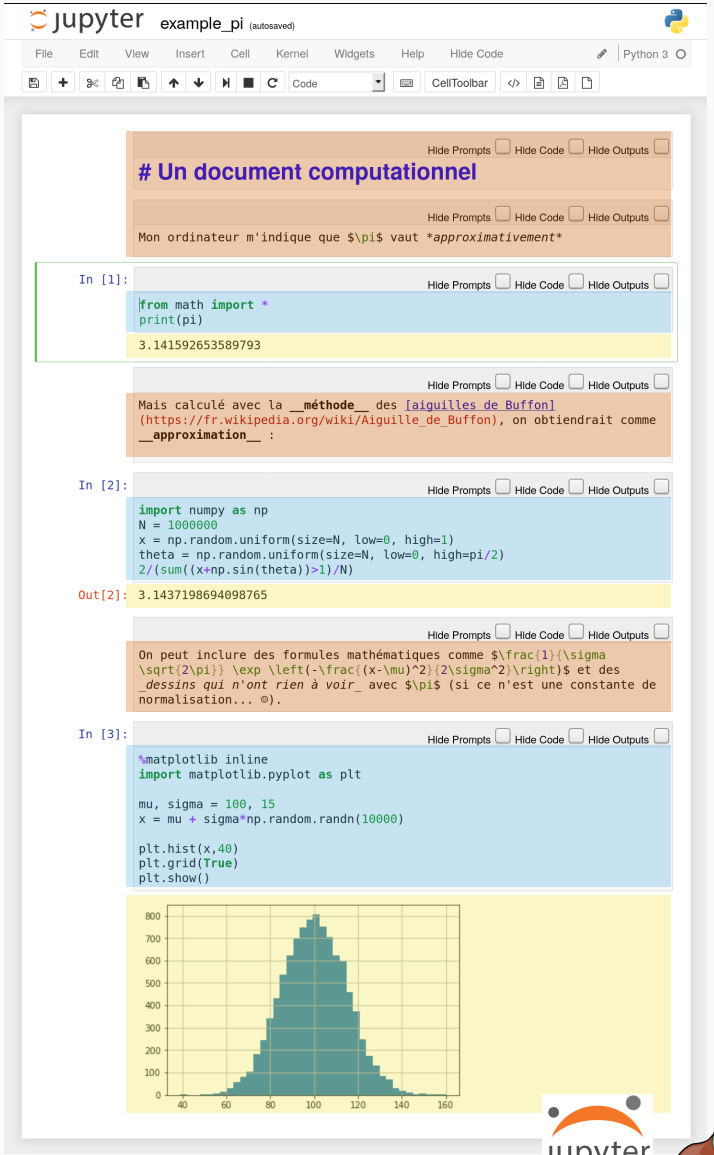
```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Export

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

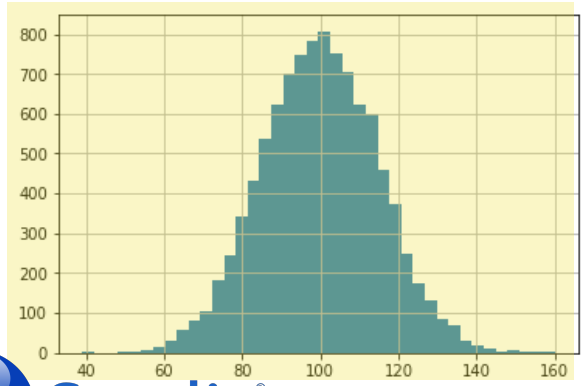
```
3.141592653589793
```

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

```
3.1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 2: Fighting Software Environments Nightmare

What is hiding behind a simple

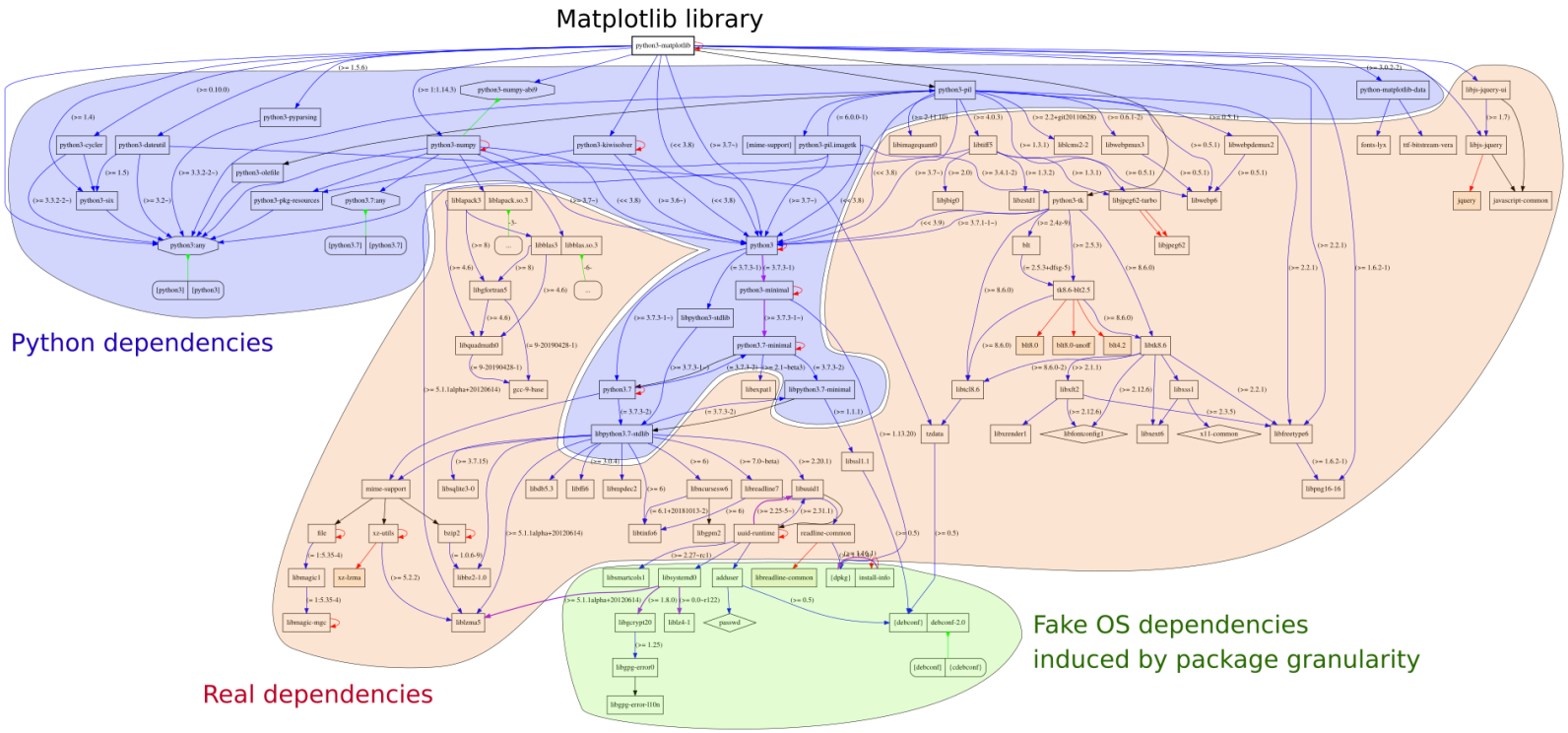
```
import matplotlib
```

```
Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2),
python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz,
libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1),
python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~),
python3-cycler (>= 0.10.0), python3:any (>= 3.3.2-2~), libc6 (>=
2.14), libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libpng16-16 (>=
1.6.2-1), libstdc++6 (>= 5.2), zlib1g (>= 1:1.1.4)
```

What is hiding behind a simple

```
import matplotlib
```

`Package: python3-matplotlib`



Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

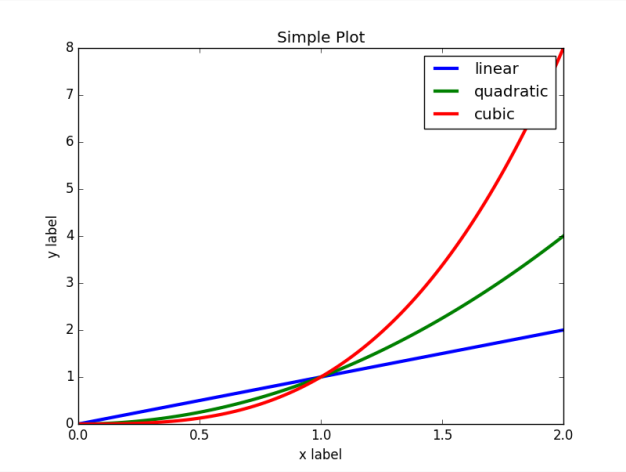
```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

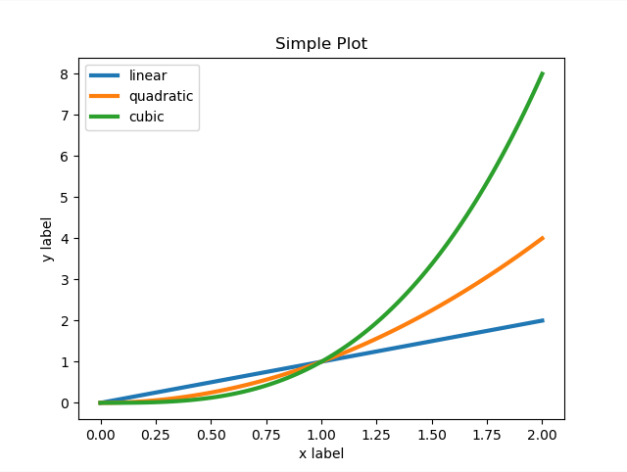
```
3
3.3333333333333335
```

Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

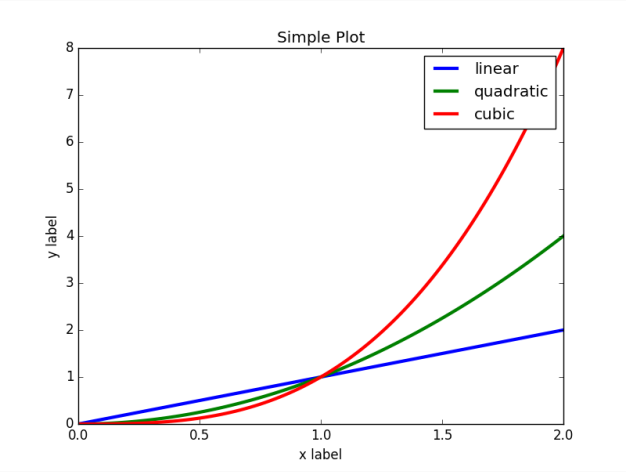


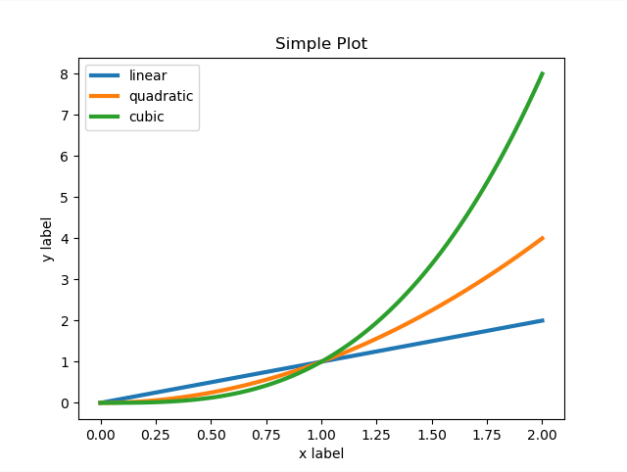


Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```



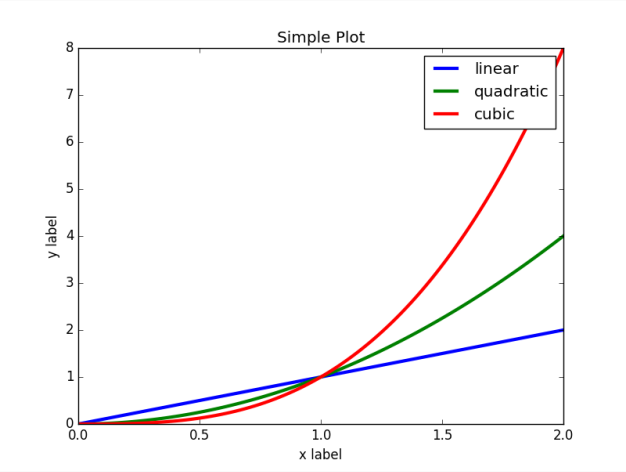


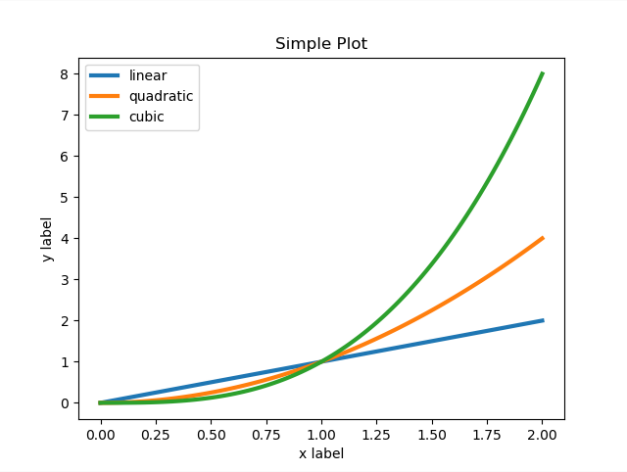
Cortical Thickness Measurements (PLOS ONE, June 2012): *FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```





Cortical Thickness Measurements (PLOS ONE, June 2012): *FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

ReproZip

Guix

Tool 3: Fighting Information Loss with Archives

D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), Jan 2003.

> *The half-life of a referenced URL is approximately 4 years from its publication date.*

P. Habibzadeh. *Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals*. Applied Clinical Informatics. 4 (4), 2013

> *half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*

Tool 3: Fighting Information Loss with Archives

D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), Jan 2003.

> *The half-life of a referenced URL is approximately 4 years from its publication date.*

P. Habibzadeh. *Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals*. Applied Clinical Informatics. 4 (4), 2013

> *half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*

Article archives arXiv.org HAL archives-ouvertes.fr

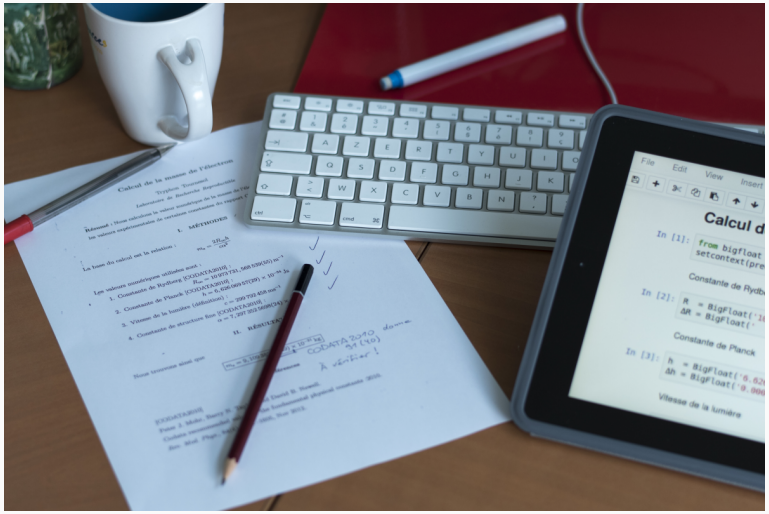
Data archives figshare zenodo

Software Archive Software Heritage

or = awesome collaborations $\neq$ archive

Changing Research Practices

Soft. Engineering, Statistics, and Reproducible Research in the curricula



- Book on RR *Vers une recherche reproductible: Faire évoluer ses pratiques*
- MOOC on RR (3rd edition, ongoing)
- A new "Advanced RR" MOOC (2021?)
  - Software environment control
  - Scientific workflow
  - Managing data

Manifesto: "*I solemnly pledge*" (WSSSPE, Lorena Barba, FAIR)

1. I will teach my graduate students about reproducibility
2. All our research code (and writing) is under version control
3. We will always carry out verification and validation
4. We will share data, plotting script & figure under CC-BY
5. We will upload the preprint to arXiv at the time of submission of a paper
6. We will release code at the time of submission of a paper
7. We will add a "Reproducibility" declaration at the end of each paper
8. I will keep an up-to-date web presence

What about Computer Science ?

All this is about Computational Sciences. Should we care ?

Computer Science is young and inherits from Mathematics, Engineering, Nat. Sciences, Linguistic, ...

Purely theoretical scientists whose practice is close to mathematics *may* not be concerned (can't publish a math article without releasing the proofs).

- Have a look at talk by Vladimir Voevodsky in 2014 at Princeton 😉

All this is about Computational Sciences. Should we care ?

Computer Science is young and inherits from Mathematics, Engineering, Nat. Sciences, Linguistic, ...

Purely theoretical scientists whose practice is close to mathematics *may* not be concerned (can't publish a math article without releasing the proofs).

- Have a look at talk by Vladimir Voevodsky in 2014 at Princeton 😉

  Computer science is not more related to computers than Astronomy to telescopes

  – Dijkstra (mis-attributed)

Right, why should we care about computers? They are deterministic machines after all, right? 😉

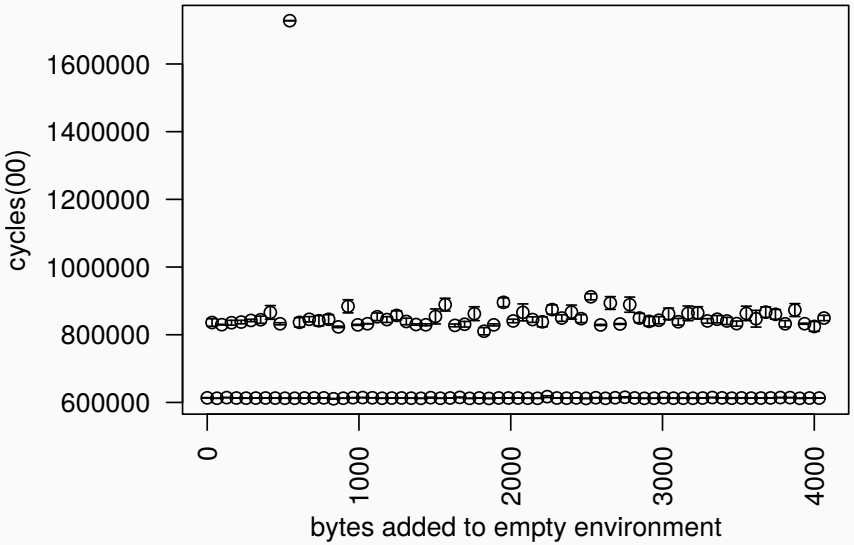
Model $\neq$ **Reality**. Although designed and built by human beings, computer systems are so complex that mistakes easily slip in...

Experimenting with computers

Machines are real!

Brendan Gregg: Shouting in the data center

Machines are complicated



Mytkowicz et al. *Producing wrong data without doing anything obviously wrong!* ACM SIGPLAN Not. 44(3), March 2009

Our reality evolves!!! The hardware keeps evolving so most results on old platforms quickly become obsolete (although, we keep building on such results 😉).

We need to regularly revisit and allow others to build on our work!

Computer performance ? Well, I design algorithms!

- ”Real” problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof widgets~~, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular workloads that impact both their running time and the quality of the solutions

Computer performance ? Well, I design algorithms!

- ”Real” problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof widgets~~, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular workloads that impact both their running time and the quality of the solutions

Image Processing: True horror stories, E. Meinhardt-Llopis, CANUM 2016

- *The proposed multigrid algorithm converges to the solution of the problem in O(N)* using biharmonic functions
- Surprisingly, our naive multi-scale Gauss-Seidel converges much faster

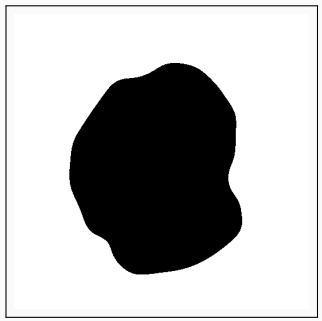
Computer performance ? Well, I design algorithms!

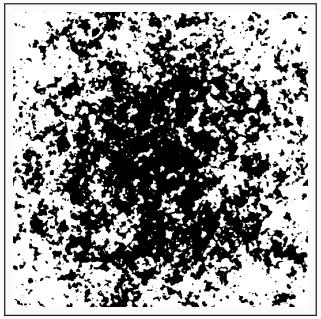
- ”Real” problems are all NP-hard, Log-APX, etc.
- Real workload = ~~NP-completeness proof widgets~~, regularities and properties (difficult to formally state but that should be exploited)

Algorithms are evaluated on particular workloads that impact both their running time and the quality of the solutions

Image Processing: True horror stories, E. Meinhardt-Llopis, CANUM 2016

- *The proposed multigrid algorithm converges to the solution of the problem in O(N)* using biharmonic functions
- Surprisingly, our naive multi-scale Gauss-Seidel converges much faster





I design Machine Learning Algorithms

Machine Learning: Trouble at the lab, The Economist 2013

> *According to some estimates, three-quarters of published scientific papers in the field of machine learning are bunk because of this "overfitting".* – *Sandy Pentland (MIT)*

Every month in CACM, there is an article about the ethical consequences of Machine Learning on:

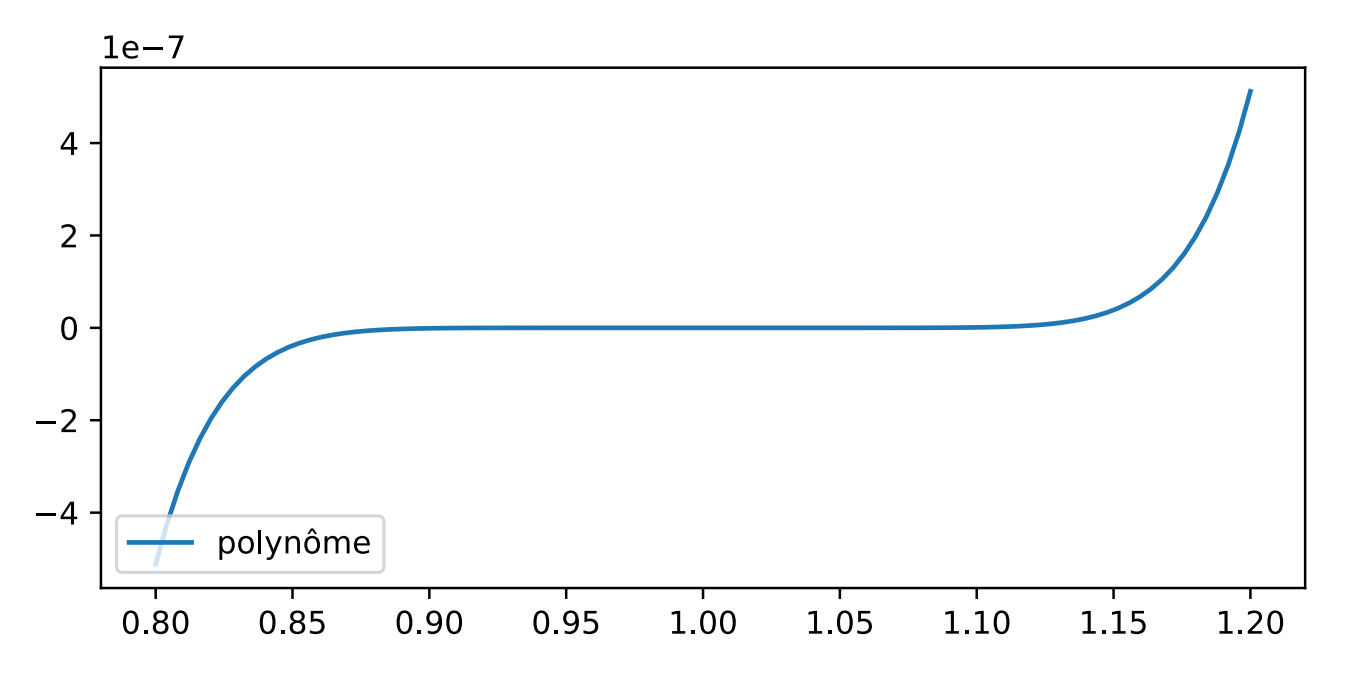
- Car driving, Autonomous guns, Law enforcement (risk assessment, predictive policing), ...

  It's Not the Algorithm, It's the Data (CACM, Feb. 2017)

- Advertising, Loan attribution, Selection at University, Organ transplant

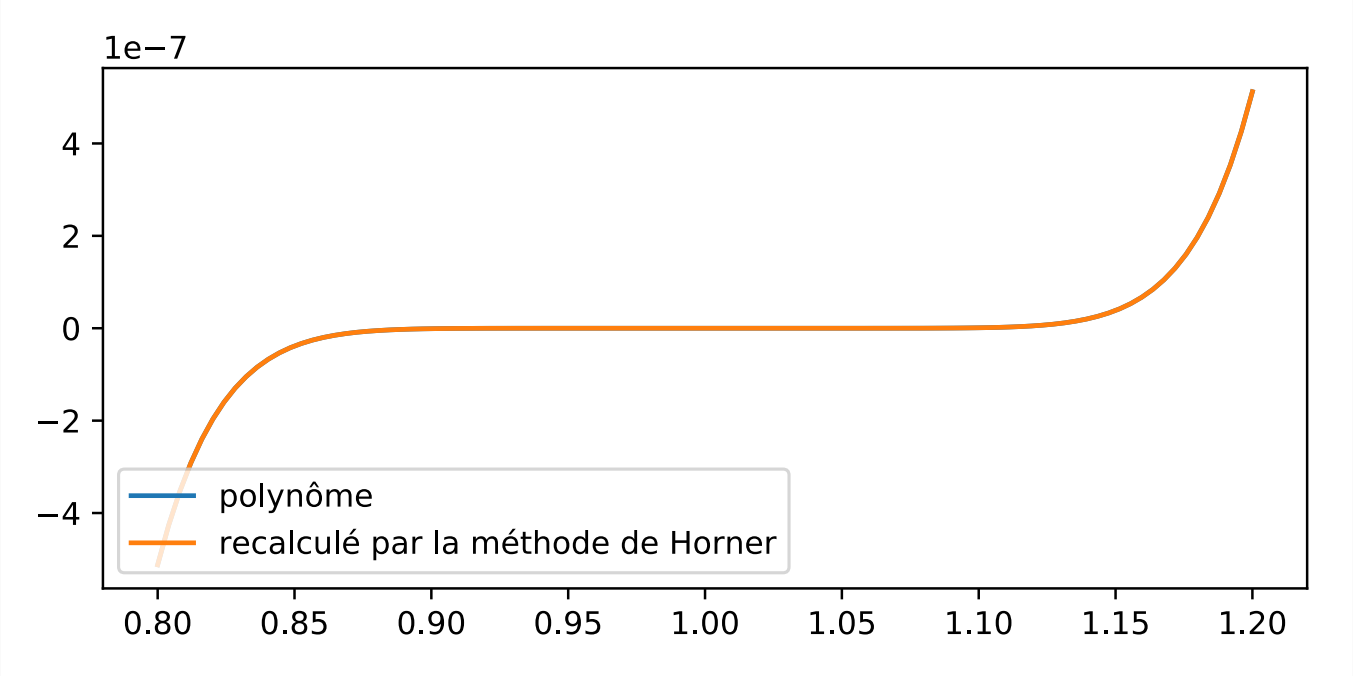
Increasing society concern about fairness and transparency

All I care about is the algorithm output (FP)



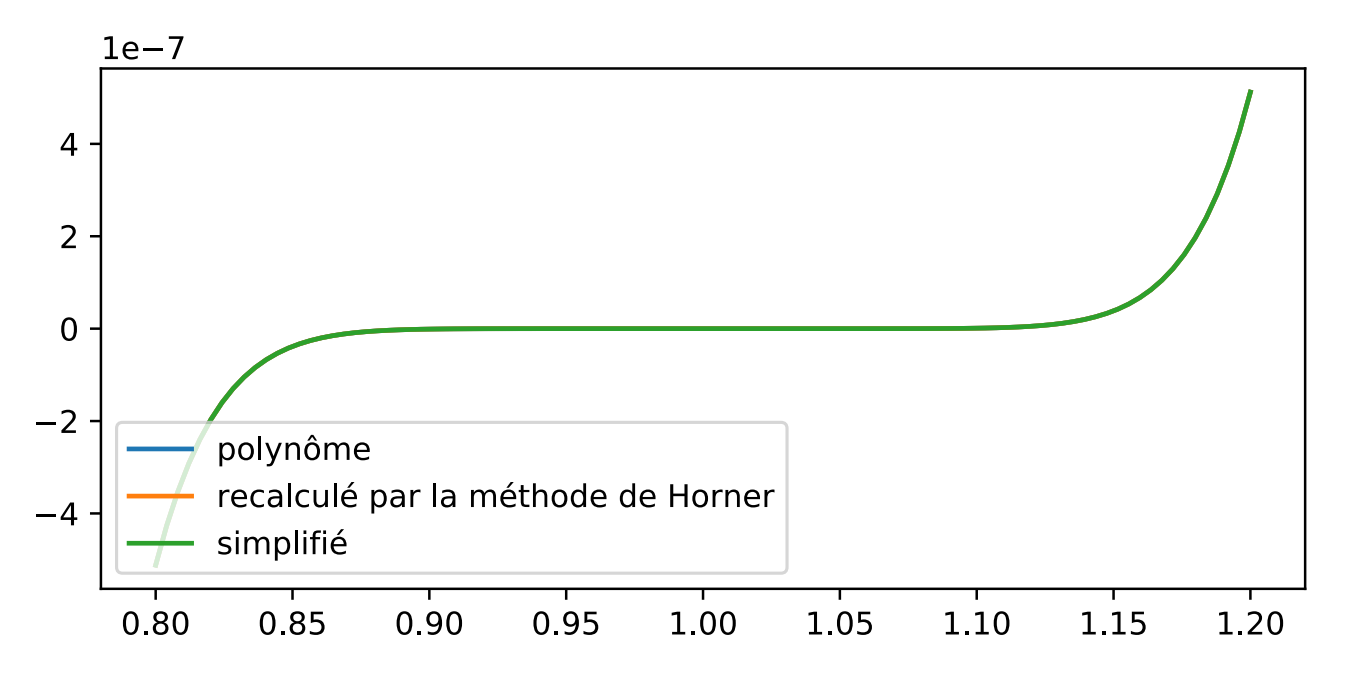

```
def polynome(x):
    return x**9 - 9.*x**8 + 36.*x**7 - 84.*x**6 + 126.*x**5 \
           - 126.*x**4 + 84.*x**3 - 36.*x**2 + 9.*x - 1.
```

Floating-point arithmetic



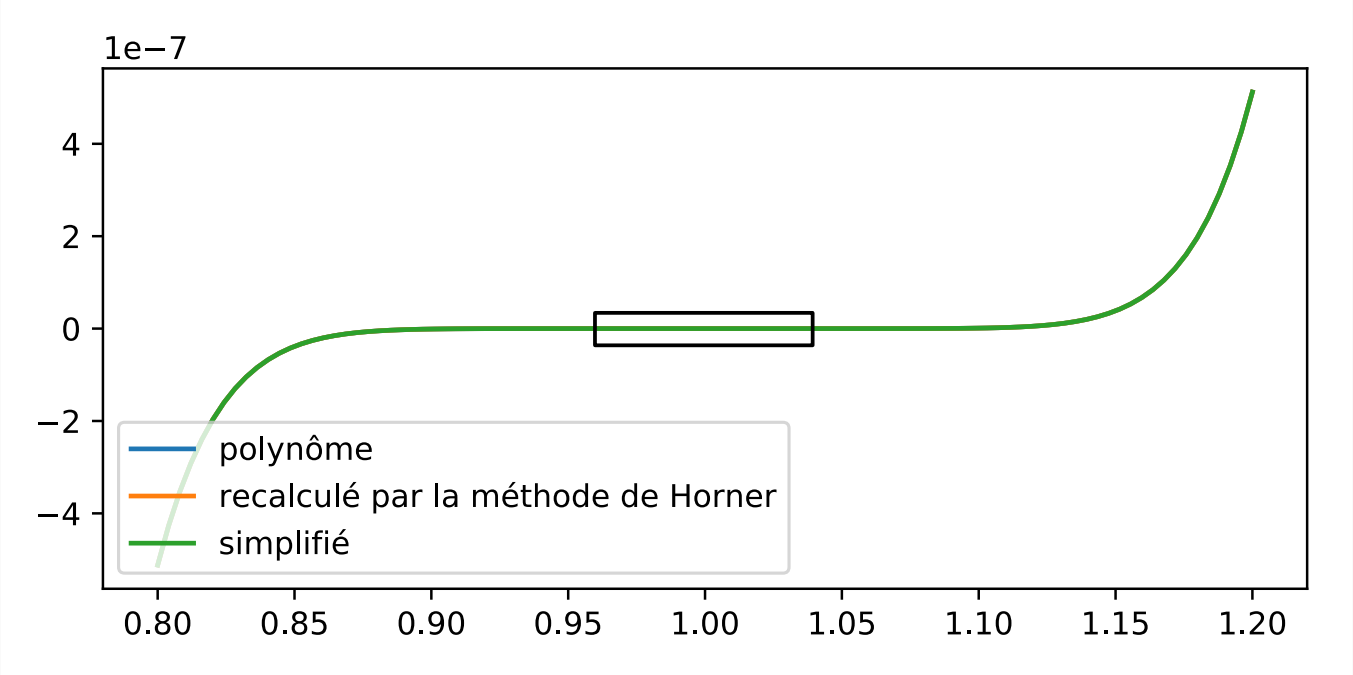

```
def horner(x):
    return x*(x*(x*(x*(x*(x*(x*(x*(x - 9.) + 36.) - 84.) + 126.) \
           - 126.) + 84.) - 36.) + 9.) - 1.
```

Floating-point arithmetic

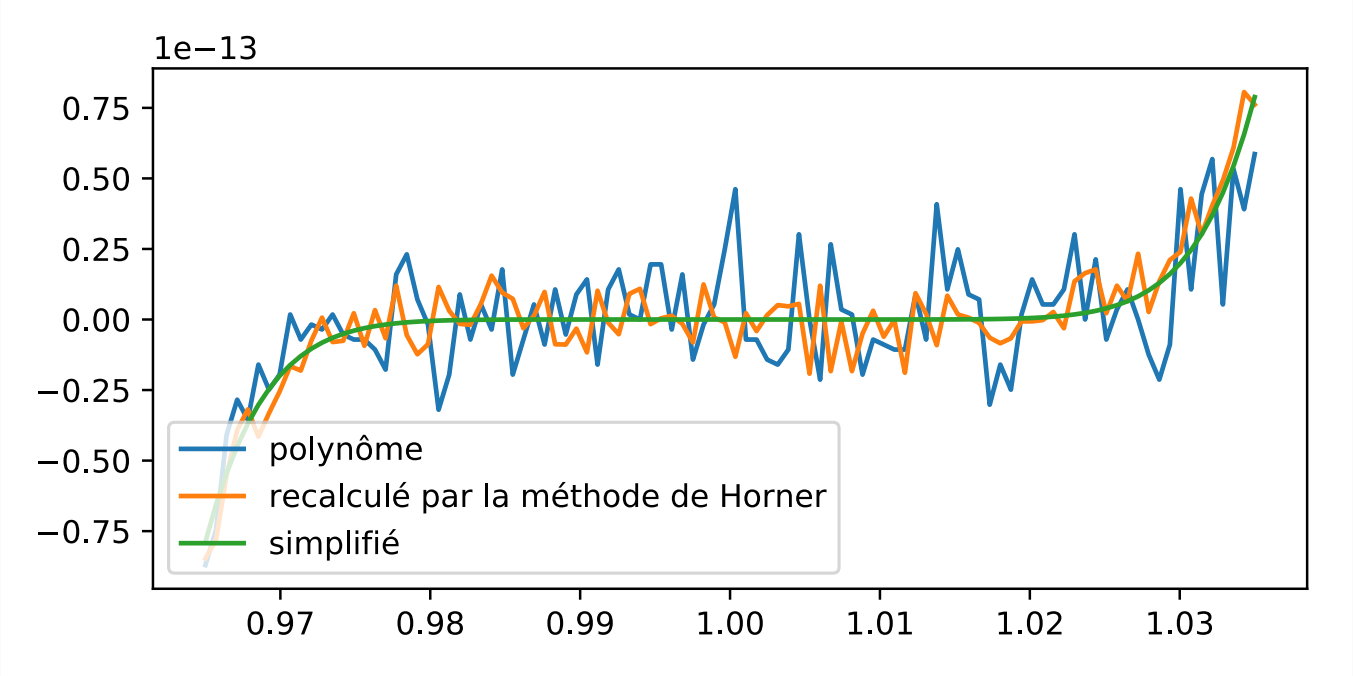



```
def simple(x):
    return (x-1.)**9
# Easy! ;)
```

Floating-point arithmetic



Floating-point arithmetic



Rounding

- Every operation includes implicit rounding.
- a+b is actually $\mathbf{round}(a+b)$.
- Unfortunately:

$$\mathbf{round}(\mathbf{round}(a+b)+c) \neq \mathbf{round}(a+\mathbf{round}(b+c)).$$

- Operation order therefore matters.

For a reproducible computation, operation order must be preserved!!!

How to explain it to my compiler?

To speed up computations, compilers may change operation order, and thus results.

Two options for computing reproducibly:

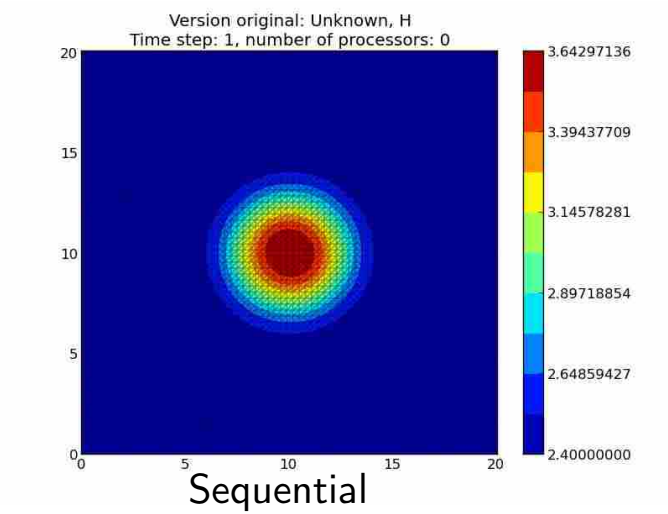
1. Insist on the preservation of operation order,
   - if the language permits it.
   - Example: Module ‘ieee_arithmetic‘ in Fortran 2003
2. Make compilation reproducible:
   - Record the precise compiler version
   - Record all compilation options

Telemac2D: the simplest gouttedo simulation

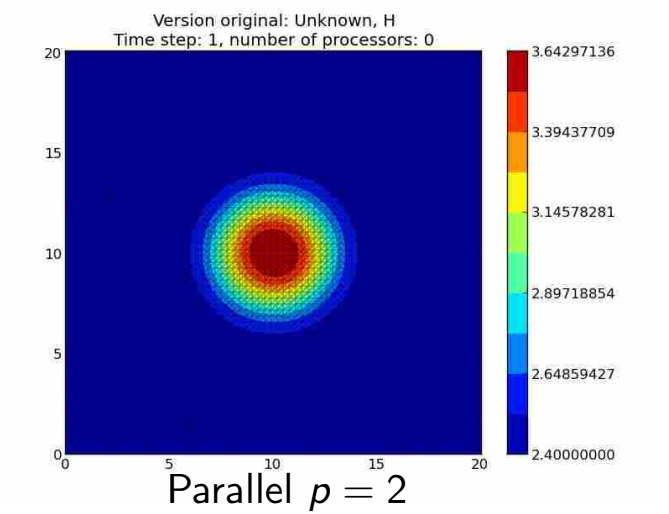
The gouttedo test case

- 2D-simulation of a water drop fall in a square bassin
- Unknown: water depth for a 0.2 sec time step
- Triangular mesh: 8978 elements and 4624 nodes

Expected numerical reproducibility (time step = 1, 2, ...)



Sequential



Parallel $p = 2$

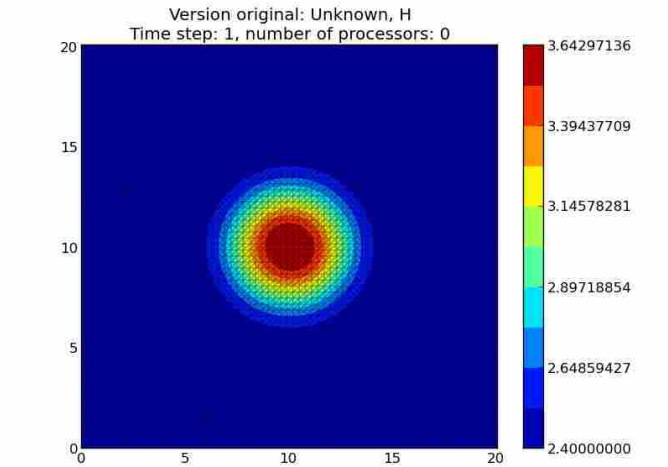
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

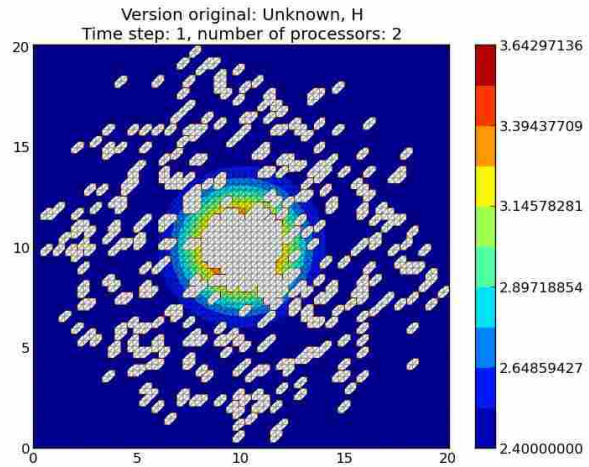
A white plot displays a non-reproducible value

Numerical reproducibility?

time step $= 1$



Sequential



Parallel $p = 2$

14 / 54

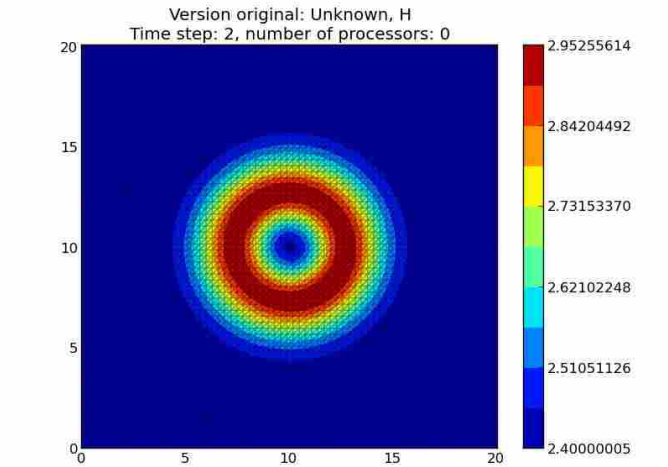
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

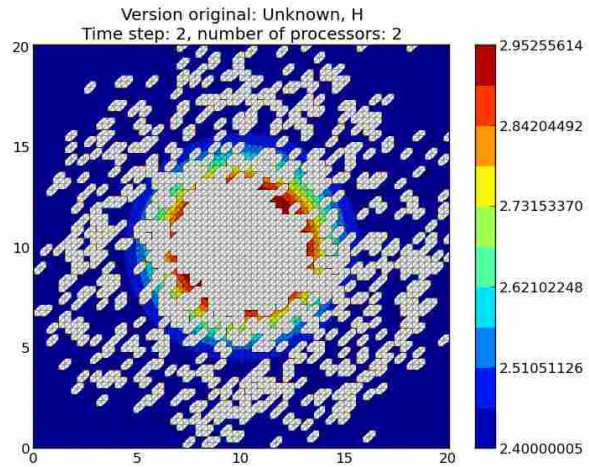
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 2



Sequential



Parallel $p = 2$

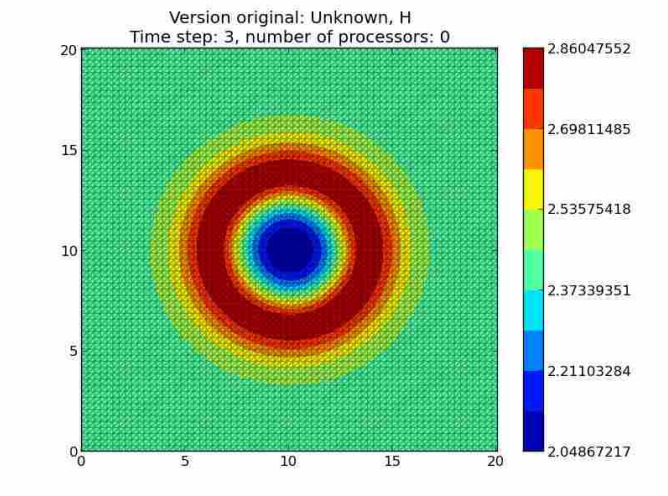
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

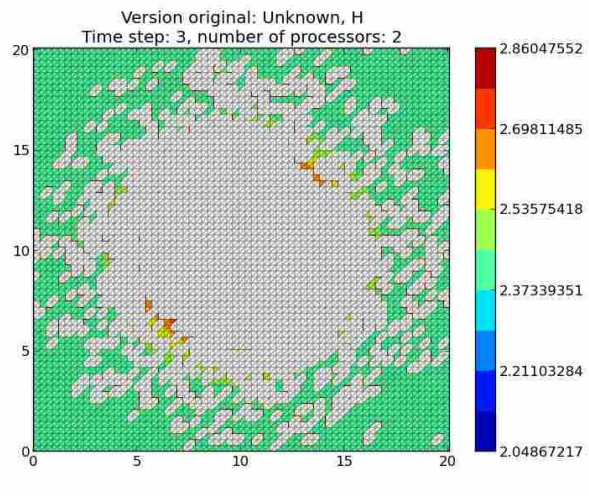
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 3



Sequential



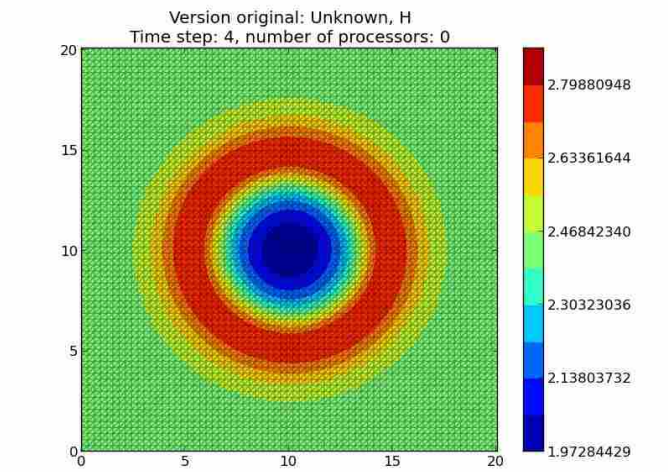
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

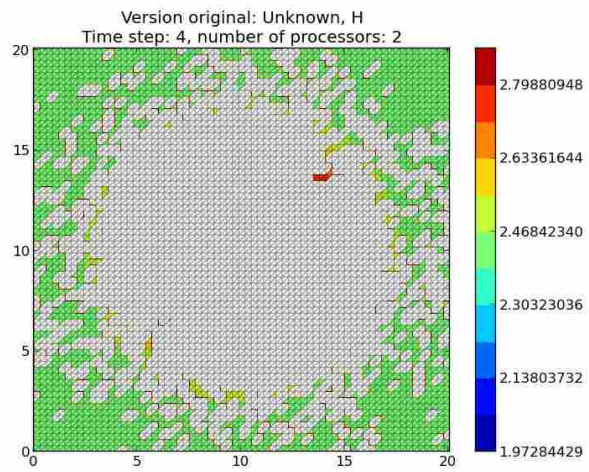
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 4



Sequential



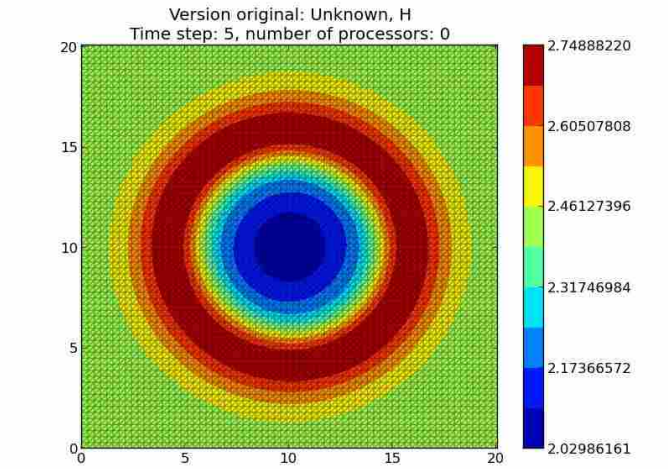
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

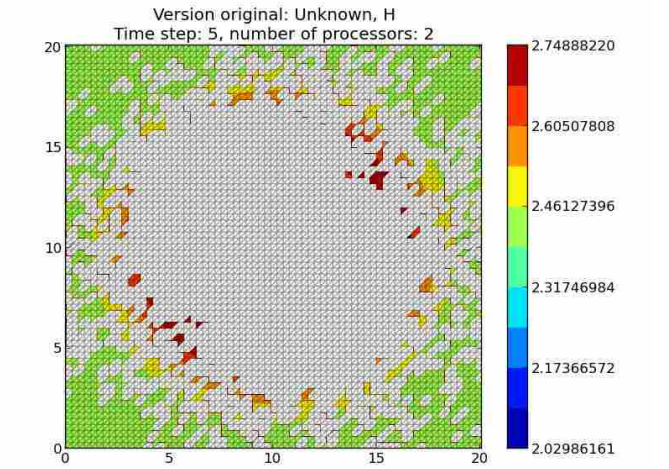
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 5



Sequential



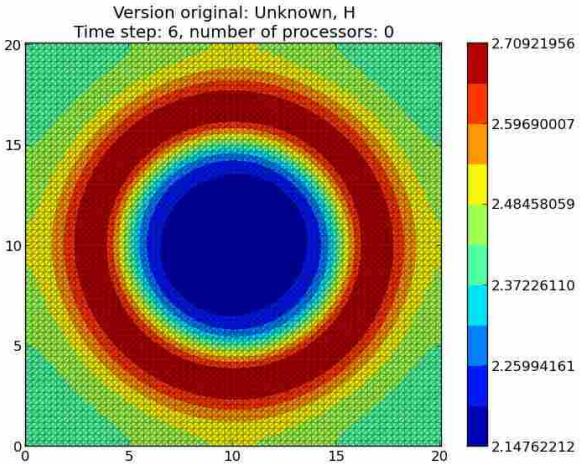
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

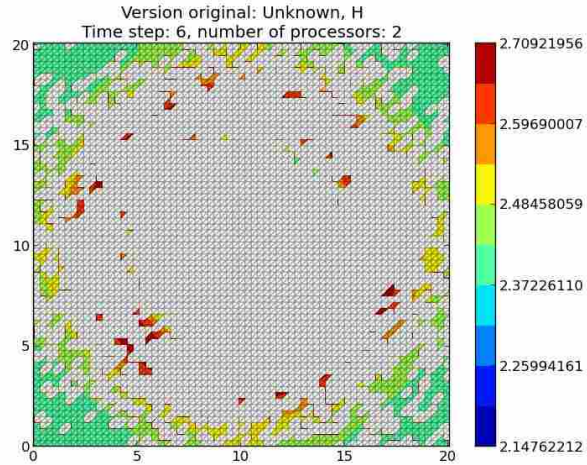
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 6



Sequential



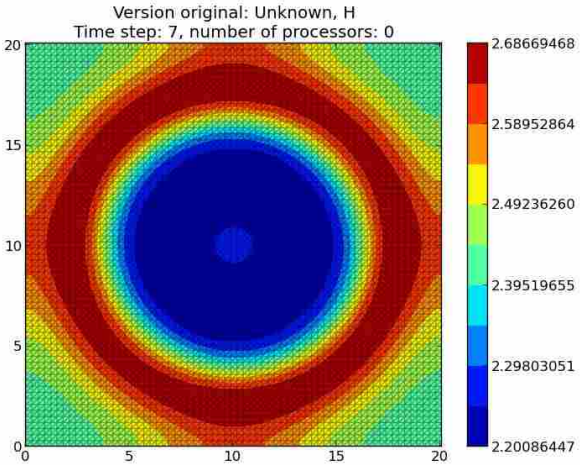
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

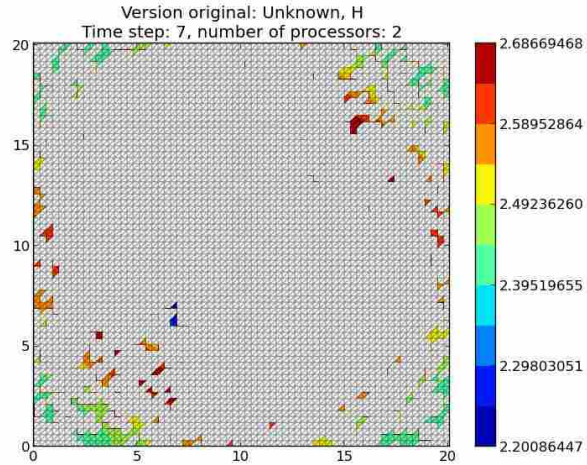
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 7



Sequential



Parallel $p = 2$

14 / 54

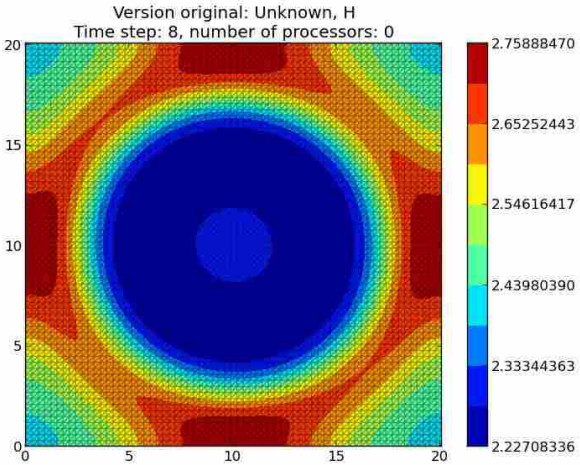
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

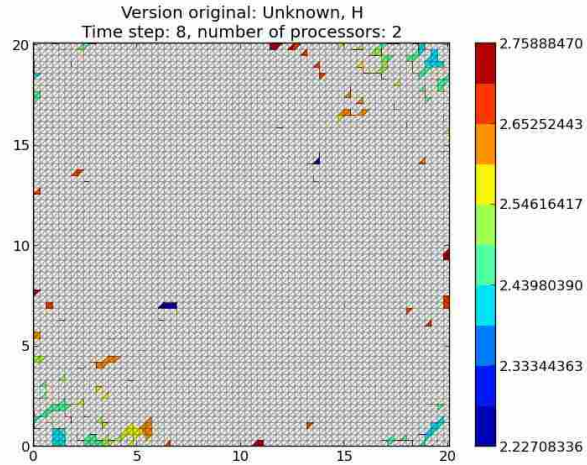
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 8



Sequential



Parallel $p = 2$

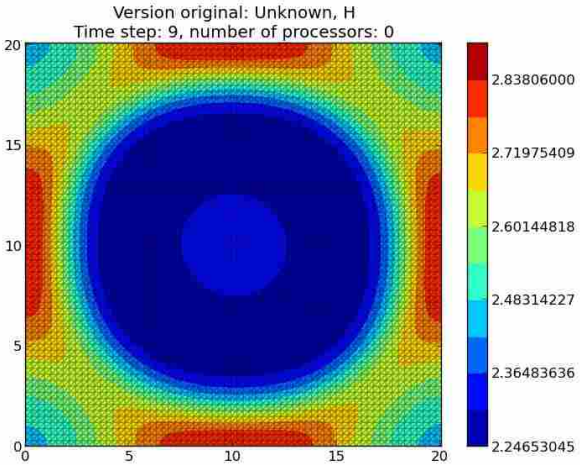
14 / 54

Courtesy of P. Langlois and R. Nheili

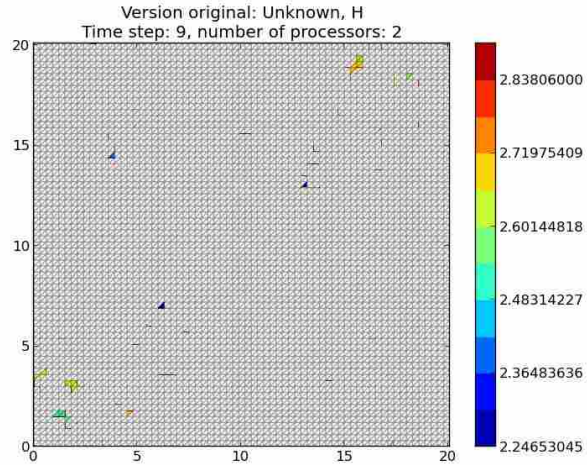
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 9



Sequential



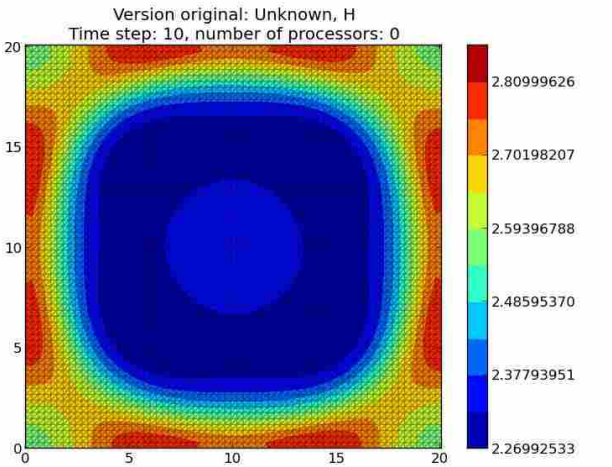
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

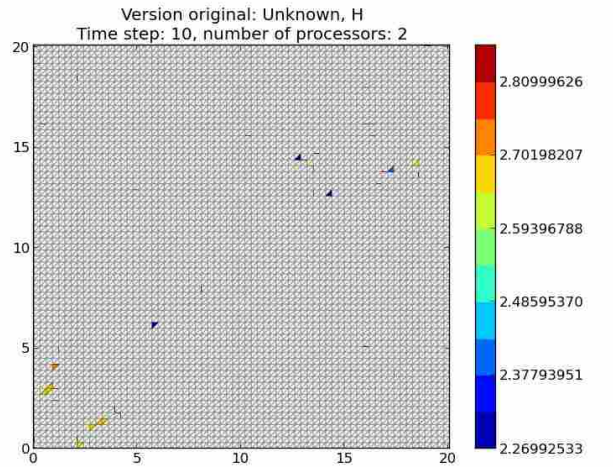
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 10



Sequential



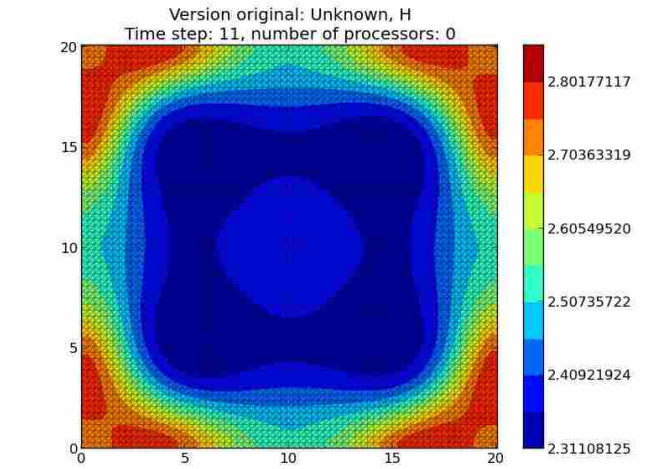
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

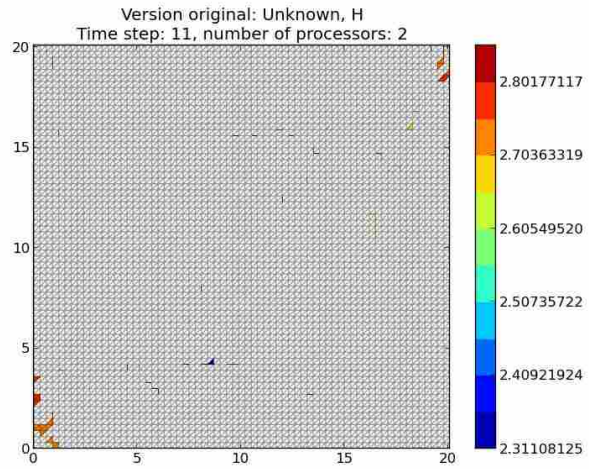
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 11



Sequential



Parallel $p = 2$

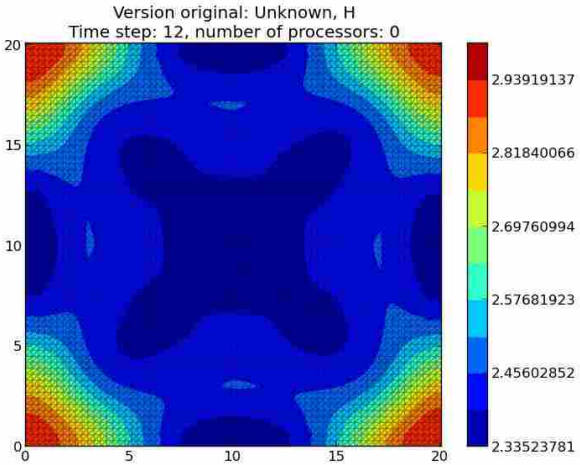
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

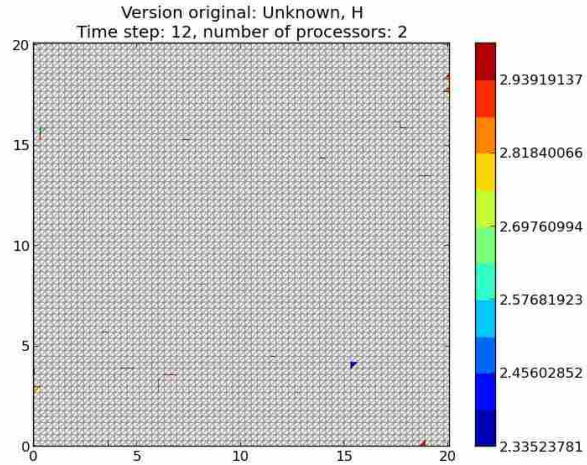
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 12



Sequential



Parallel $p = 2$

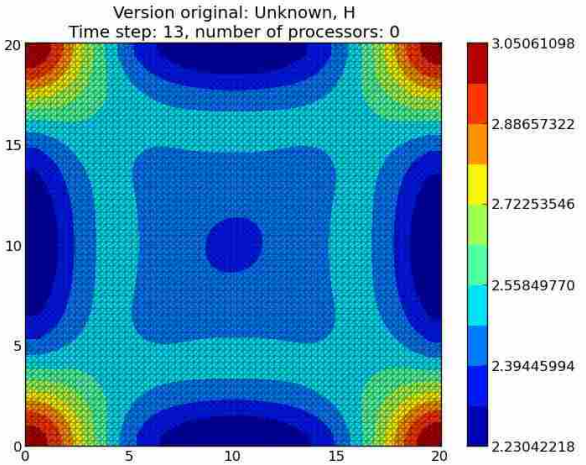
14 / 54

Courtesy of P. Langlois and R. Nheili

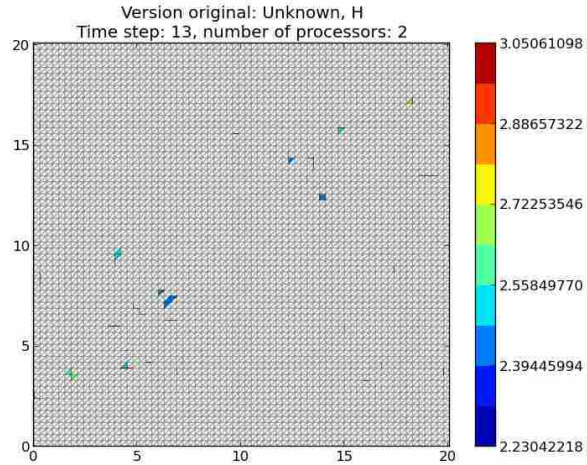
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 13



Sequential



Parallel $p = 2$

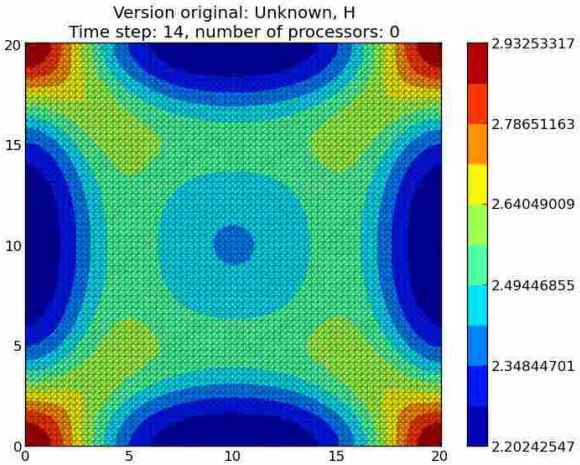
14 / 54

Courtesy of P. Langlois and R. Nheili

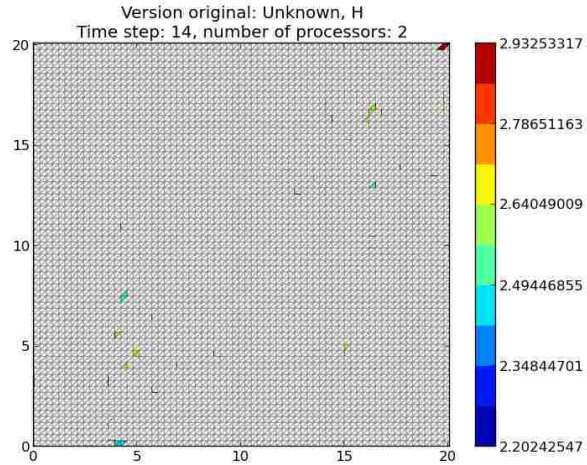
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 14



Sequential



Parallel $p = 2$

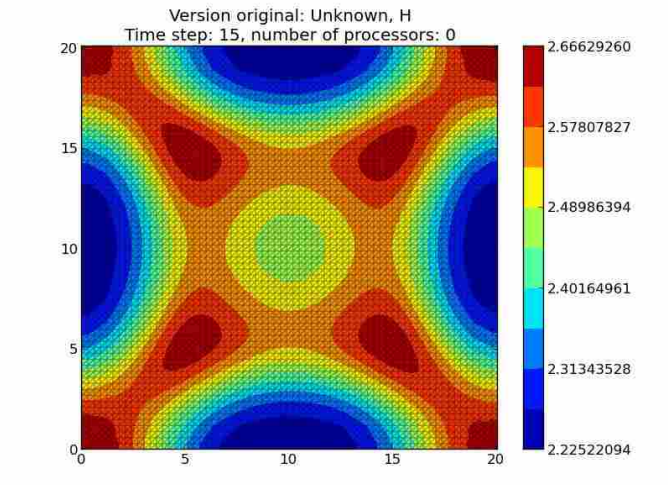
14 / 54

Courtesy of P. Langlois and R. Nheili

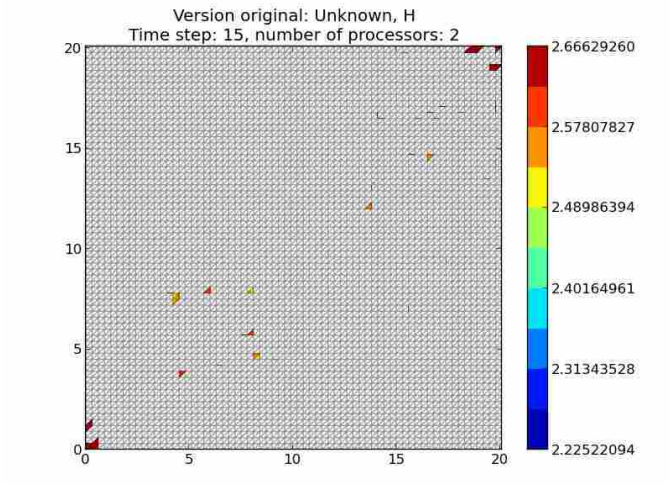
A white plot displays a non-reproducible value

NO numerical reproducibility!

time step = 15



Sequential

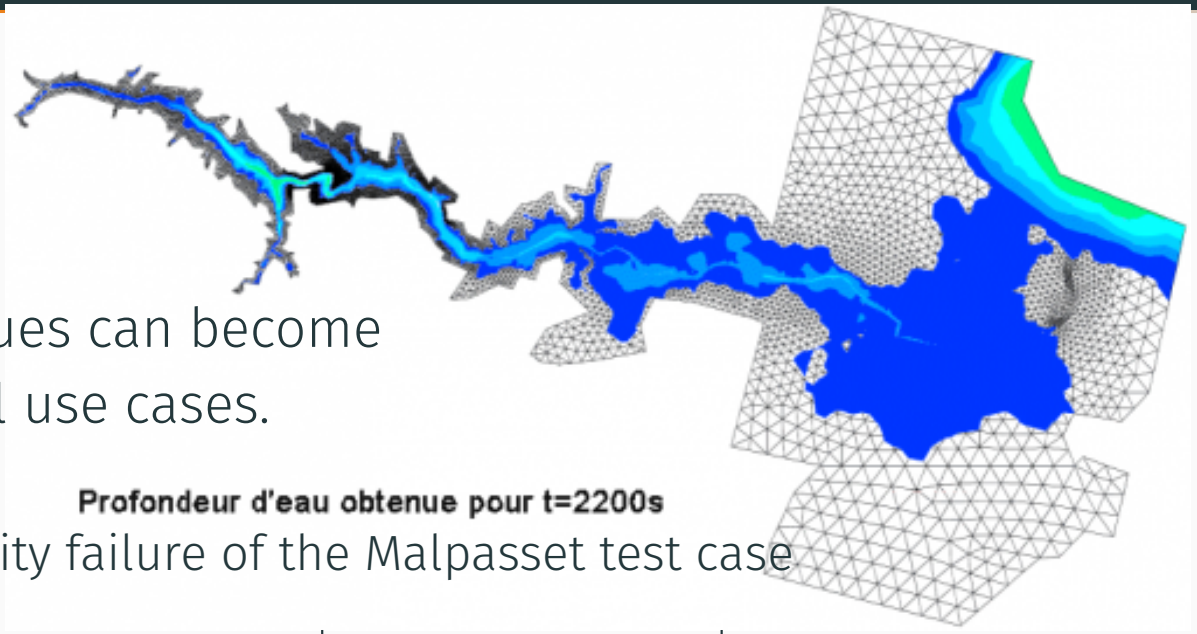


Parallel $p = 2$

14 / 54

Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?



These numerical issues can become quite harmful in real use cases.

TABLE 1.1: Reproducibility failure of the Malpasset test case

	The sequential run	a 64 procs run	a 128 procs run
depth H	0.3500122E-01	0.2748817E-01	0.1327634E-01
velocity U	0.4029747E-02	0.4935279E-02	0.4512116E-02
velocity V	0.7570773E-02	0.3422730E-02	0.7545233E-02

Numerical reproducibility?: Approximations in the model, in the algorithm, in its implementation, in its execution.

The whole chain needs to be revisited.

Courtesy of P. Langlois and R. Nheili

Changing Publishing Practices

Artifact evaluation and ACM badges

Major conferences

- Supercomputing: Artifact Description (AD) mandatory, Artifact Evaluation (AE) still optional, Double blind vs. RR
- NeurIPS, ICLR: open reviews, reproducibility challenge

Joelle Pineau @ NeurIPS'18

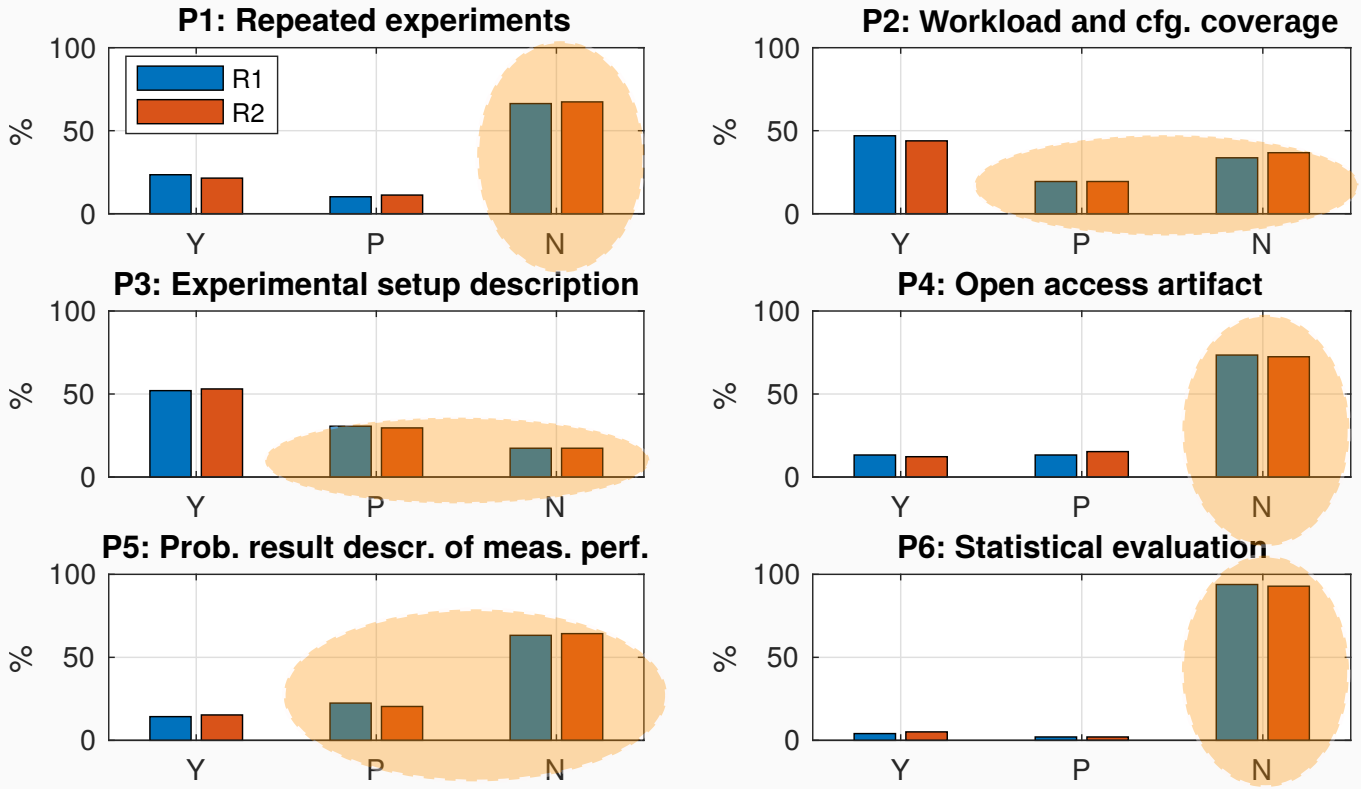
- ACM SIGMOD 2015-2019, Most Reproducible Paper Award...

Mentalitie are evolving people care, make stuff available, errors are found and fixed* Resources and Acknowledgments

Toward Reproducible Computer Science Experiments ?

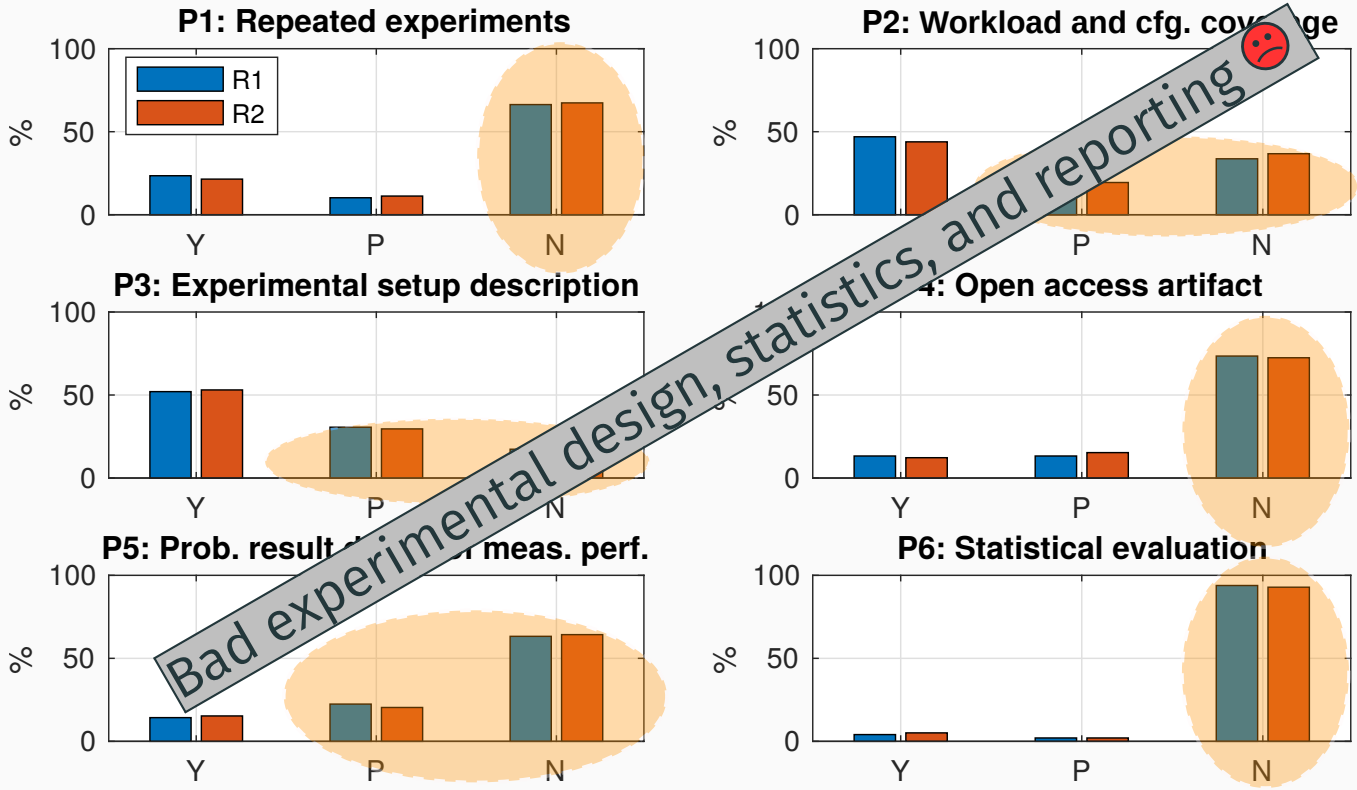
Key Concerns for our Community (Room for Improvement)

How are cloud performance currently obtained and reported?, *Methodological Principles for Reproducible Performance Evaluation in Cloud Computing*, IEEE Trans. on Soft. Eng., July 2019



Key Concerns for our Community (Room for Improvement)

How are cloud performance currently obtained and reported?, *Methodological Principles for Reproducible Performance Evaluation in Cloud Computing*, IEEE Trans. on Soft. Eng., July 2019



Key DoE principles:

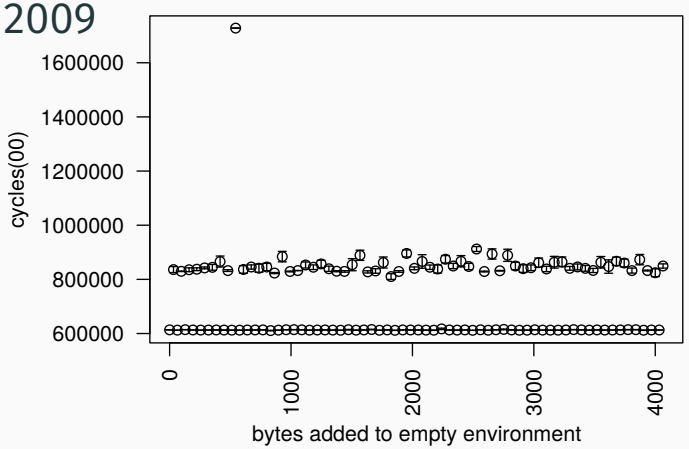
1. Replicate to increase reliability.
2. Randomize to reduce bias ⇝ Evaluate statistical confidence.

Measuring performance is difficult

Producing wrong data without doing anything obviously wrong!

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

changing the size of environment variables can trigger performance degradation as high as 300%; simply changing the link order of object files can cause performance to decrease by as much as 57%.

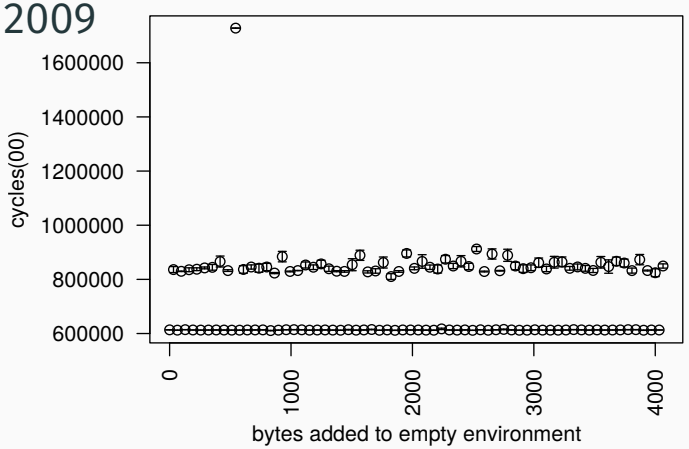


Measuring performance is difficult

Producing wrong data without doing anything obviously wrong!

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

changing the size of environment variables can trigger performance degradation as high as 300%; simply changing the link order of object files can cause performance to decrease by as much as 57%.



Taming the Influence of Memory Layout.

Stabilizer: Statistically Sound Performance Evaluation, C. Curtsinger and E. Berger in ASPLOS 2013

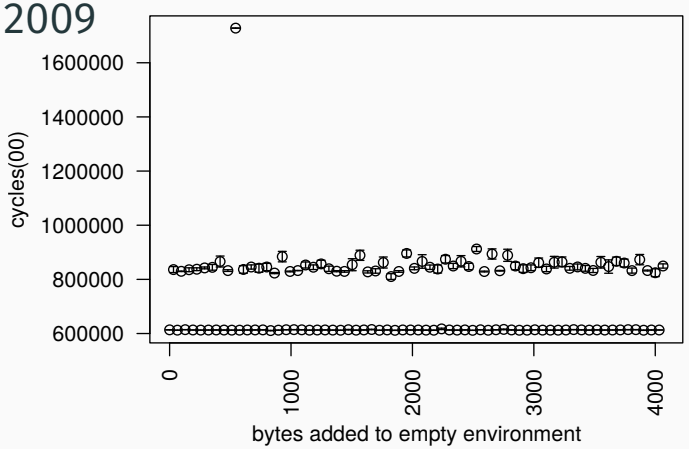
> *Stabilizer forces executions to sample the space of memory configurations by repeatedly rerandomizing layouts of code, stack, and heap objects at run-time. [..] Re-randomization ensures that layout effects follow a Gaussian distribution, enabling the use of statistical tests like ANOVA.*

Measuring performance is difficult

Producing wrong data without doing anything obviously wrong!

Mytkowicz et al. in ACM SIGPLAN Not. 44(3), March 2009

changing the size of environment variables can trigger performance degradation as high as 300%; simply changing the link order of object files can cause performance to decrease by as much as 57%.



Taming the Influence of Memory Layout.

Stabilizer: Statistically Sound Performance Evaluation, C. Curtsinger and E. Berger in ASPLOS 2013

> *Stabilizer forces executions to sample the space of memory configurations by repeatedly rerandomizing layouts of code, stack, and heap objects at runtime. [..] Re-randomization ensures that layout effects follow a Gaussian distribution, enabling the use of statistical tests like ANOVA.*

Randomization helps fighting bias incured by:

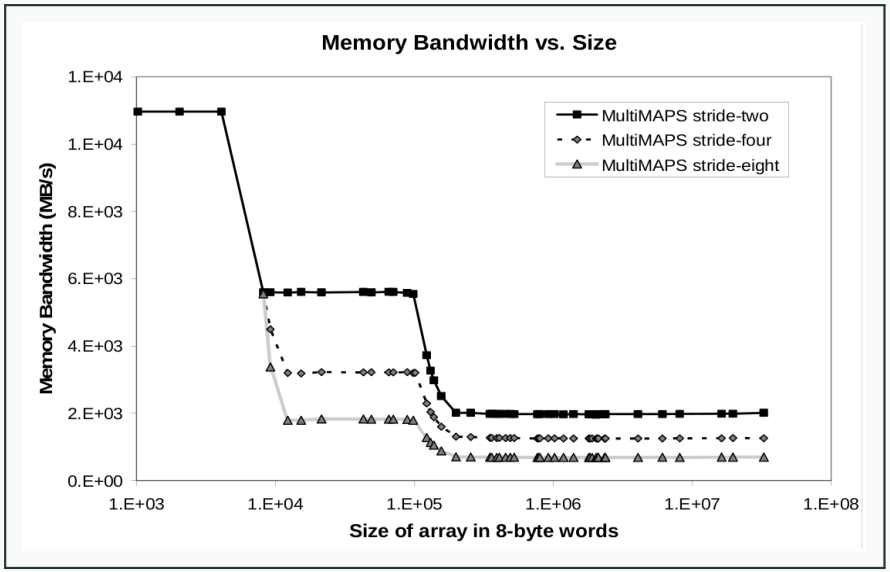
1. specific configurations $AA\ldots A \rightarrow A_1A_2\ldots A_n$ (~~pseudo-replication~~)
2. temporary perturbations $AA\ldots A\,BB\ldots B \rightarrow ABBAAAB\ldots$

Impact of Working Set Size on Effective Bandwdith

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level $\Rightarrow$ operate at the corresponding bandwidth
- Stride access decrease bandwidth

Impact of Working Set Size on Effective Bandwdith

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level $\Rightarrow$ operate at the corresponding bandwidth
- Stride access decrease bandwidth



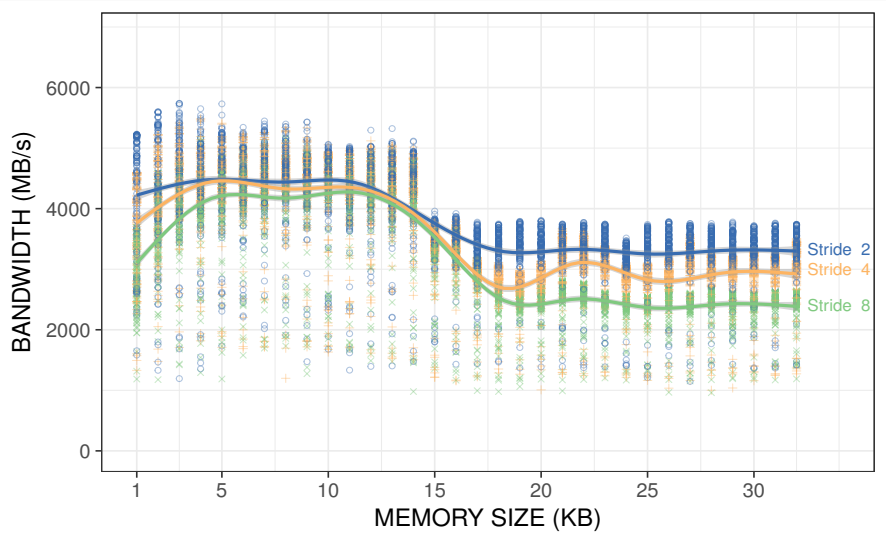
MultiMAPS on an Opteron

Genetic Algorithms Approach to Modeling the Performance of Memory-bound Computations, Tikir et. al. in SC'07

```
MultiMAPS(size, stride, nloops) {
  allocate buffer[size];
  timer_start();
  for rep in (1..nloops)
    for i in (0..size/stride)
      access buffer[stride*i];
  timer_stop();
  bandwidth = nb_access /
              elapsed_time;
  deallocate buffer;
}
```

Impact of Working Set Size on Effective Bandwdith

- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level ⇒ operate at the corresponding bandwidth
- Stride access decrease bandwidth

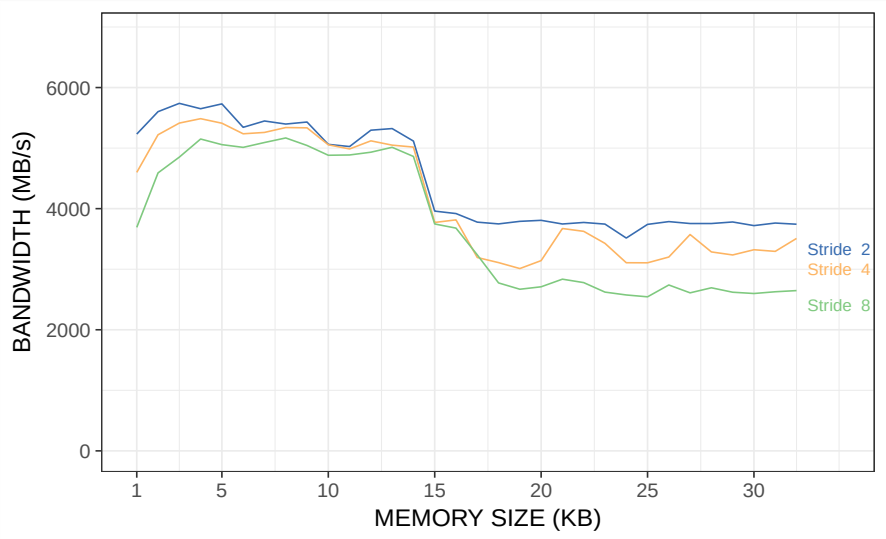


Our first attempt on a Pentium 4...

```
MultiMAPS(size, stride, nloops) {
  allocate buffer[size];
  timer_start();
  for rep in (1..nloops)
    for i in (0..size/stride)
       access buffer[stride*i];
  timer_stop();
  bandwidth = nb_access /
              elapsed_time;
  deallocate buffer;
}
```

Impact of Working Set Size on Effective Bandwdith

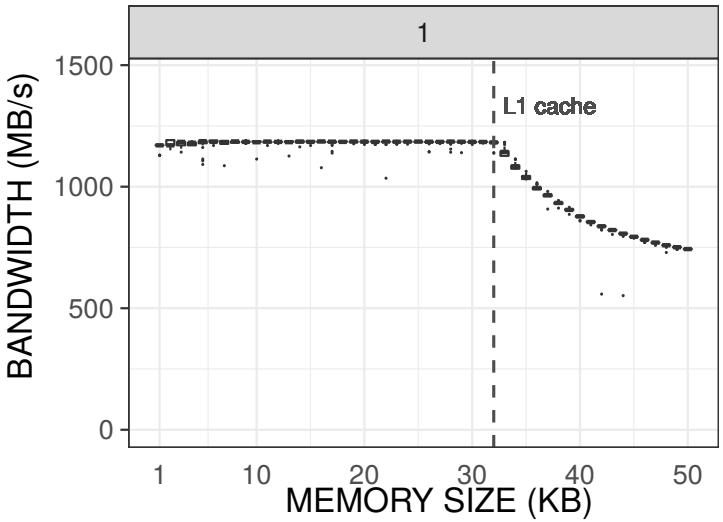
- Cache hierarchy (L1, L2, L3, RAM) with different bandwidth
- LRU, pre-fetching for linear access
- An array fits in a cache level ⇒ operate at the corresponding bandwidth
- Stride access decrease bandwidth



Our first attempt on a Pentium 4...

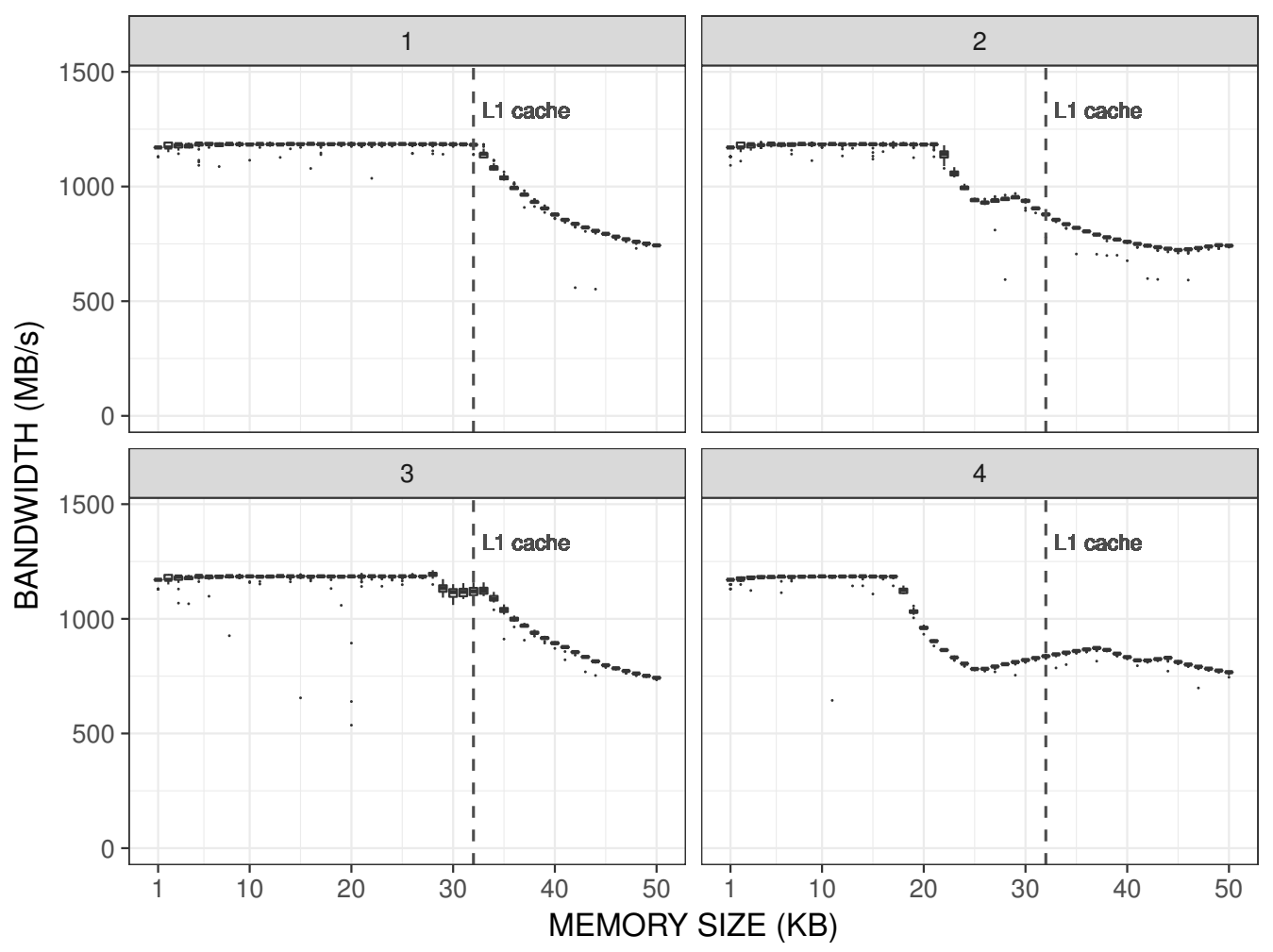
```
MultiMAPS(size, stride, nloops) {
  allocate buffer[size];
  timer_start();
  for rep in (1..nloops)
    for i in (0..size/stride)
       access buffer[stride*i];
  timer_stop();
  bandwidth = nb_access /
              elapsed_time;
  deallocate buffer;
}
```

Impact of Architecture



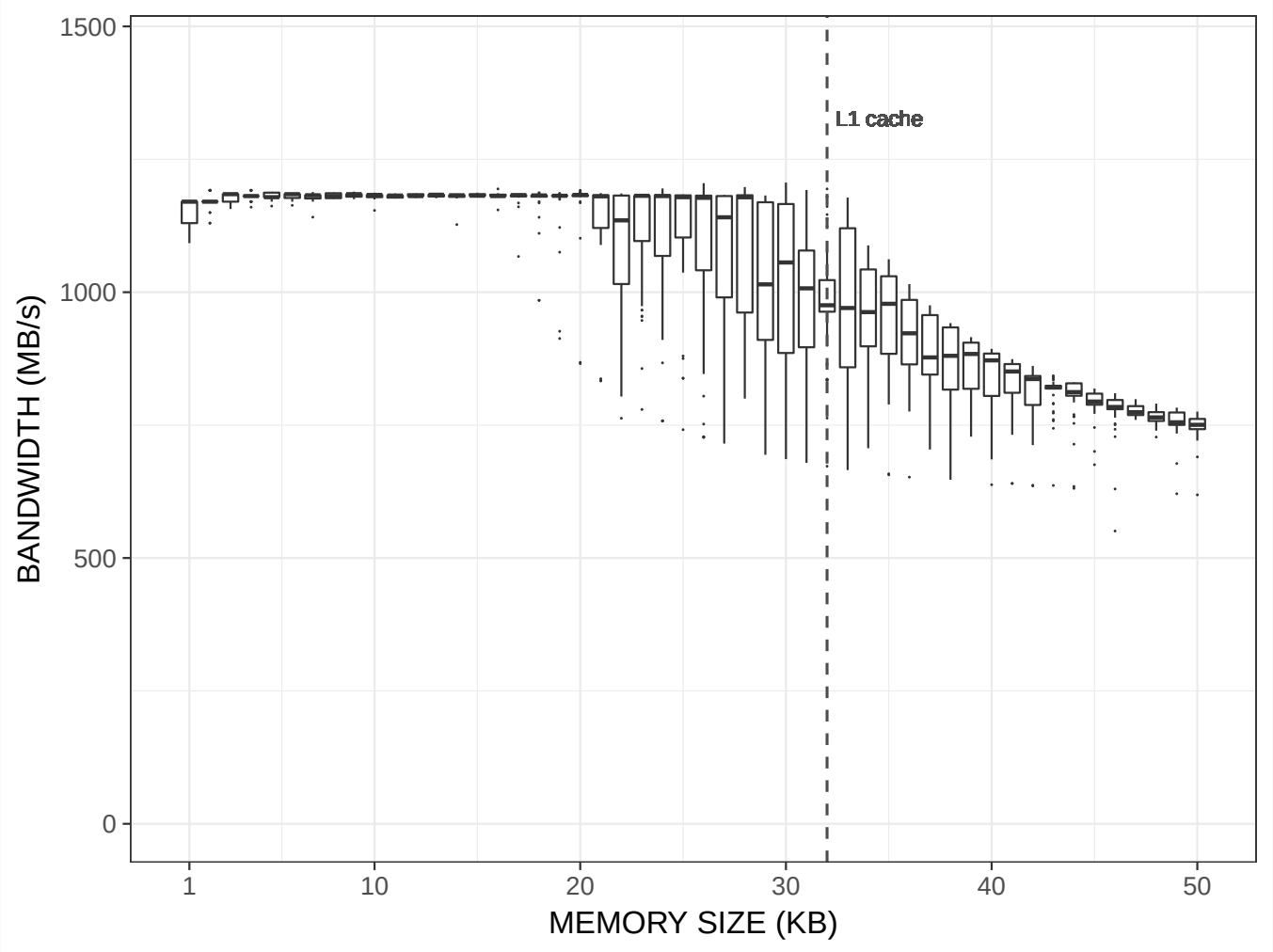
Courtesy of L. Stanisic

Impact of Architecture



Courtesy of L. Stanisic

Impact of Architecture (the ARM associativity issue)

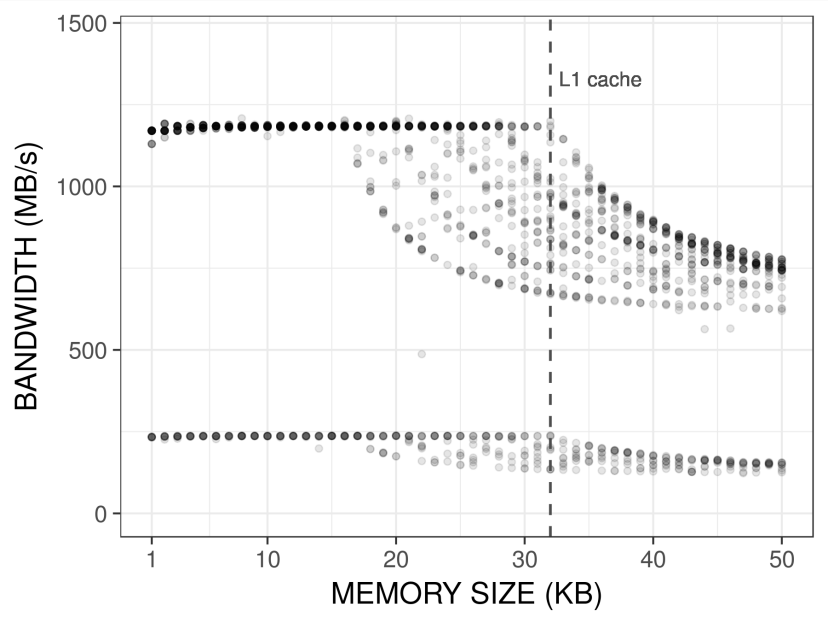


Randomize **physical address** start!

Courtesy of L. Stanisic

Impact of Operating System Scheduler

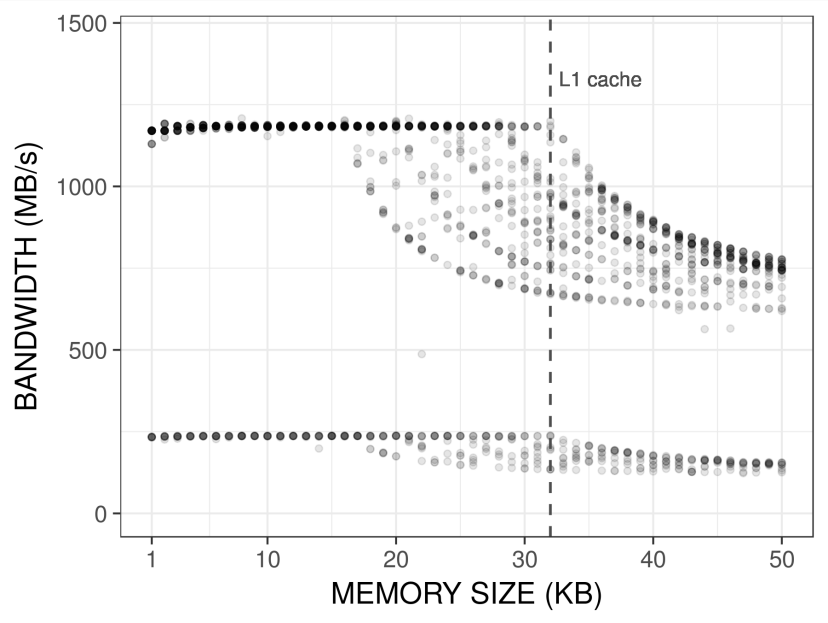
- Activating **real-time kernel scheduler**

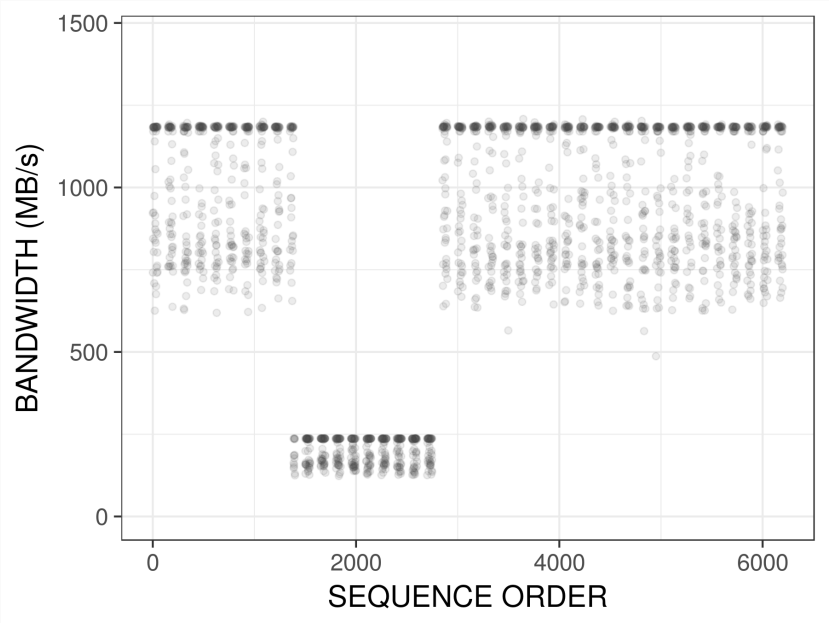


Courtesy of L. Stanisic

Impact of Operating System Scheduler

- Activating **real-time kernel scheduler**

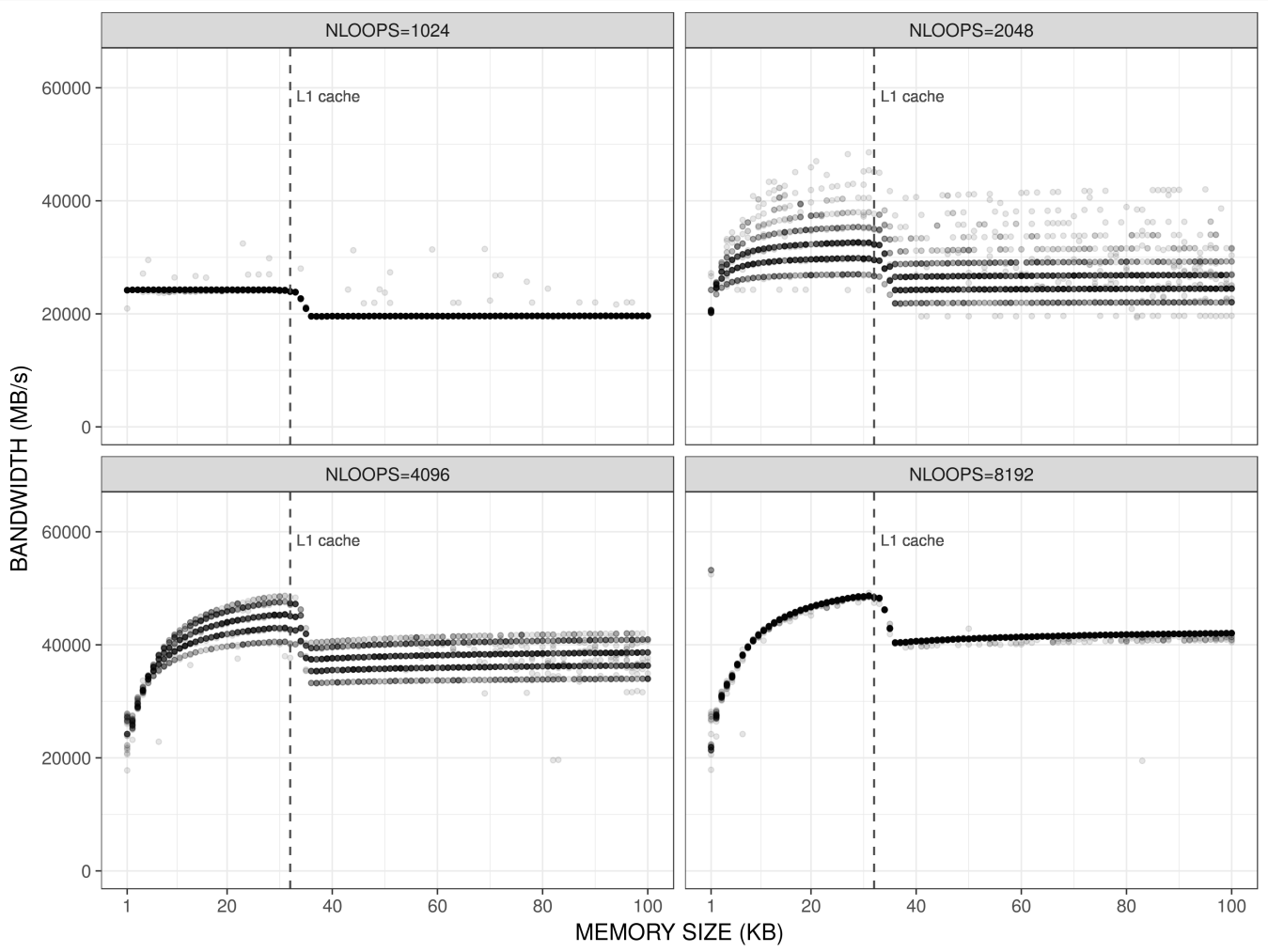




Courtesy of L. Stanisic

Impact of Repetitions

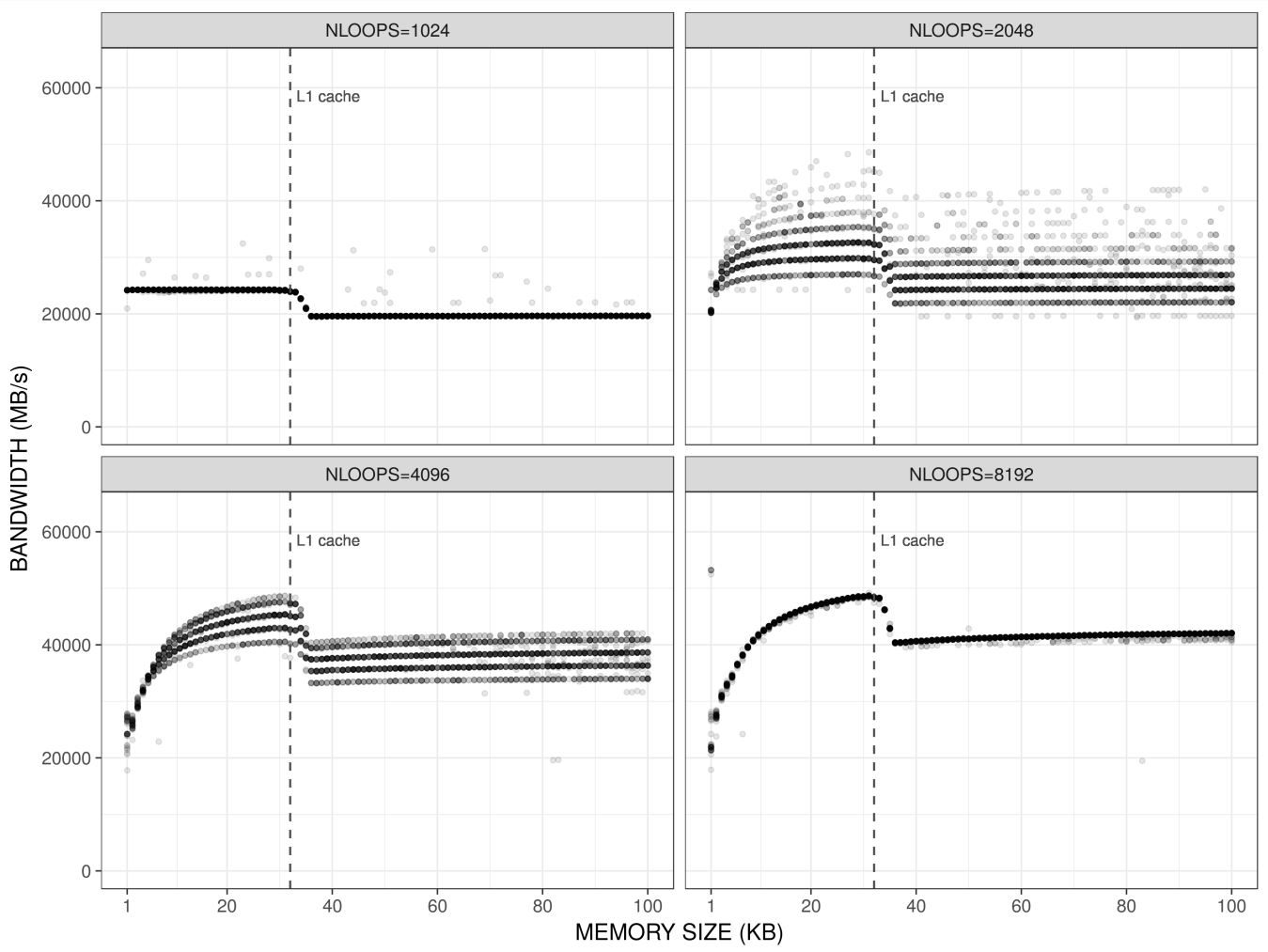
- Remember `nloops` ?



Courtesy of L. Stanisic

Impact of ~~Repetitions~~ DVFS

- Remember `nloops` ?



Courtesy of L. Stanisic

MPI Performance Characterization: The many biases

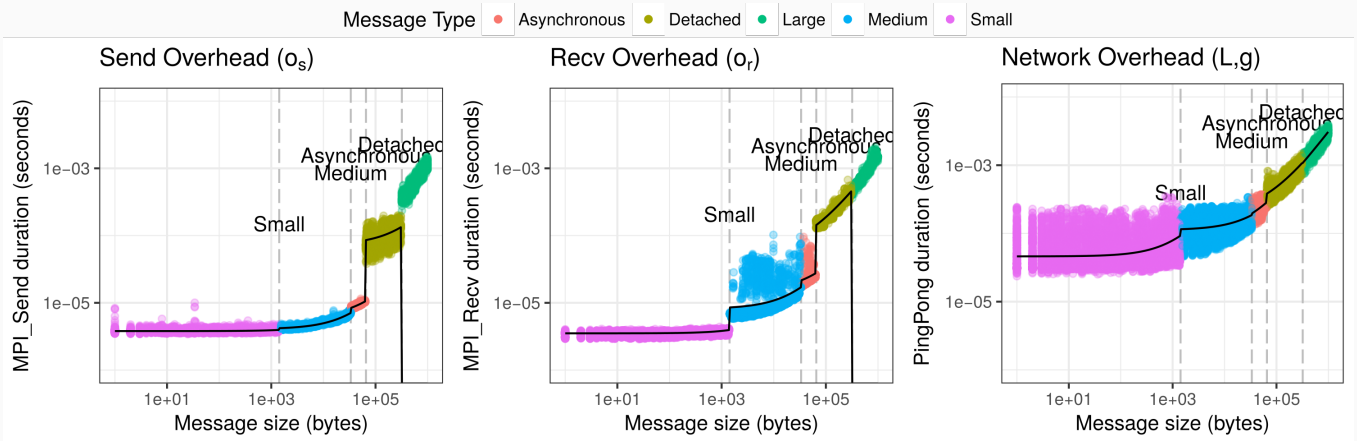
- Powers of two for message sizes ? Linear ?
- Sensitivity to temporal perturbations (in order `N_rep`)
- Breakpoint detection (increasing message size, minimal range length, ...)
- Outlier removal (assumes normality, ignores uncertainty of previous measurements)

Measurement proposal:

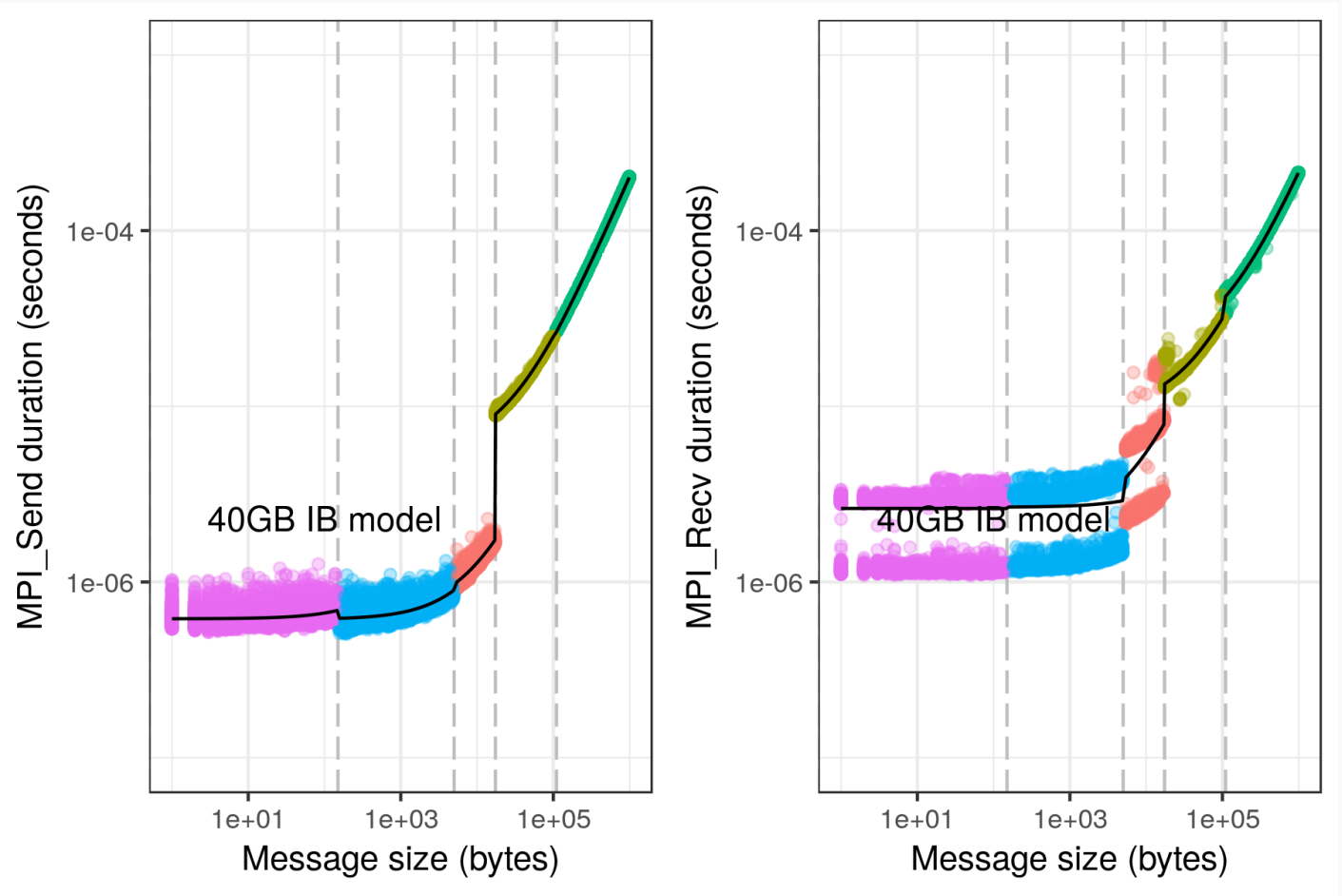
- Ping-Pong `data_size` $= 10^X$, where $X \sim \mathrm{Unif}(\log_{10}(a), \log_{10}(b))$.
- Record the time taken in every `MPI_Send` and `MPI_Receive` operation

Analysis proposal:

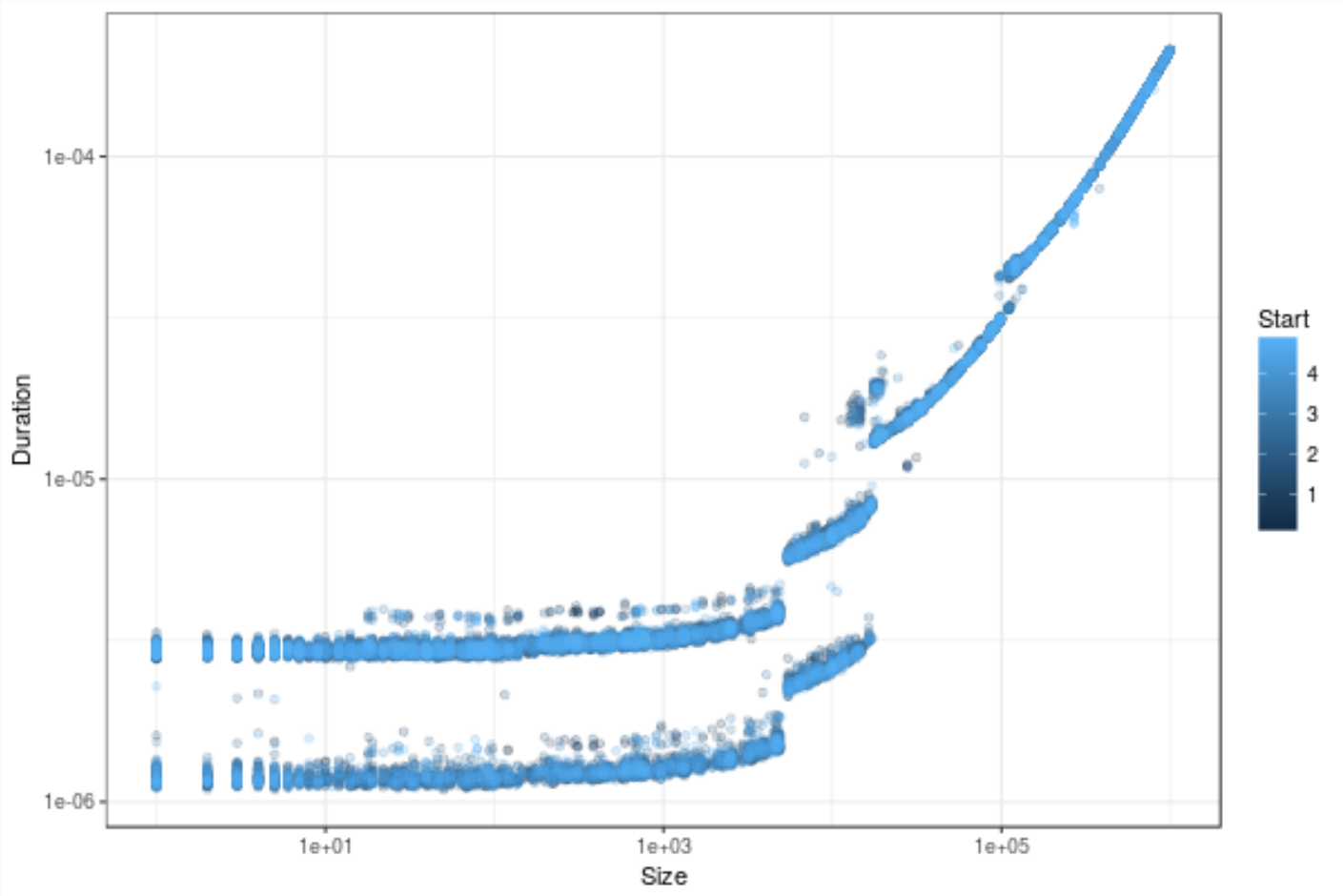
- Manually provided breakpoints
- Regression in R
- Inspect regression output and hypothesis (linearity, noise, "outliers")



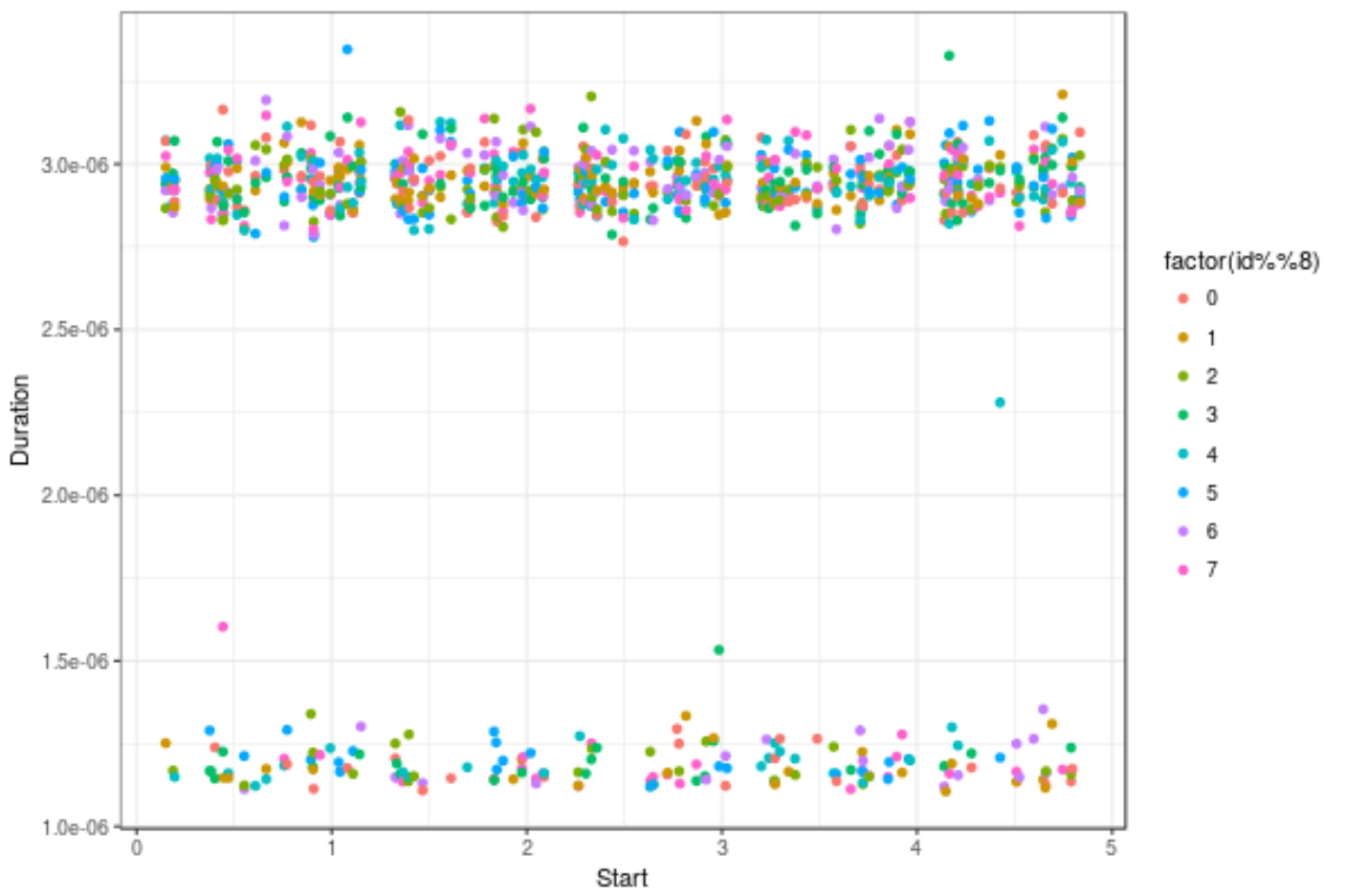
MPI Measurement: Randomization in action (Stampede@TACC)



MPI Measurement: Randomization in action (Stampede@TACC)

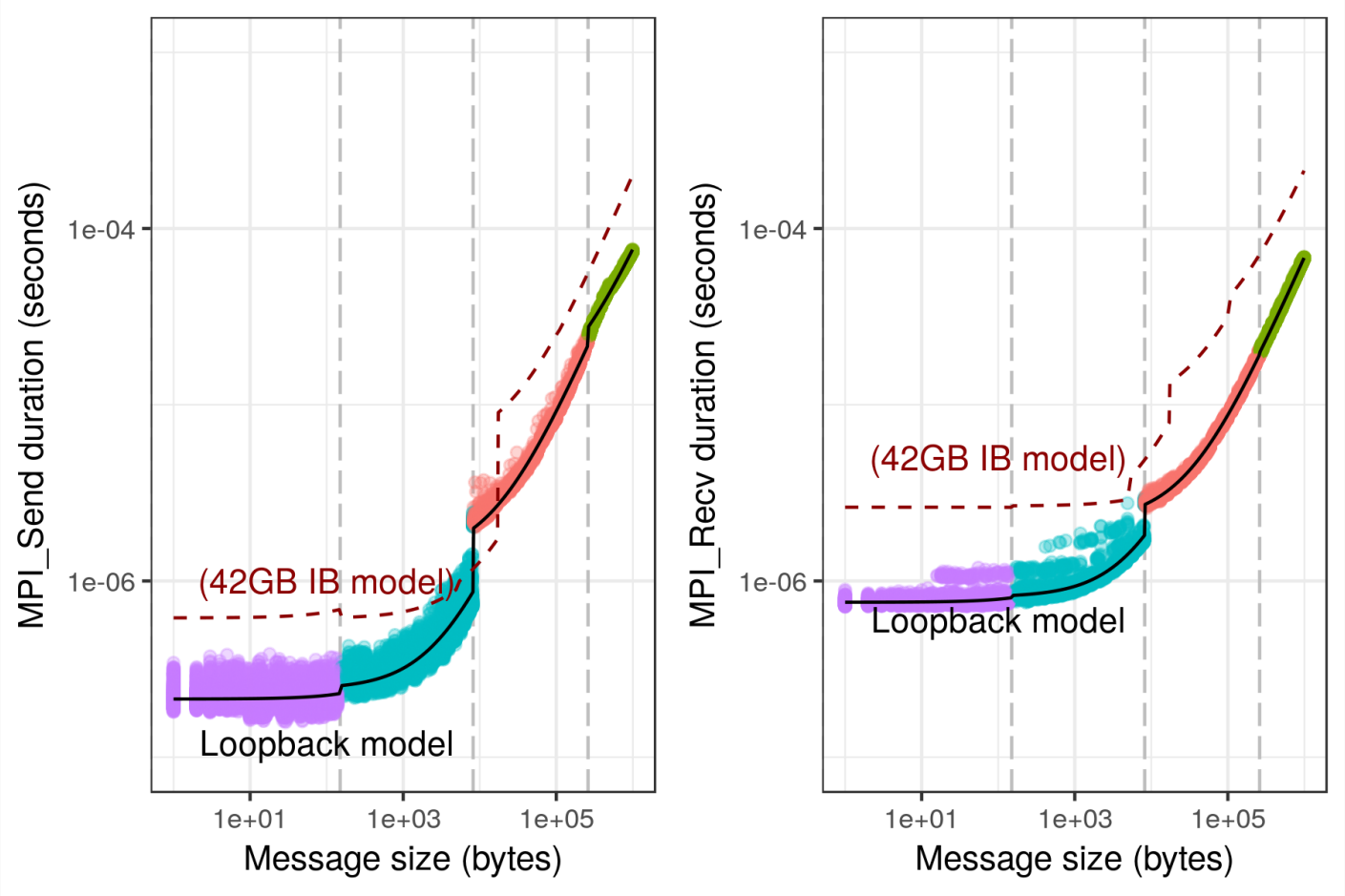


MPI Measurement: Randomization in action (Stampede@TACC)

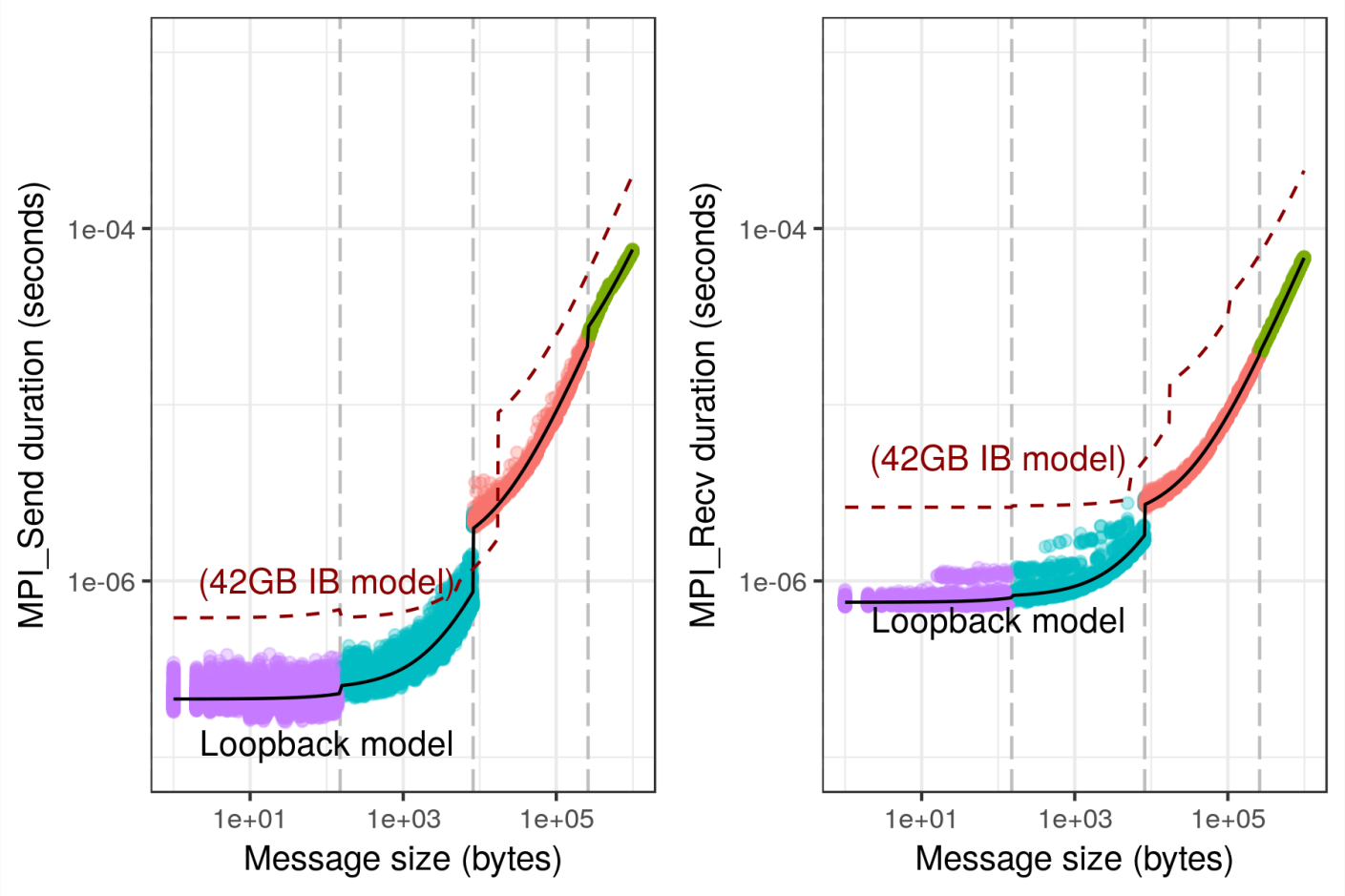


c558-[203,304]

MPI Measurement: Randomization in action (Stampede@TACC)



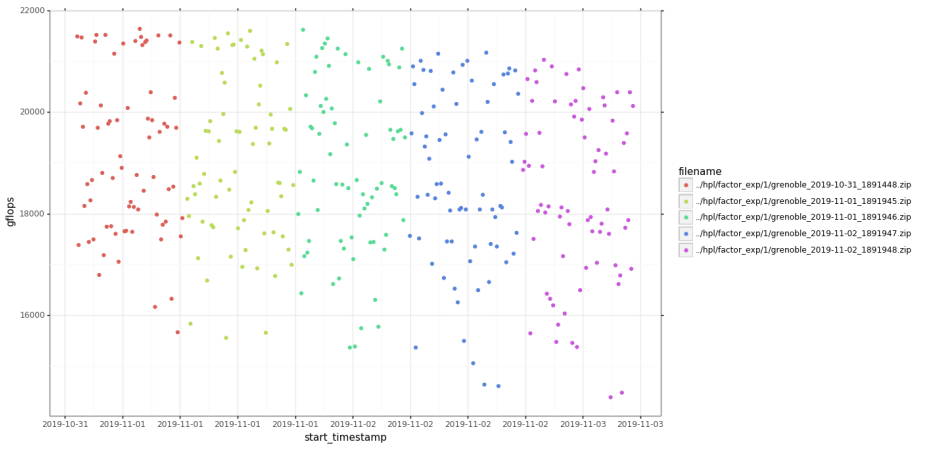
MPI Measurement: Randomization in action (Stampede@TACC)



Even the simpler physical quantities can be very tricky to measure because our models and protocols are often naive

Avoiding "Temporary" Perturbations (Randomizing a Factorial Design)

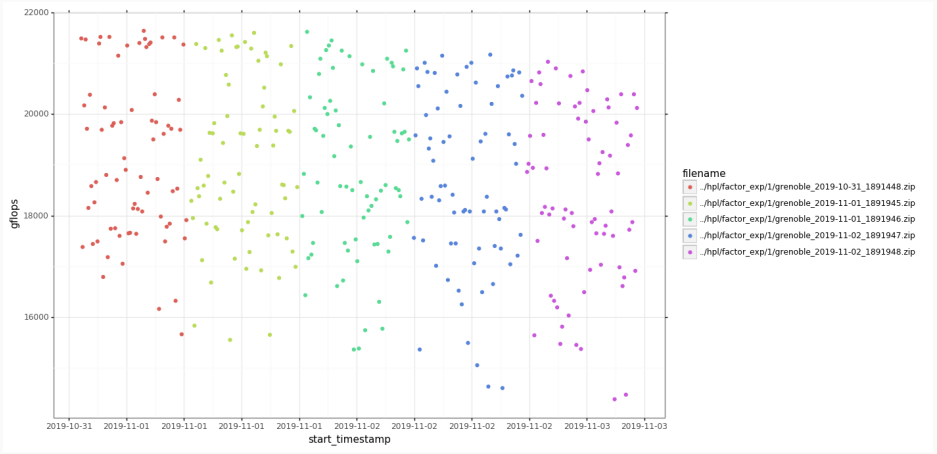
- HPL performance (32 nodes, 70 cfg., 5 repetitions)
- Time scale = 3 days

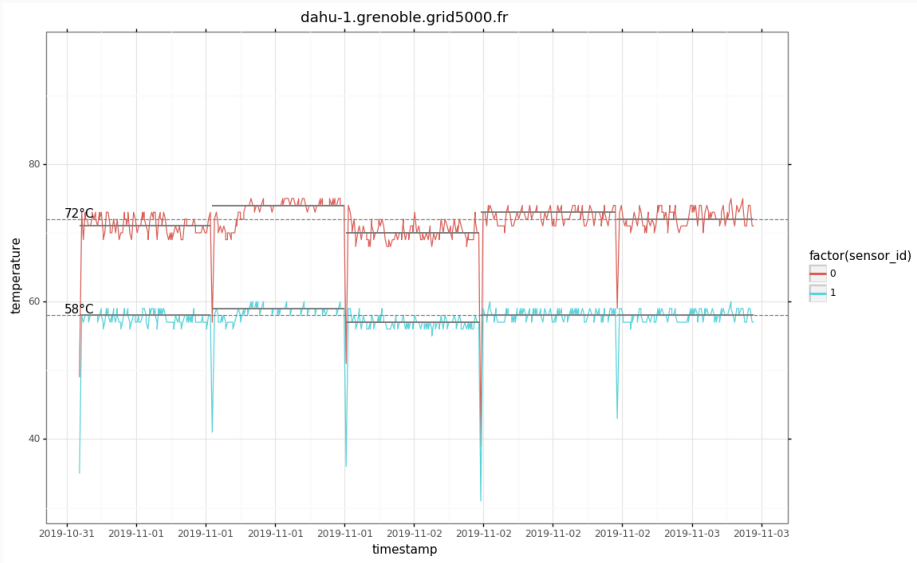


Courtesy of T. Cornebize

Avoiding "Temporary" Perturbations (Randomizing a Factorial Design)

- HPL performance (32 nodes, 70 cfg., 5 repetitions)
- Time scale = 3 days

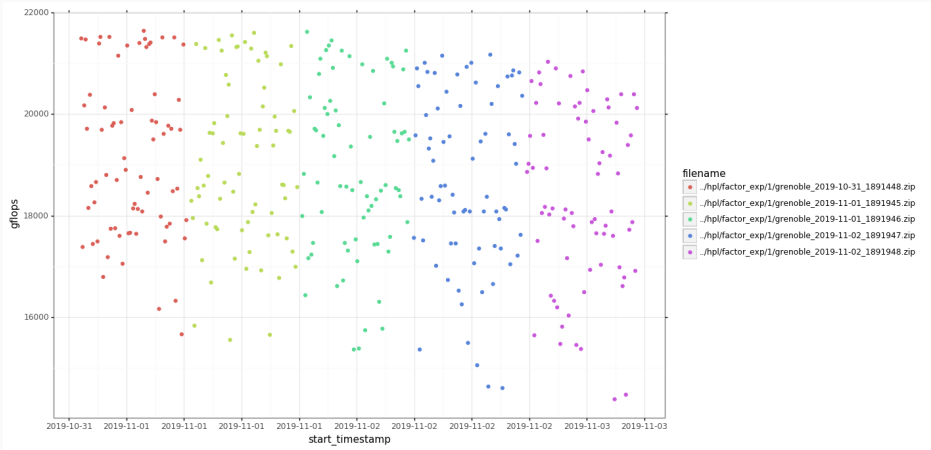


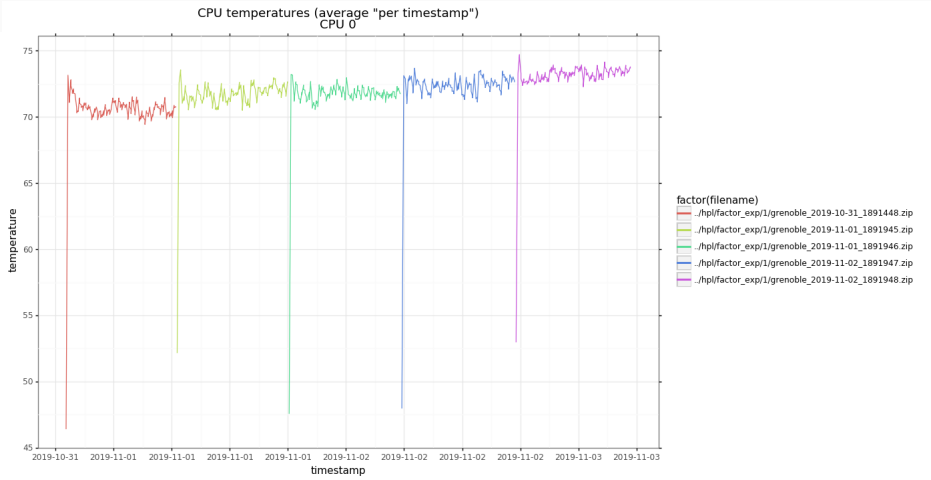


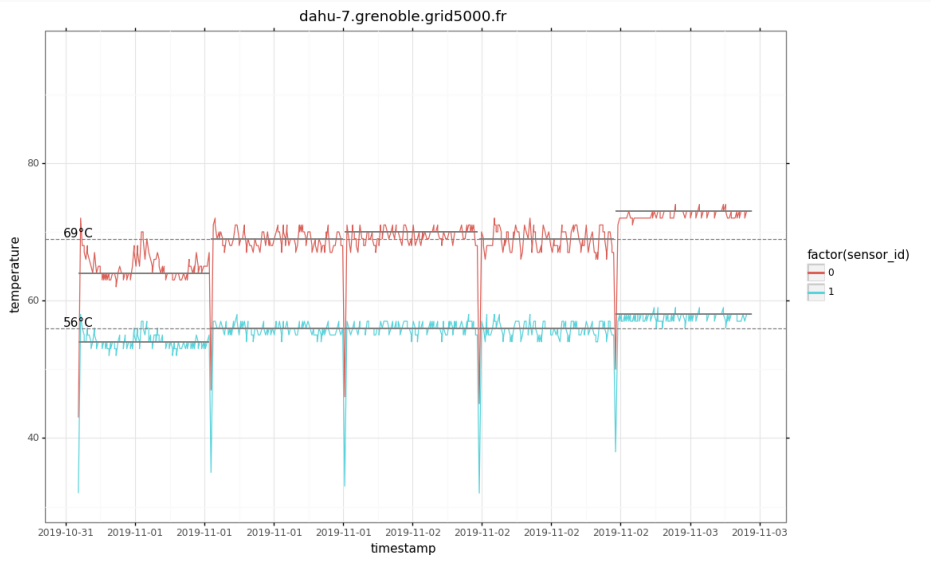
Courtesy of T. Cornebize

Avoiding "Temporary" Perturbations (Randomizing a Factorial Design)

- HPL performance (32 nodes, 70 cfg., 5 repetitions)
- Time scale = 3 days







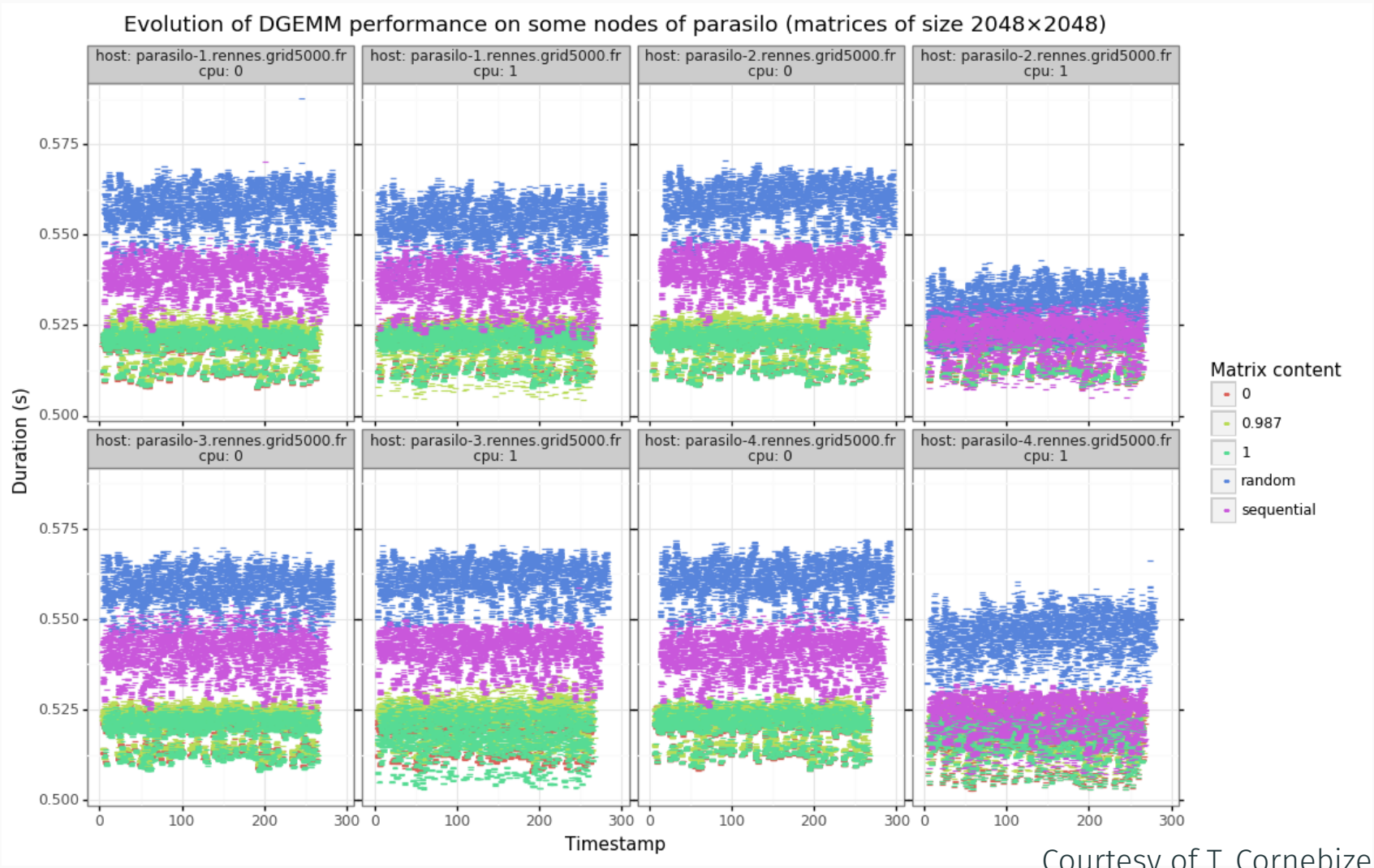
Courtesy of T. Cornebize

On the Importance of Content Initialization

- $C = A \times A$ (2048 $\times$ 2048), independant
- A initialized with $\boxed{0}$ $\boxed{1}$ $\boxed{0.987}$ $\boxed{1, 2, 3, \ldots}$ $\boxed{\textit{random}}$?
- Time scale = 5 minutes

On the Importance of Content Initialization

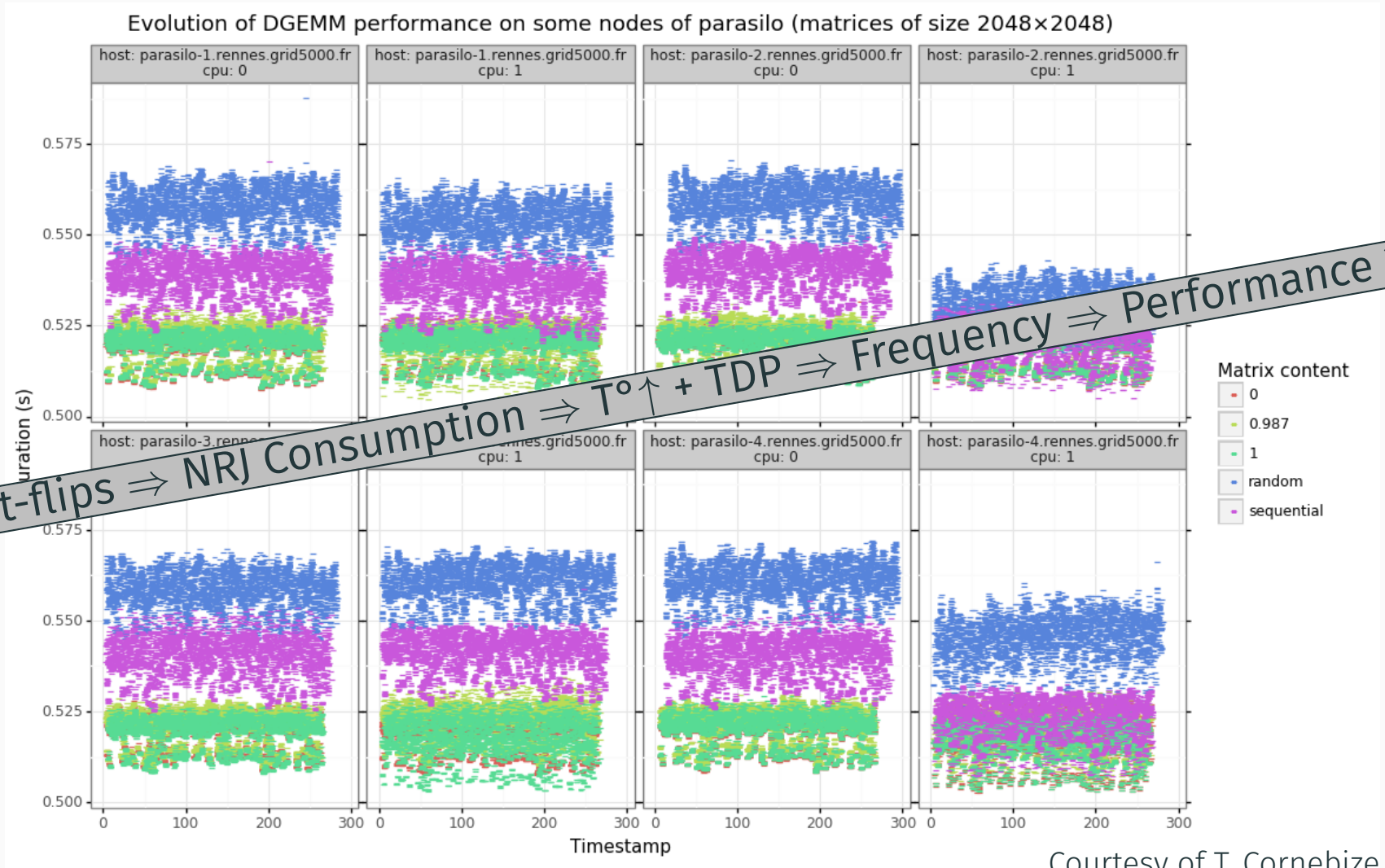
- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with $\boxed{0}$ $\boxed{1}$ $\boxed{0.987}$ $\boxed{1, 2, 3, \ldots}$ $\boxed{\textit{random}}$?



Courtesy of T. Cornebize

On the Importance of Content Initialization

- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with $\boxed{0}$ $\boxed{1}$ $\boxed{0.987}$ $\boxed{1, 2, 3, \ldots}$ $\boxed{\textit{random}}$?

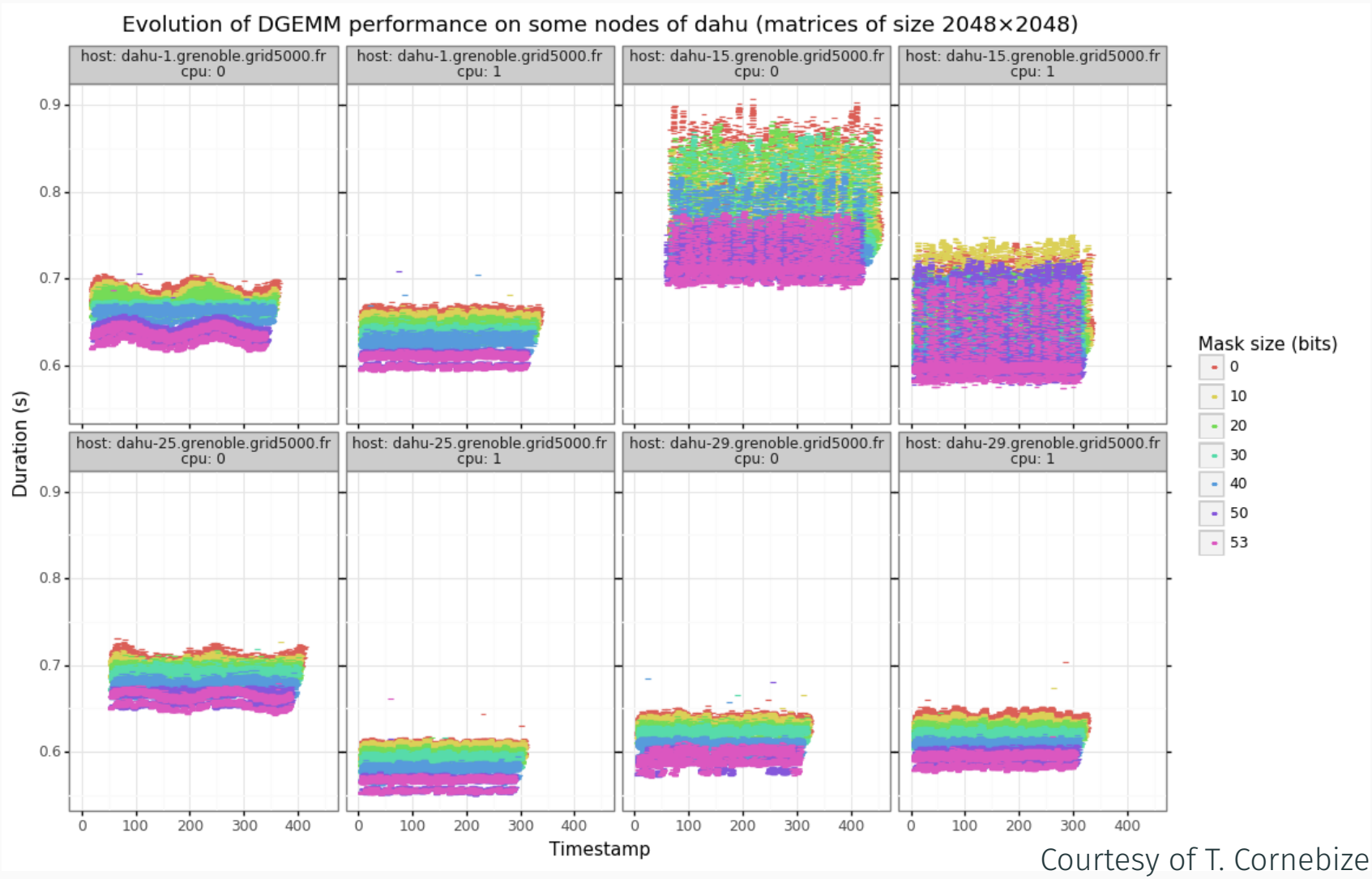


Bit-flips ⇒ NRJ Consumption ⇒ T°↑ + TDP ⇒ Frequency ⇒ Performance ?!?

Courtesy of T. Cornebize

On the Importance of Content Initialization

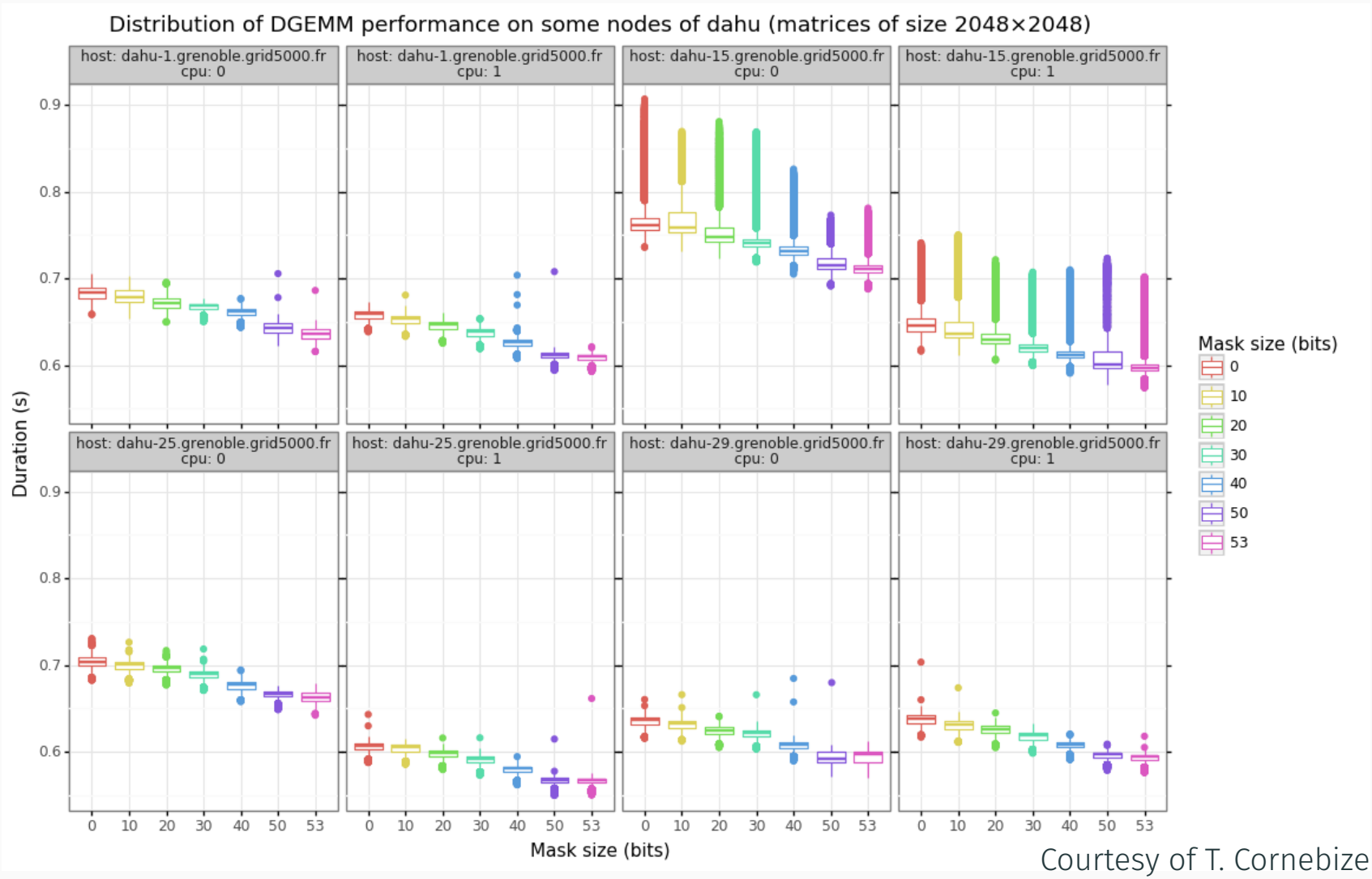
- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with [0] [1] [0.987] [$1, 2, 3, \ldots$] [*random*] ?



Courtesy of T. Cornebize

On the Importance of Content Initialization

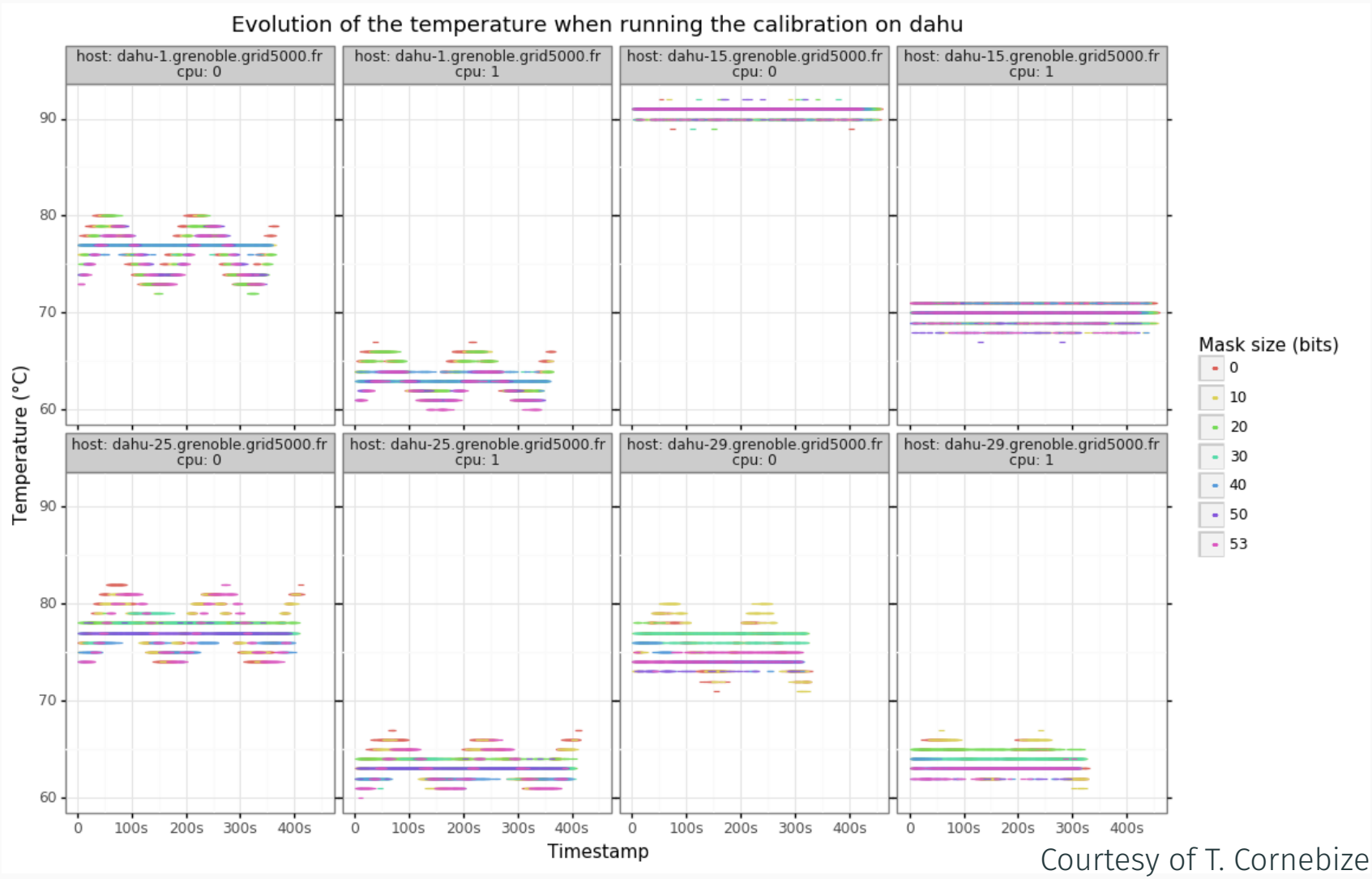
- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with 0 | 1 | 0.987 | $1, 2, 3, \ldots$ | *random* ?



Courtesy of T. Cornebize

On the Importance of Content Initialization

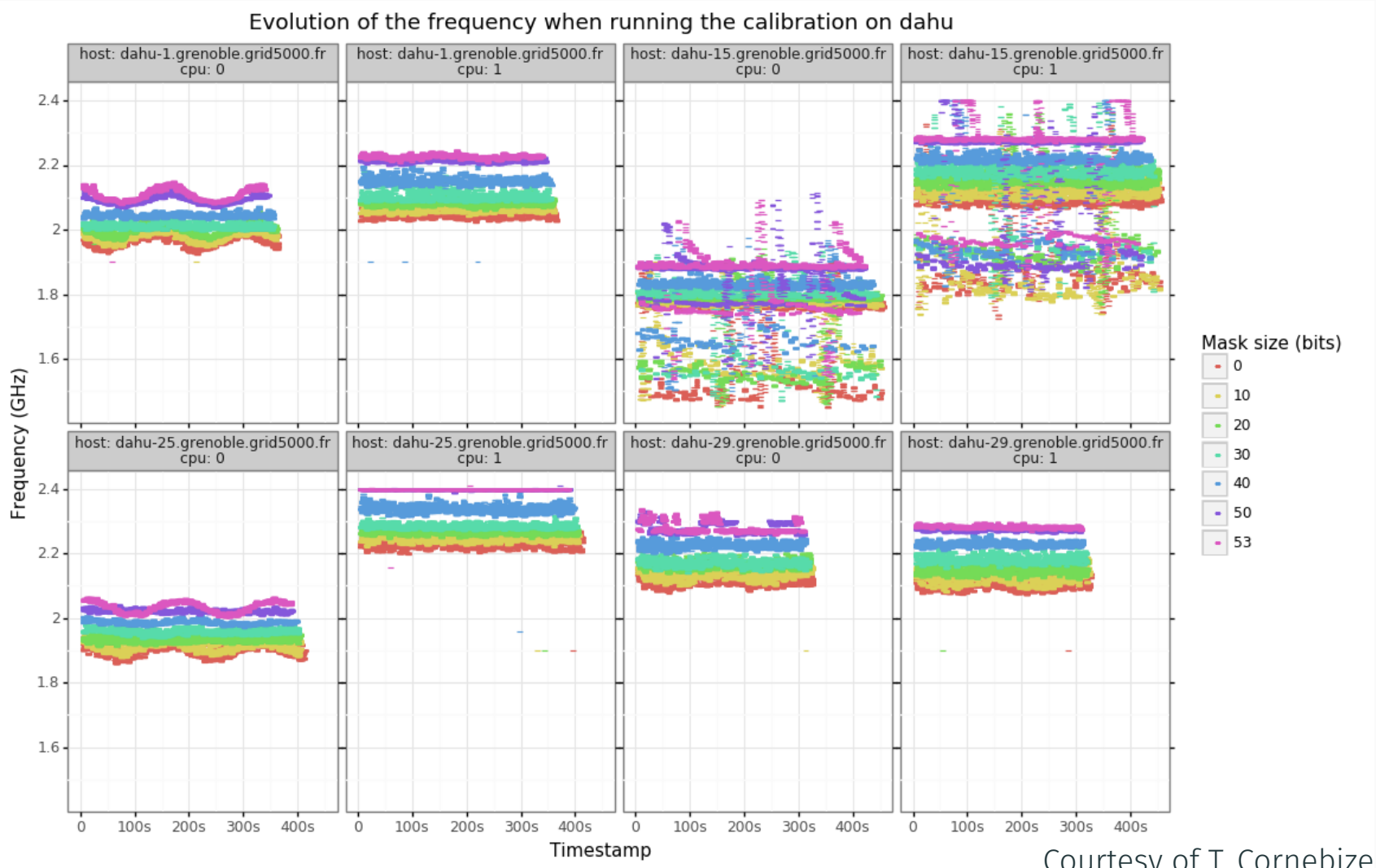
- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with $\boxed{0}$ $\boxed{1}$ $\boxed{0.987}$ $\boxed{1, 2, 3, \ldots}$ $\boxed{\textit{random}}$?



Courtesy of T. Cornebize

On the Importance of Content Initialization

- $C = A \times A$ (2048×2048), independant
- Time scale = 5 minutes
- A initialized with [0] [1] [0.987] [$1, 2, 3, \ldots$] [*random*] ?



Courtesy of T. Cornebize

Conclusion

Reproducible Research = Rigor and Transparency

To err is human.

Good research requires time and resources

1. **Train yourself and your students**: RR, statistics, experiments
   - Beware of checklists and norms
   - Understand what's at stake
2. **Change the norm:** make publication practices evolve
3. **Incentive**: consider RR/open science when hiring/promoting

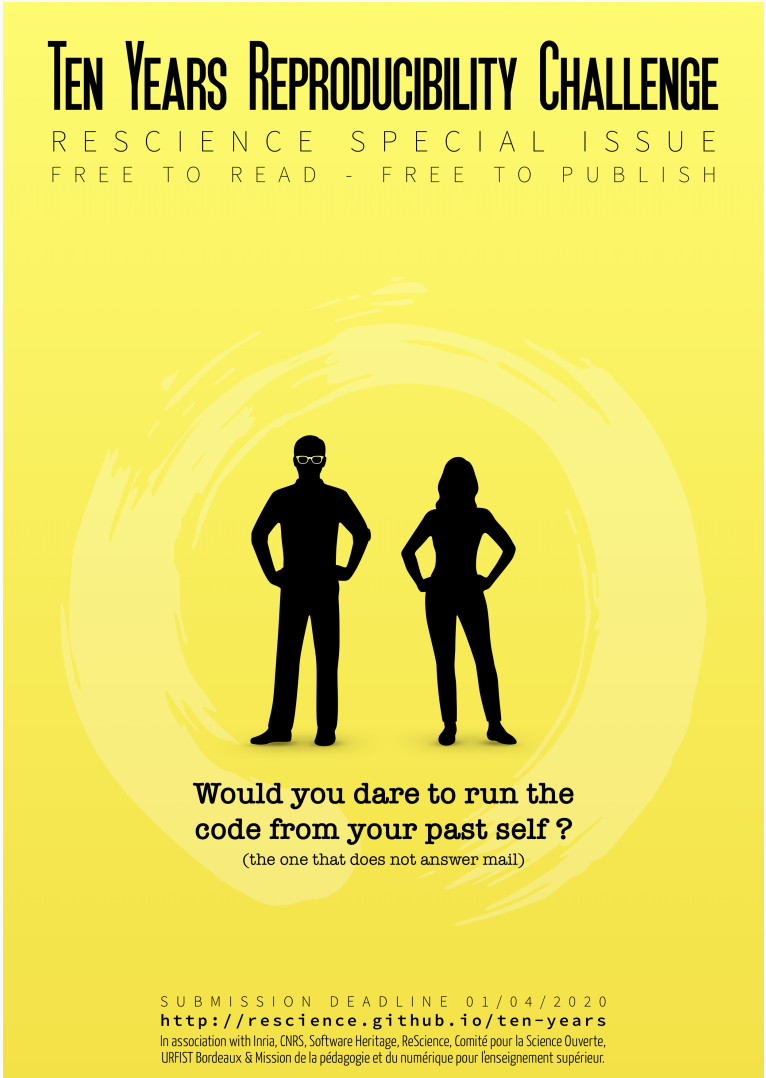
Reproducible Research = Rigor and Transparency

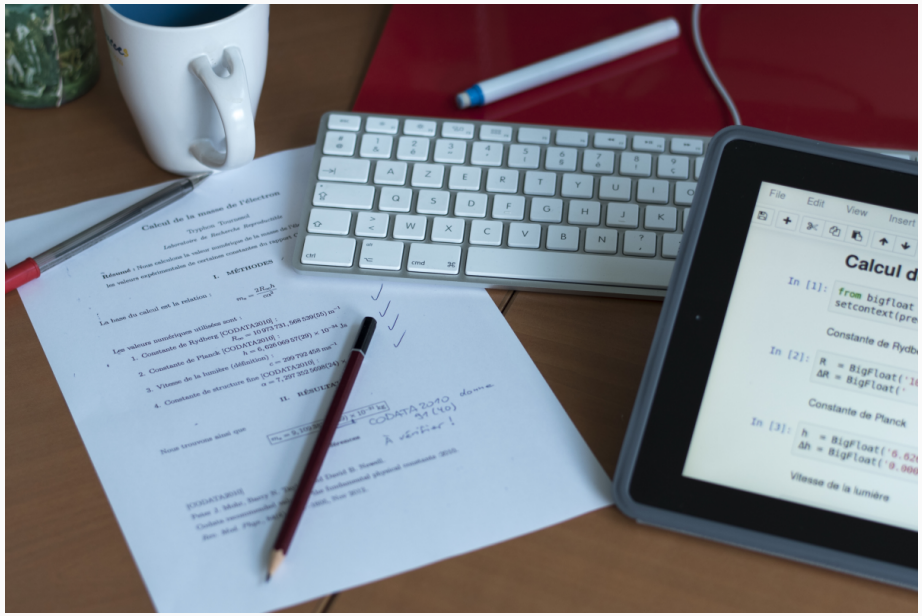
To err is human.

Good research requires time and resources

1. **Train yourself and your students**: RR, statistics, experiments
   - Beware of checklists and norms
   - Understand what's at stake
2. **Change the norm:** make publication practices evolve
3. **Incentive**: consider RR/open science when hiring/promoting
4. **Prepare the Future:** Toward literate experimentation?
   - Reuse, reuse, reuse!
   - Shared and controled testbeds (e.g., Grid'5000/SILECS)
   - How to share Experiments ?

Some Advertising





3rd Edition: ongoing

A new MOOC: "Advanced RR" (2021)

- Software environment control
- Scientific workflow
- Managing data

http://rescience.github.io/ten-years/