Recherche reproductible:
bonnes pratiques pour une science transparente

Arnaud Legrand

Congrès de la ROADEF

Champs-sur-Marne, Février 2025

What's expected from Science?

Goal

- Describe and explain **Nature** (human organizations and artificial objects as well)

What's expected from Science?

Goal

- Describe and explain **Nature** (human organizations and artificial objects as well)
- Find solutions to **society** needs and issues
  - Feed technological development and industry: health, energy, communications, ...
  - Enlighten public decisions: educate citizens, politics, regulation, ...

What's expected from Science?

Goal

- Describe and explain **Nature** (human organizations and artificial objects as well)
- Find solutions to **society** needs and issues
  - Feed technological development and industry: health, energy, communications, ...
  - Enlighten public decisions: educate citizens, politics, regulation, ...

What else distinguishes science from other human activities?

What's expected from Science?

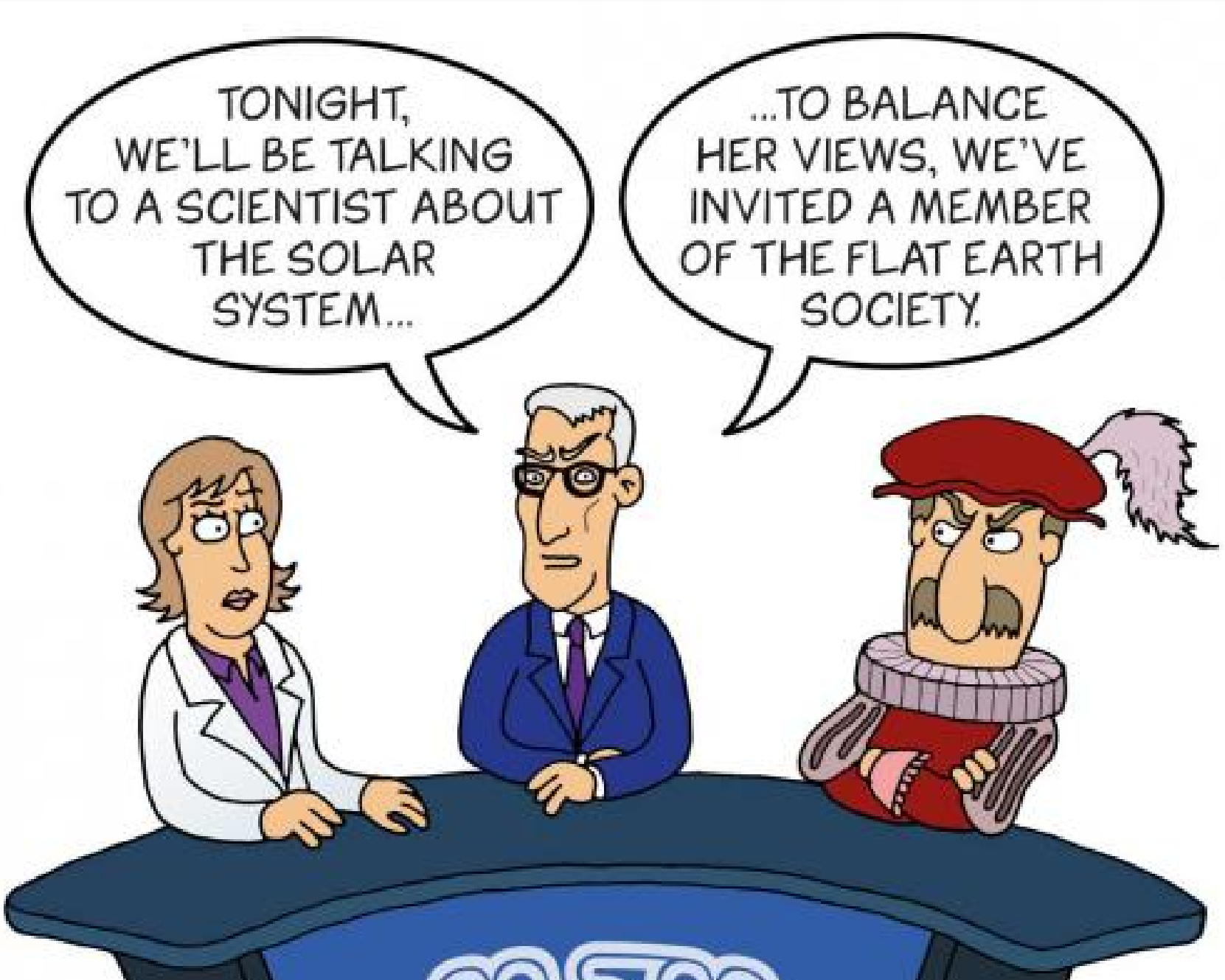
Goal

- Describe and explain **Nature** (human organizations and artificial objects as well)
- Find solutions to **society** needs and issues
  - Feed technological development and industry: health, energy, communications, ...
  - Enlighten public decisions: educate citizens, politics, regulation, ...

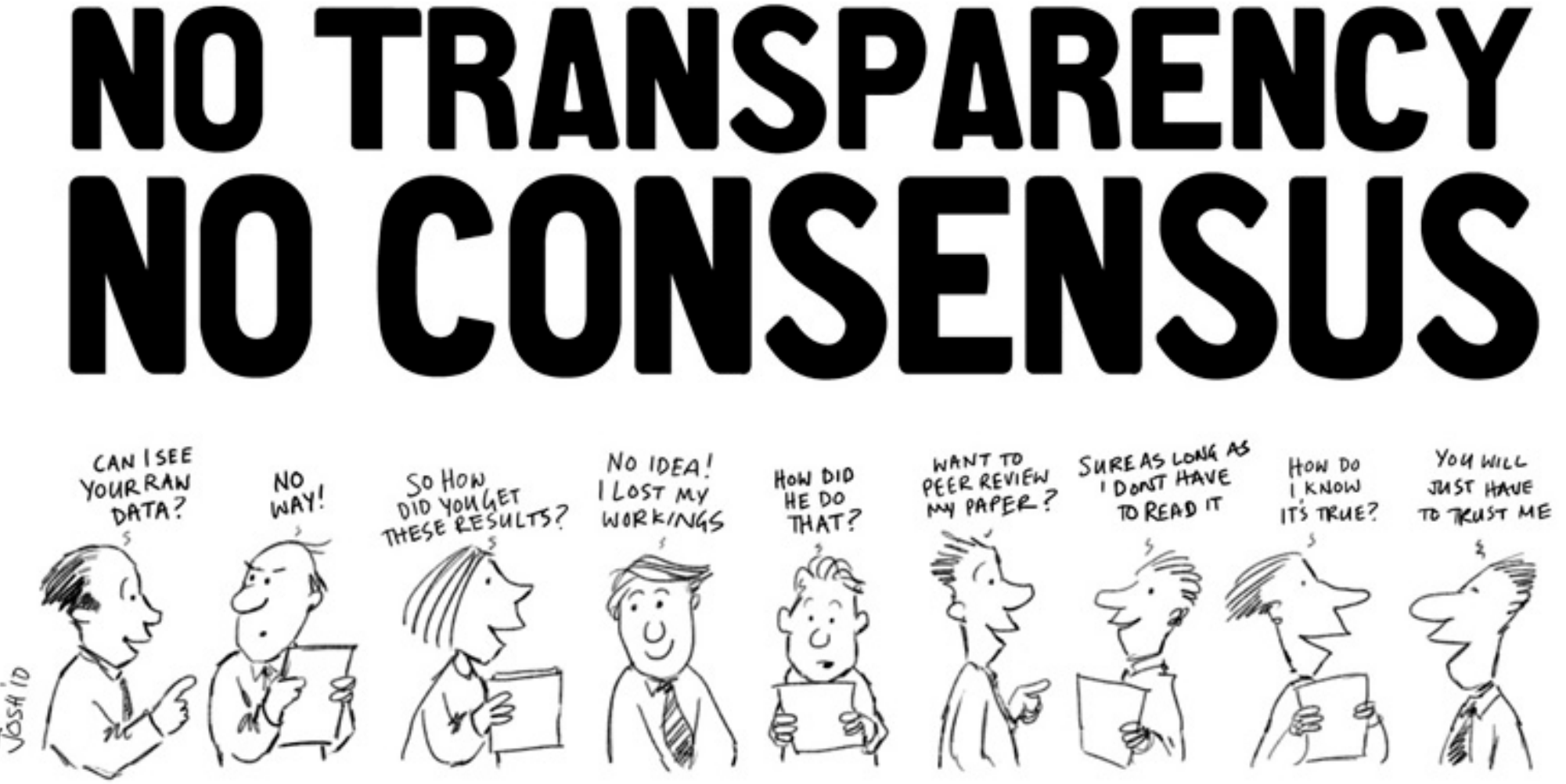
What else distinguishes science from other human activities?

Method

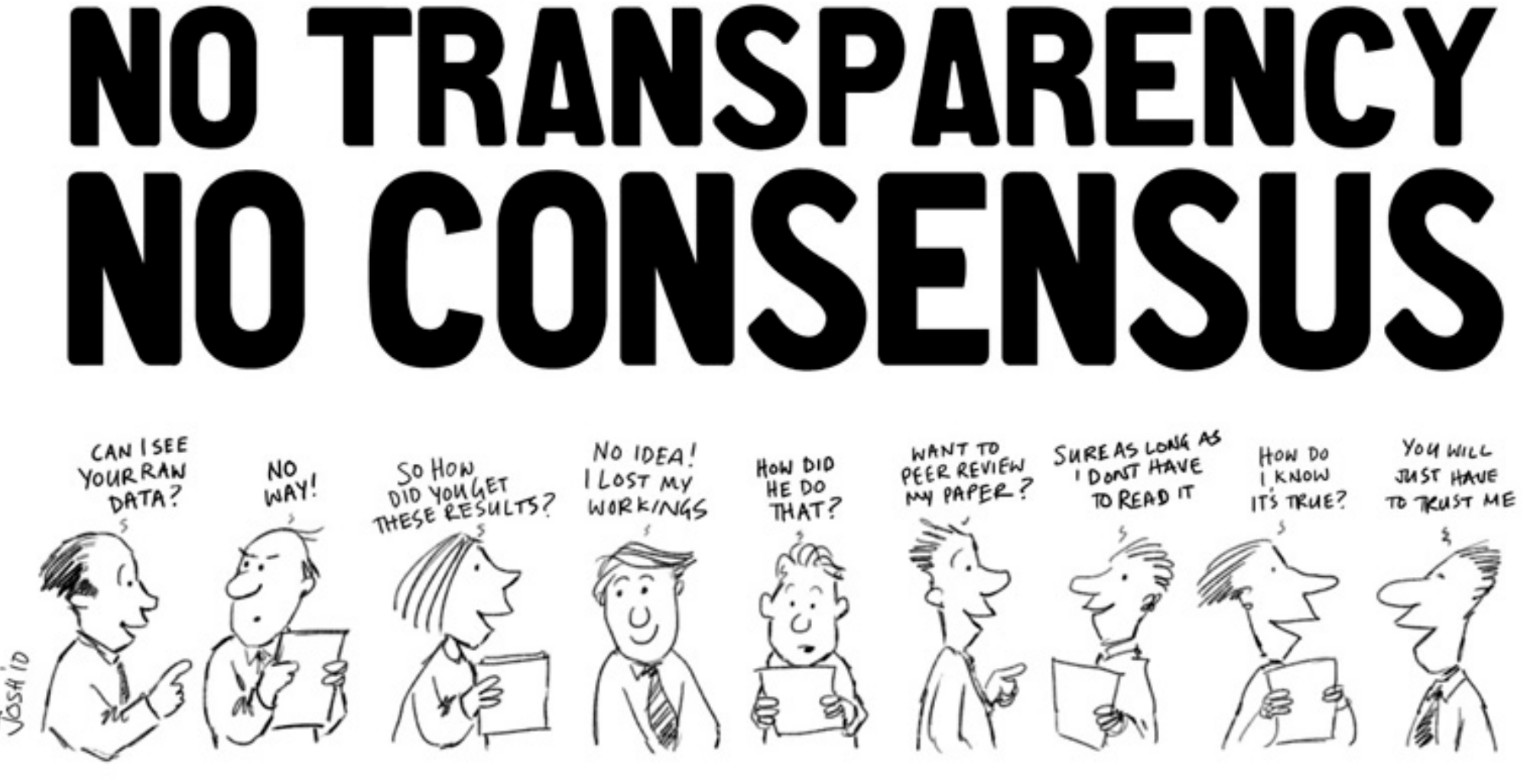
Scientific Consensus



Scientific Consensus



Scientific Consensus



Enlightening the society requires moral/methodological/technical warranties

Merton's four norms of science (1940)

Universalism Scientific validity is independent of the sociopolitical status/personal attributes of its participants (origin, gender, sexuality, religion, etc.)

Communism/communalism All scientists should have common ownership of scientific goods (intellectual property), to promote collective collaboration; secrecy is the opposite of this norm

- We say "Newton's law" to remember that Newton made the original discovery, but not because he has any property of this law

Disinterestedness Scientific institutions act for the benefit of a common scientific enterprise, rather than for the personal gain of individuals

Organized skepticism Scientific claims should be exposed to critical scrutiny (including reproduction) before being accepted

Merton's four norms of science (1940)

Universalism Scientific validity is independent of the sociopolitical status/personal attributes of its participants (origin, gender, sexuality, religion, etc.)

Communism/communalism All scientists should have common ownership of scientific goods (intellectual property), to promote collective collaboration; secrecy is the opposite of this norm

- We say "Newton's law" to remember that Newton made the original discovery, but not because he has any property of this law

Disinterestedness Scientific institutions act for the benefit of a common scientific enterprise, rather than for the personal gain of individuals

Organized skepticism Scientific claims should be exposed to critical scrutiny (including reproduction) before being accepted

Merton was not saying *"you should follow these norms"*, but rather

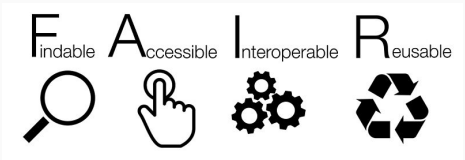
"this is the ~~behavior~~ belief I am observing in the scientific community"

Transparent Research ∼ Open Science ?

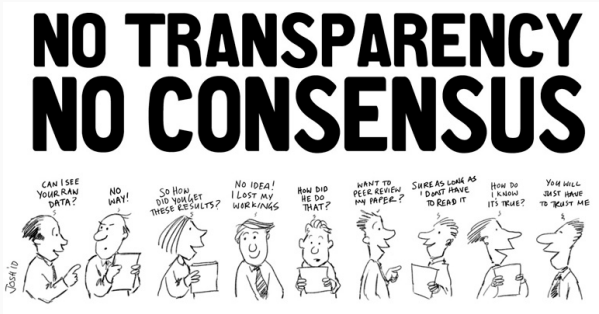
Plan National pour la Science Ouverte: CNRS, Inria, INRAE, *Citizen Science*

Main pillars:

1. Open access (started before 2000)
2. Open data
3. Open source
   - *Open hardware*
4. **Open methodology** (Reproducible Research)
   - *Open-notebook science*
   - *Open science infrastructures*
5. Open peer review (avoid collusion)
6. Open educational resources



CC-BY-SA by default at CNRS!)

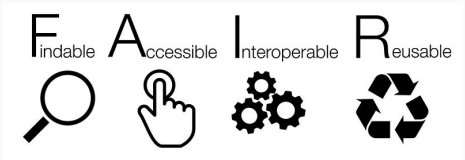


Transparent Research ~ Open Science ?

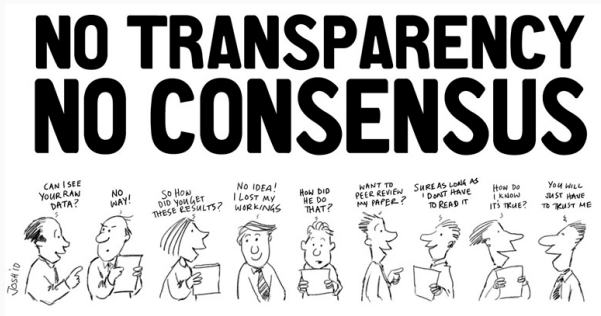
Plan National pour la Science Ouverte: CNRS, Inria, INRAE, *Citizen Science*

Main pillars:

1. Open access (started before 2000)
2. Open data
3. Open source
   - *Open hardware*
4. **Open methodology** (Reproducible Research)
   - *Open-notebook science*
   - *Open science infrastructures*
5. Open peer review (avoid collusion)
6. Open educational resources



CC-BY-SA by default at CNRS!)



Making code/data available for the reproduction of results from published papers has become the new norm

E.g., Artifact evaluation and ACM badges

Common Reproducibility Pitfalls

Go read the paper by Smith et. al. 2009

Access through **your institution** | Purchase PDF

Article preview

ELSEVIER

Future Generation Computer Systems

Volume 25, Issue 3, March 2009, Pages 315-325

Secure on-demand grid computing

M. Smith, M. Schmidt, N. Fallenbeck, T. Dörnemann, C. Schridde, B. Freisleben

Show more

+ Add to Mendeley | Share | Cite

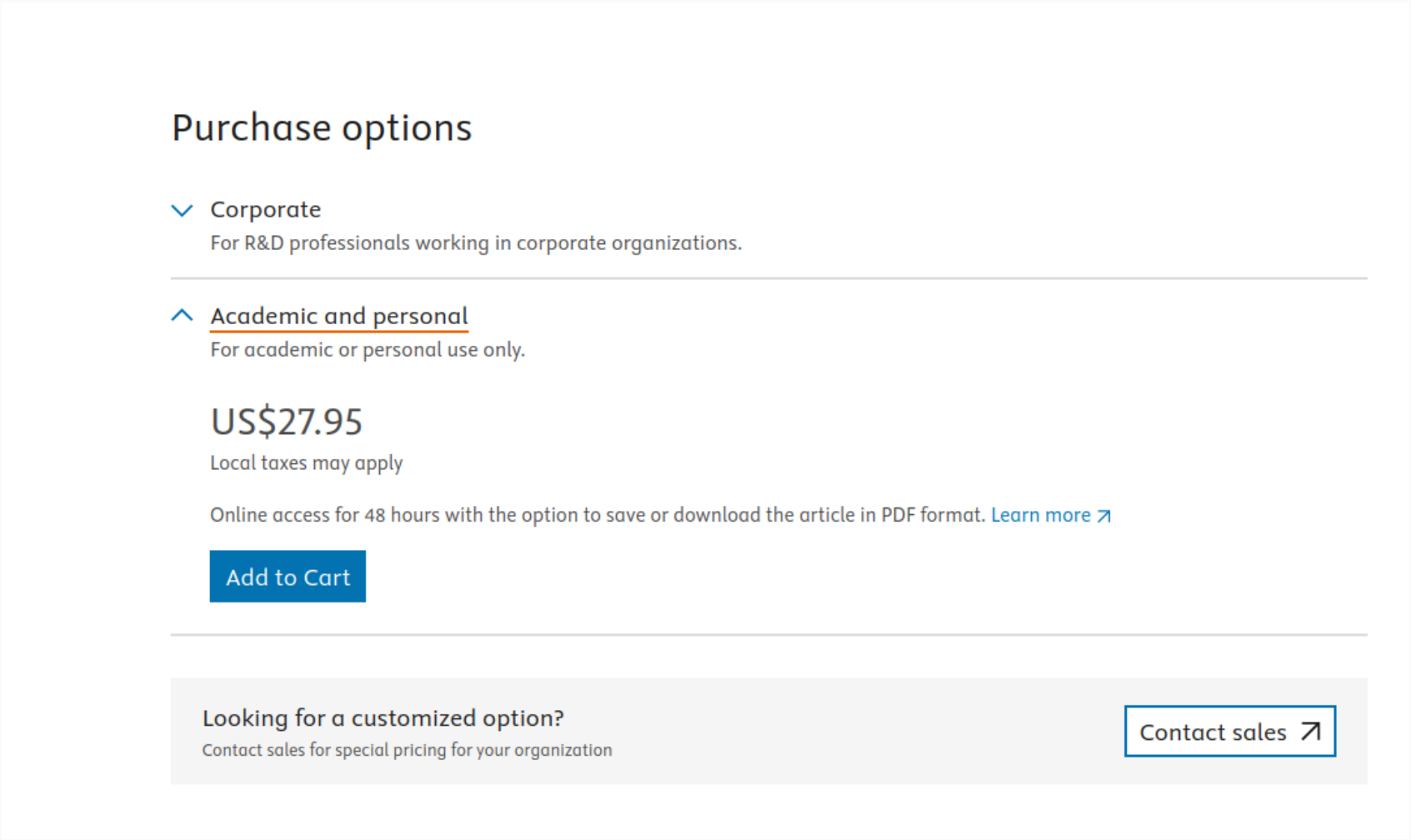
https://doi.org/10.1016/j.future.2008.03.002

Get rights and content

Abstract

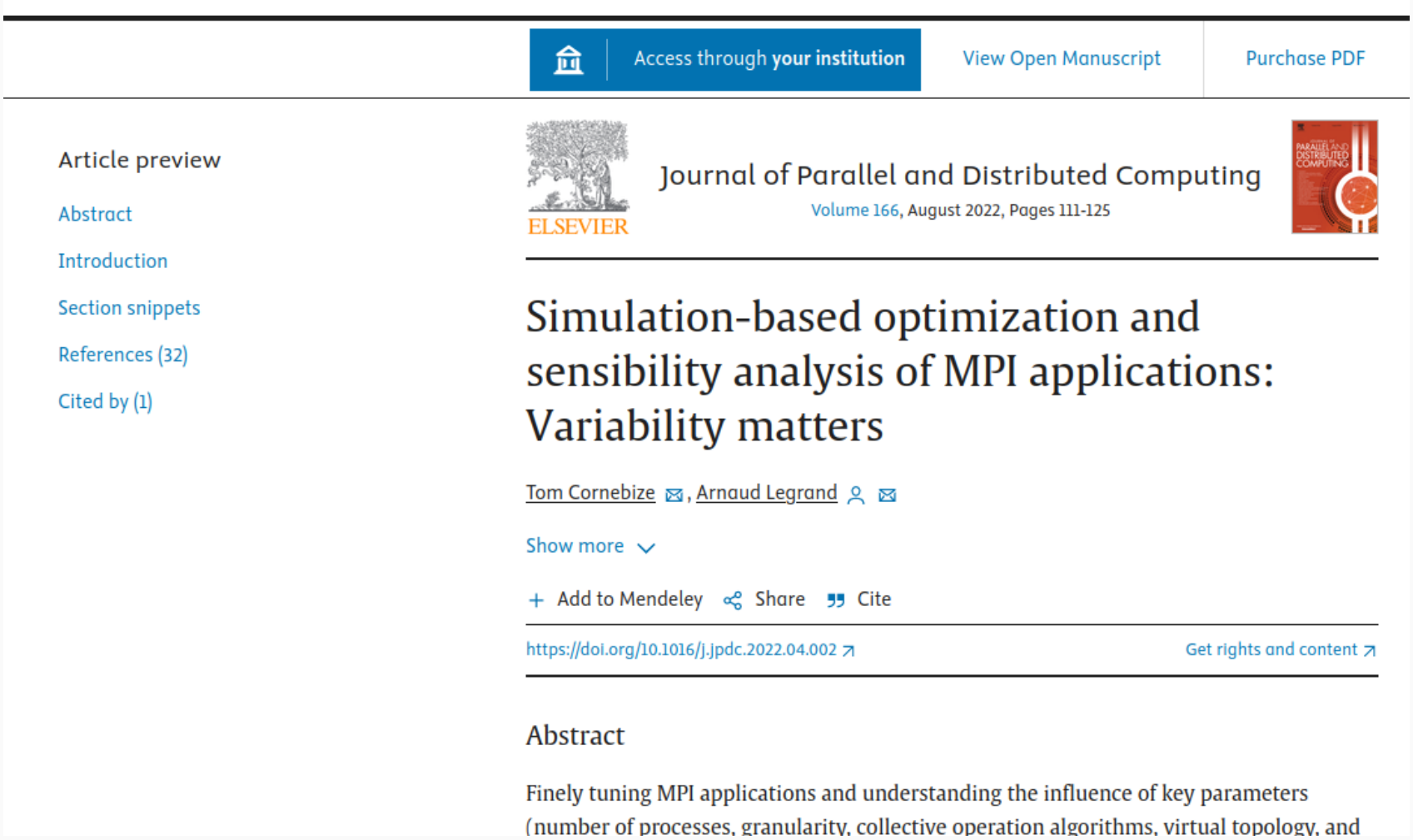
In this paper, a novel approach for enabling Grid users to autonomously install and use custom software on demand using an image creation station is presented, while at the same time offering new security mechanisms to protect both software and data from other Grid users and external attackers. An automated dynamic firewalling mechanism

Go read the paper by Smith et. al. 2009



- Use your institution subscription... or Sci-Hub 😉,

Go read the paper by Smith et. al. 2009



Access through your institution | View Open Manuscript | Purchase PDF

Article preview

Abstract

Introduction

Section snippets

References (32)

Cited by (1)

ELSEVIER

Journal of Parallel and Distributed Computing

Volume 166, August 2022, Pages 111-125

Simulation-based optimization and sensibility analysis of MPI applications: Variability matters

Tom Cornebize, Arnaud Legrand

Show more

+ Add to Mendeley | Share | Cite

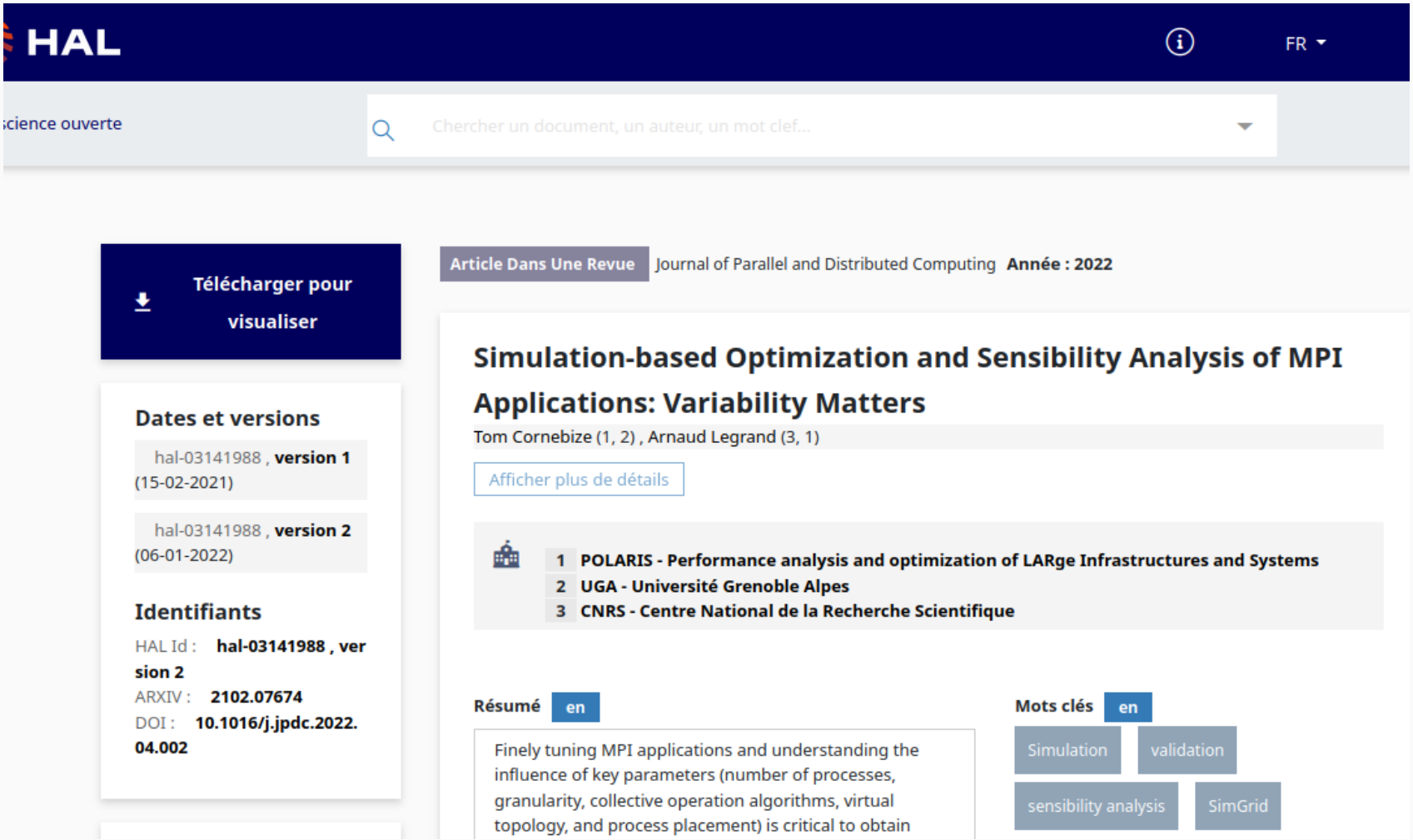
https://doi.org/10.1016/j.jpdc.2022.04.002 | Get rights and content

Abstract

Finely tuning MPI applications and understanding the influence of key parameters (number of processes, granularity, collective operation algorithms, virtual topology, and

- Use your institution subscription... or Sci-Hub 😉, or HAL/Arxiv

Go read the paper by Smith et. al. 2009



- Use your institution subscription... or Sci-Hub 😉, or HAL/Arxiv

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

Unfolding-based Partial Order Reduction*

César Rodríguez[1], Marcelo Sousa[2], Subodh Sharma[3], and Daniel Kroening[4]

1 Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS, France
2,4 Department of Computer Science, University of Oxford, UK
3 Indian Institute of Technology Delhi, India

Abstract

Partial order reduction (POR) and net unfoldings are two alternative methods to tackle state-space explosion caused by concurrency. In this paper, we propose the combination of both approaches in an effort to combine their strengths. We first define, for an abstract execution model, unfolding semantics parameterized over an arbitrary independence relation. Based on it, our main contribution is a novel stateless POR algorithm that explores at most one execution per Mazurkiewicz trace, and in general, can explore exponentially fewer, thus achieving a form of *super-optimality*. Furthermore, our unfolding-based POR copes with non-terminating executions and incorporates state-caching. Over benchmarks with busy-waits, among others, our experiments show a dramatic reduction in the number of executions when compared to a

Jul 2015

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

Algorithm 1: An unfolding-based POR exploration algorithm.

```
1  Initially, set U := {⊥}, set G := ∅, and call Explore({⊥}, ∅, ∅).
2  Procedure Explore(C, D, A)
3      Extend(C)
4      if en(C) = ∅ return
5      if A = ∅
6          Choose e from en(C)
7      else
8          Choose e from A ∩ en(C)
9      Explore(C ∪ {e}, D, A ∖ {e})
10     if ∃J ∈ Alt(C, D ∪ {e})
11         Explore(C, D ∪ {e}, J ∖ C)
12     Remove(e, C, D)
13 Procedure Extend(C)
14     Add ex(C) to U
15 Procedure Remove(e, C, D)
16     Move {e} ∖ Q_{C,D,U} from U to G
17     foreach ê ∈ #^i_U(e)
18         Move [ê] ∖ Q_{C,D,U} from U to G
```

- Looks good!

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

We give some new definitions. Let C be a configuration of $\mathcal{U}$. The *extensions* of C, written $ex(C)$, are all those events outside C whose causes are included in C. Formally, $ex(C) := \{e \in E : e \notin C \wedge \lceil e \rceil \subseteq C\}$. We let $en(C)$ denote the set of events *enabled* by C, i.e., those corresponding to the transitions enabled at $state(C)$, formally defined as $en(C) := \{e \in ex(C) : C \cup \{e\} \in conf(\mathcal{U})\}$. All those events in $ex(C)$ which are not in $en(C)$ are the *conflicting extensions*, $cex(C) := \{e \in ex(C) : \exists e' \in C, e \#^i e'\}$. Clearly, sets $en(C)$ and $cex(C)$ partition the set $ex(C)$. Lastly, we define $\#^i(e) := \{e' \in E : e \#^i e'\}$, and $\#^i_U(e) := \#^i(e) \cap U$. The difference between both is that $\#^i(e)$ contains events from *anywhere* in the unfolding structure, while $\#^i_U(e)$ can only *see* events in U.

The algorithm is given in Alg. 1. `Explore`(C, D, A), the main procedure, is given the configuration that is to be explored as the parameter C. The parameter D (for *disabled*) is the set of set of events that have already been explored and prevents that `Explore()` repeats work. It can be seen as a *sleep set* [7]. Set A (for *add*) is occasionally used to guide the direction of the exploration.

Additionally, a global set U stores all events presently known to the algorithm. Whenever some event can safely be discarded from memory, `Remove` will move it from U to G (for *garbage*). Once in G, it can be discarded at any time, or be preserved in G in order to save work when it is re-inserted in U. Set G is thus our *cache memory* of events.

The key intuition in Alg. 1 is as follows. A call to `Explore`(C, D, A) visits all maximal configurations of $\mathcal{U}$ which contain C and do not contain D; and the first one explored will

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

POSIX threads. The analyzer accepts deterministic programs, implements a variant of Alg. 1 where the computation of the alternatives is memoized, and supports cutoffs events with the criteria defined in § 5.

We ran Poet on a number of multi-threaded C programs. Most of them are adapted from benchmarks of the Software Verification Competition [17]; others are used in related works [8, 19, 2]. We investigate the characteristics of average program unfoldings (depth, width, etc.) as well as the frequency and impact of cutoffs on the exploration. We also compare Poet with Nidhugg [1], a state-of-the-art stateless model checking for multi-threaded C programs that implements Source-DPOR [2], an efficient but non-optimal DPOR. All experiments were run on an Intel Xeon CPU with 2.4 GHz and 4 GB memory. Tables 1 and 2 give our experimental data for programs with acyclic and non-acyclic state spaces, respectively.

For programs with acyclic state spaces (Table 1), Poet with and without cutoffs seems to perform the same exploration when the unfolding has no cutoffs, as expected. Furthermore, the number of explored executions also coincides with Nidhugg when the latter reports 0 sleep-set blocked executions (cf., § 4), providing experimental evidence of Poet's optimality.

The unfoldings of most programs in Table 1 do not contain cutoffs. All these programs are deterministic, and many of them highly sequential (Stf, Spin08, Fib), features known to make cutoffs unlikely. Ccnf(n) are concurrent programs composed of $n-1$ threads where thread i and $i+1$ race on writing one variable, and are independent of all remaining

[3] Source code and benchmarks available from: http://www.cs.ox.ac.uk/people/marcelo.sousa/poet/.

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments!

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

Table 1 Programs with acyclic state space. Columns are: $|P|$: nr. of threads; $|I|$: nr. of explored traces; $|B|$: nr. of sleep-set blocked executions; $t(s)$: running time; $|E|$: nr. of events in $\mathcal{U}$; $|E_{\text{cut}}|$: nr. of cutoff events; $|\Omega|$: nr. of maximal configurations; $\langle|U_\Omega|\rangle$: avg. nr. of events in U when exploring a maximal configuration. A $*$ marks programs containing bugs. <7K reads as "*fewer than 7000*".

Benchmark		Nidhugg			Poet (without cutoffs)				Poet (with cutoffs)				
Name	$\|P\|$	$\|I\|$	$\|B\|$	$t(s)$	$\|E\|$	$\|\Omega\|$	$\langle\|U_\Omega\|\rangle$	$t(s)$	$\|E\|$	$\|E_{\text{cut}}\|$	$\|\Omega\|$	$\langle\|U_\Omega\|\rangle$	$t(s)$
Stf	3	6	0	0.06	121	6	79	0.04	121	0	6	79	0.06
Stf*	3	-	-	0.05	-	-	-	0.02	-	-	-	-	0.03
Spin08	3	84	0	0.08	2974	84	1506	2.04	2974	0	84	1506	2.93
Fib	3	8953	0	3.36	<185K	8953	92878	305	<185K	0	8953	92878	704
Fib*	3	-	-	0.74	-	-	-	81.0	-	-	-	-	133
Ccnf(9)	9	16	0	0.05	49	16	46	0.07	49	0	16	46	0.06
Ccnf(17)	17	256	0	0.15	97	256	94	5.76	97	0	256	94	6.09
Ccnf(19)	19	512	0	0.28	109	512	106	22.5	109	0	512	106	22.0
Ssb	5	4	2	0.05	48	4	38	0.03	46	1	4	37	0.03
Ssb(1)	5	22	14	0.06	245	23	143	0.11	237	4	23	140	0.11
Ssb(3)	5	169	67	0.12	2798	172	1410	3.51	1179	48	90	618	0.90
Ssb(4)	5	336	103	0.15	<7K	340	3333	20.3	2179	74	142	1139	2.07
Ssb(8)	5	2014	327	0.85	<67K	2022	32782	4118	<12K	240	470	6267	32.1

Lastly, we note that this cutoff approach imposes no liability on what events shall be kept in the prefix, set G can be cleaned at discretion. Also, redefining (7) to use adequate orders [5] is straightforward, cf. App. C.1 (in our proofs we actually assume adequate orders).

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!
- Wait, what's this language? Did this ever run one day?

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!
- Wait, what's this language? Did this ever run one day?
- Wow, I'll need `CPLEX` and `Gurobi` but all I have is `lpsolve` or `glpk`

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!
- Wait, what's this language? Did this ever run one day?
- Wow, I'll need `CPLEX` and `Gurobi` but all I have is `lpsolve` or `glpk`
- Sweet, they provided a binary! Oh, wait, MacOSX in 2015 ?!?

Just compare to the Algorithm they proposed

Rodriguez et al., CONCUR'15

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!
- Wait, what's this language? Did this ever run one day?
- Wow, I'll need `CPLEX` and `Gurobi` but all I have is `lpsolve` or `glpk`
- Sweet, they provided a binary! Oh, wait, MacOSX in 2015 ?!?
- The GitHub webpage says it requires Foo, Bar, and Baz, but none of the versions I find appear to work.

Just compare to the Algorithm they proposed

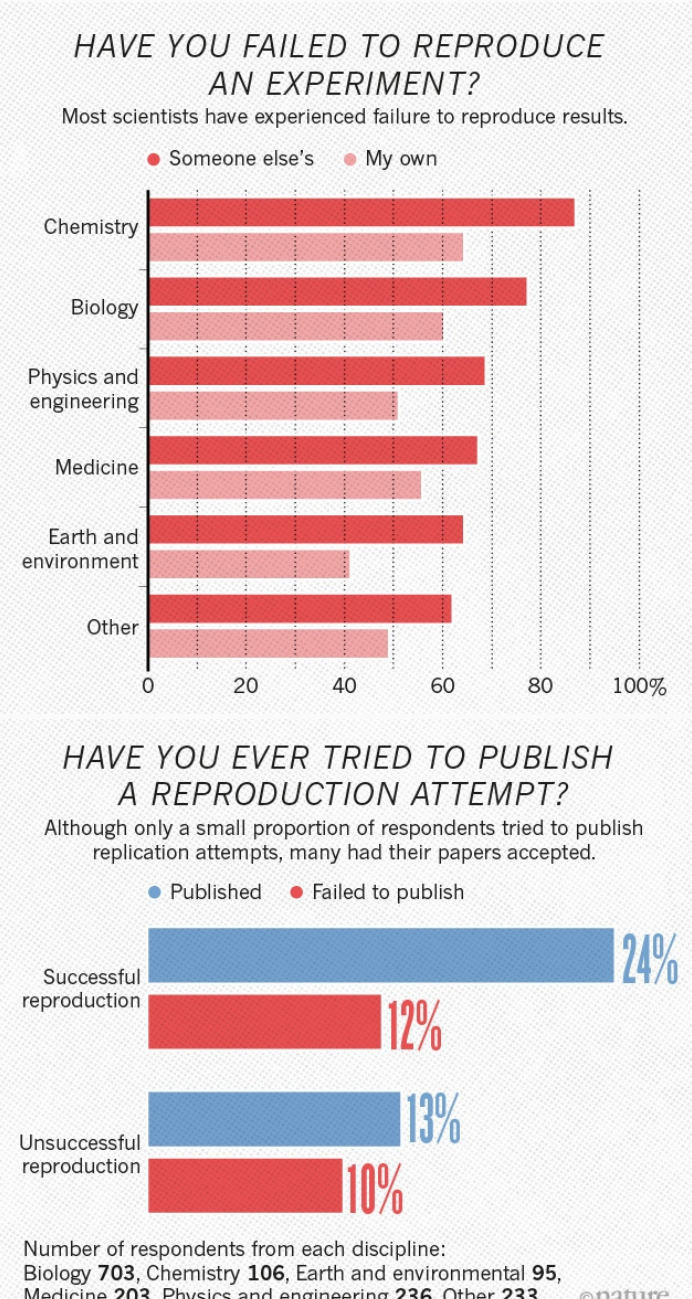
Rodriguez et al., CONCUR'15

- Looks good! Err... not so simple. Depending on how you do this, you quickly move from polynomial to exponential.
- Experiments! ⚠Possible *404, code not found!* ahead!!!
- Wait, what's this language? Did this ever run one day?
- Wow, I'll need `CPLEX` and `Gurobi` but all I have is `lpsolve` or `glpk`
- Sweet, they provided a binary! Oh, wait, MacOSX in 2015 ?!?
- The GitHub webpage says it requires Foo, Bar, and Baz, but none of the versions I find appear to work.
- With which parameters and data set do you run this code? And Why?

In the end, one new thesis to reproduce/understand this paper... and finally contribute.

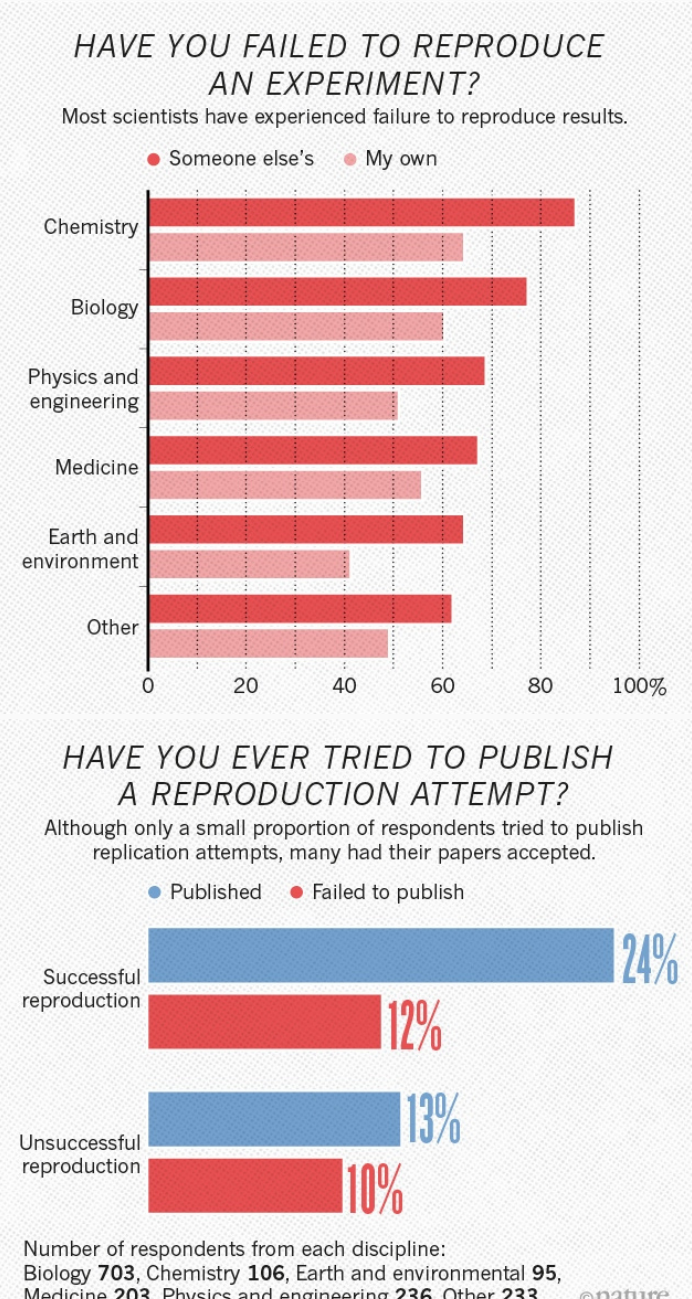
Different kinds of reproducibility

Socio-technical Challenges



1,500 scientists lift the lid on reproducibility, Nature, May 2016

Socio-technical Challenges

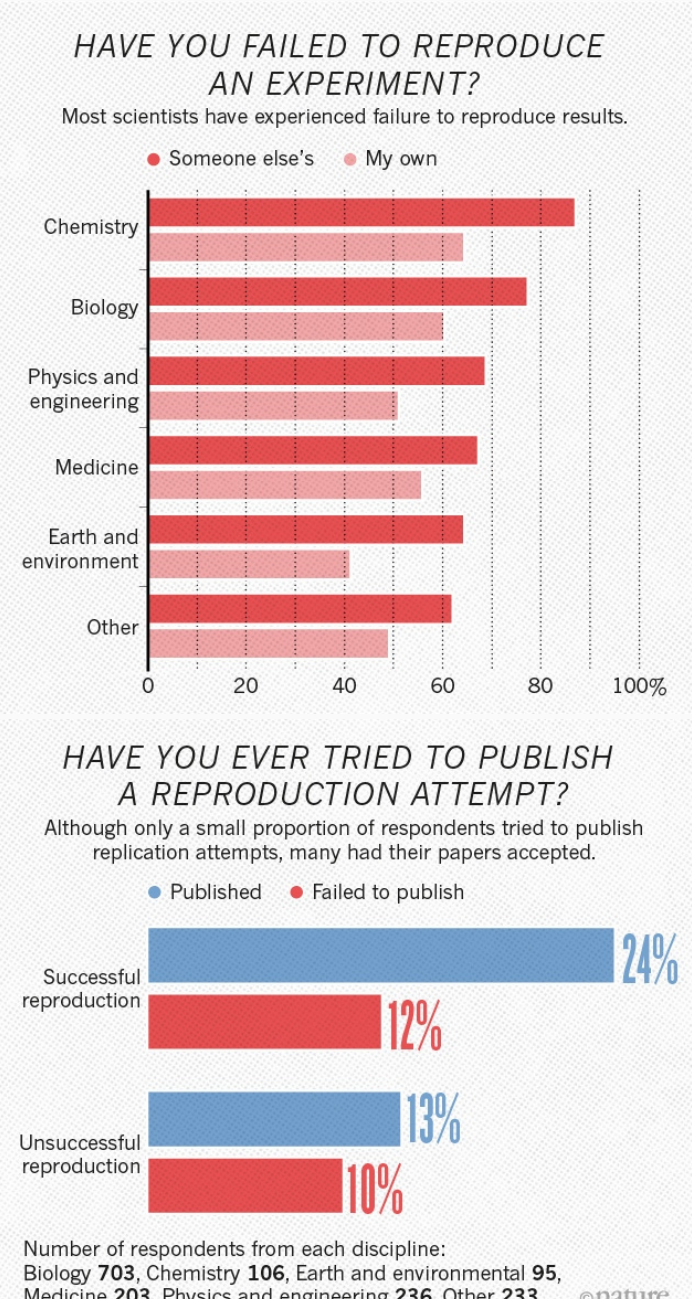


1,500 scientists lift the lid on reproducibility, Nature, May 2016

Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1M+ articles per year!

Socio-technical Challenges

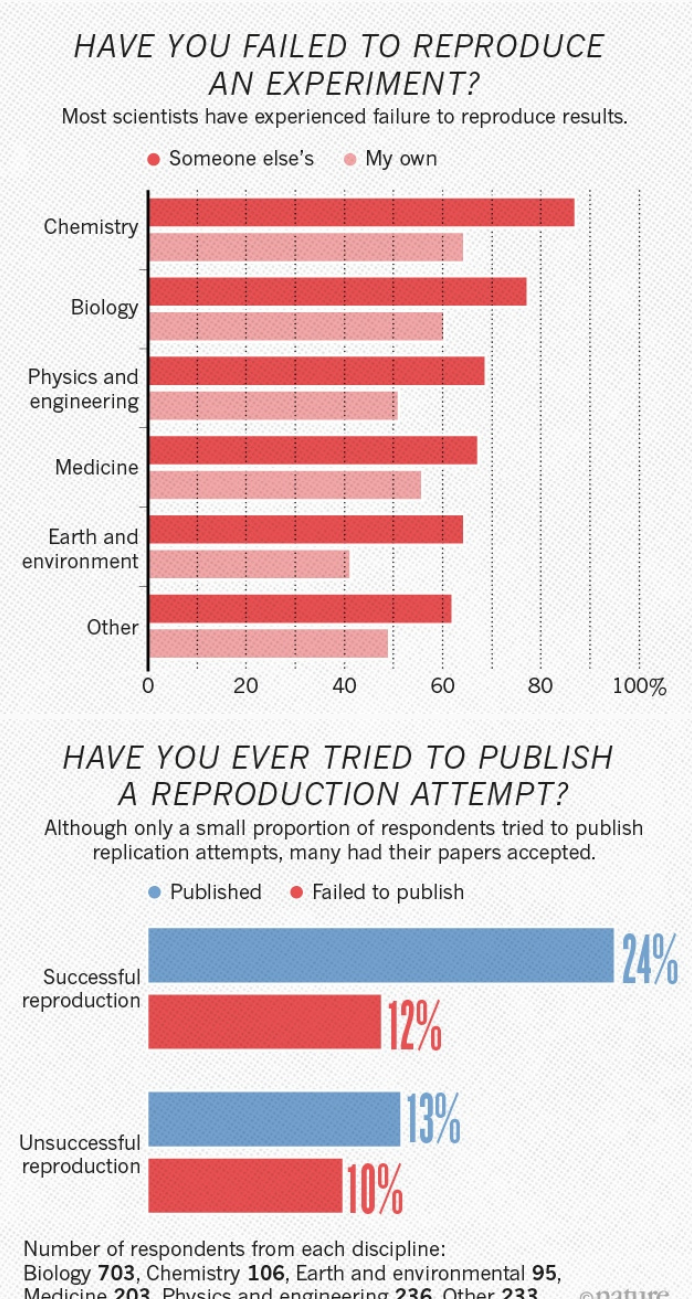


1,500 scientists lift the lid on reproducibility, Nature, May 2016

Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1M+ articles per year!
- Emerging practices: DORA/Plan S/COARA, DMP and FAIR data, artefact evaluation, reproducibility badges, reproducibility challenges, open reviews, ...

Socio-technical Challenges



1,500 scientists lift the lid on reproducibility, Nature, May 2016

Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1M+ articles per year!
- Emerging practices: DORA/Plan S/COARA, DMP and FAIR data, artefact evaluation, reproducibility badges, reproducibility challenges, open reviews, ...

Methodological/technical causes

- The many biases (apophenia, confirmation, hindsight, experimenter, ...): bad designs
- Selective reporting, weak analysis (statistics, data manipulation mistakes, computational errors)
- Lack of information, code/raw data unavailable

Different Reproducibility Concerns in Modern Science

Biology, Oncology sample provenance, clinical trials $\rightsquigarrow$ standardized protocols

Different Reproducibility Concerns in Modern Science

Biology, Oncology sample provenance, clinical trials $\rightsquigarrow$ standardized protocols

Psychology, Nutrition HARKING, p-hacking $\rightsquigarrow$ pre-registration

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

Different Reproducibility Concerns in Modern Science

Biology, Oncology sample provenance, clinical trials $\rightsquigarrow$ standardized protocols

Psychology, Nutrition HARKING, p-hacking $\rightsquigarrow$ pre-registration

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical chaos, parallel architectures

Different Reproducibility Concerns in Modern Science

Biology, Oncology sample provenance, clinical trials ⇝ standardized protocols

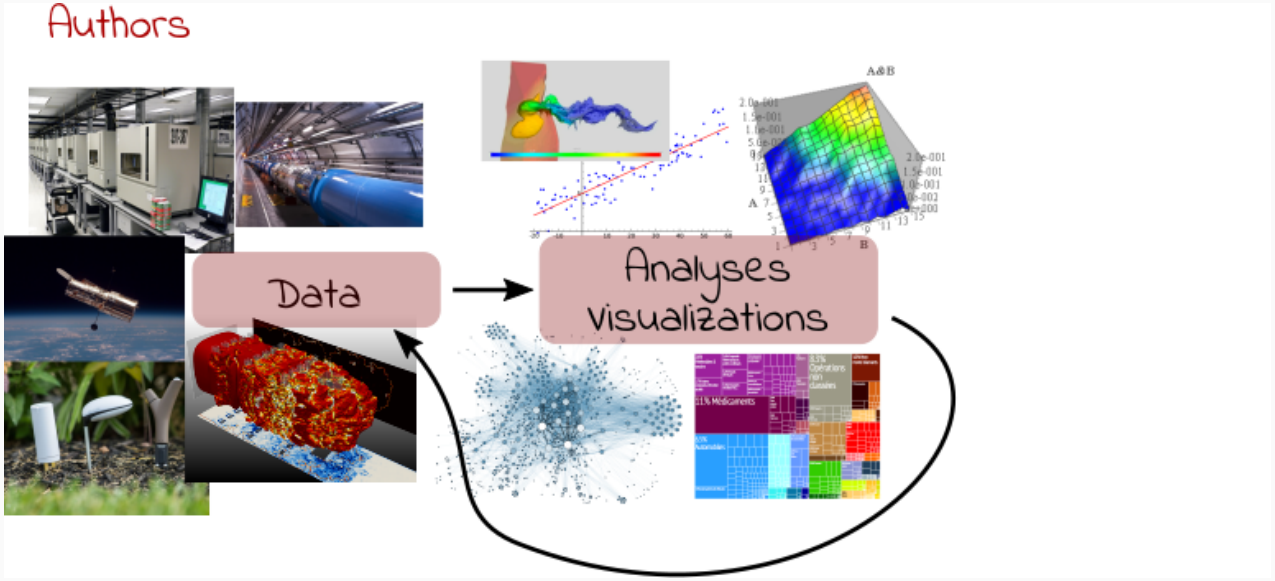
Psychology, Nutrition HARKING, p-hacking ⇝ pre-registration

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical chaos, parallel architectures

Artificial Intelligence most of the above 😉

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



Different Reproducibility Concerns in Modern Science

Biology, Oncology sample provenance, clinical trials ⇝ standardized protocols

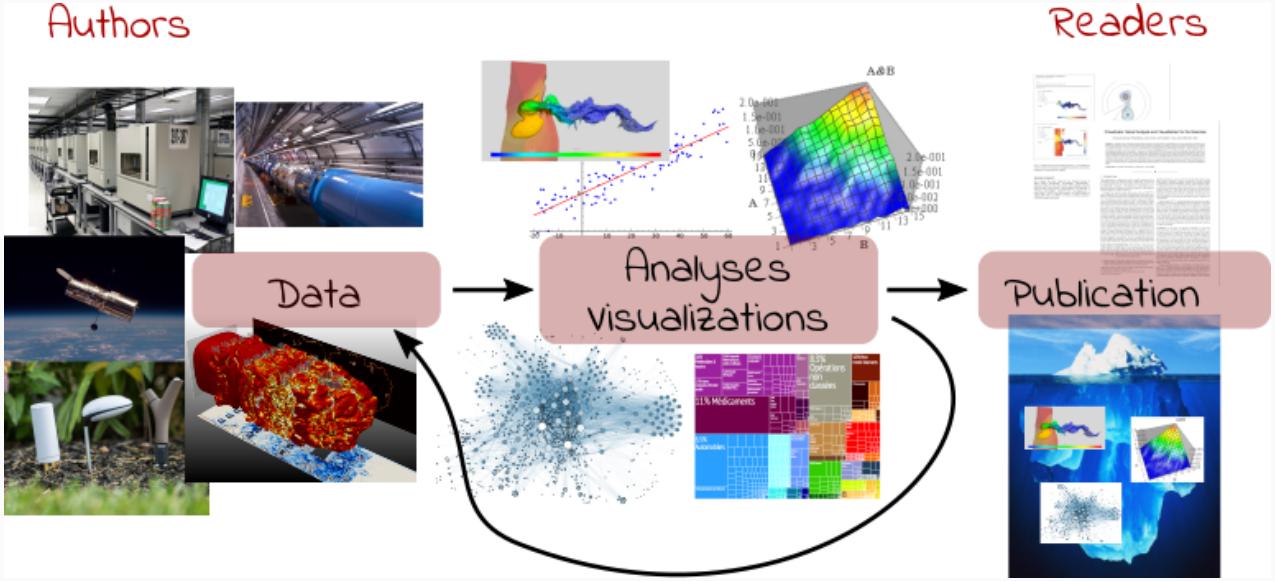
Psychology, Nutrition HARKING, p-hacking ⇝ pre-registration

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical chaos, parallel architectures

Artificial Intelligence most of the above 😉

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



How computers broke science – and what we can do about it – Ben Marwick, The conversation, 2015

Reproducible Research = Bridging the Gap by working Transparently

Good Practice #1:
Taking Notes and Documenting

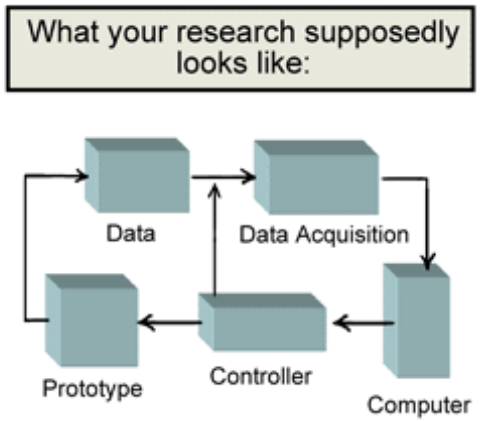


Figure 1. Experimental Diagram

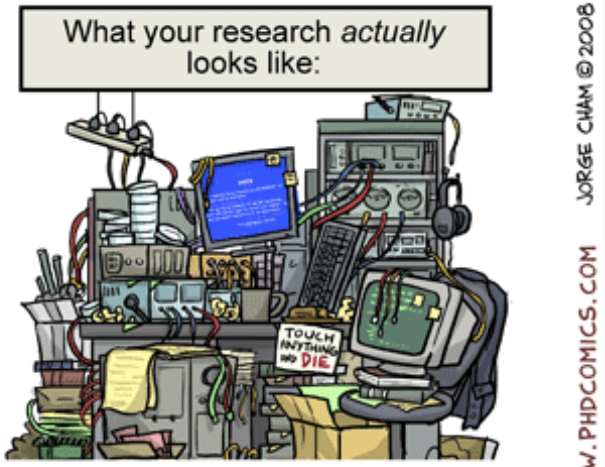


Figure 2. Experimental Mess

Main Challenges for a computational scientist

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

Main Challenges for a computational scientist

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

Main Challenges for a computational scientist

```
1 my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

Tracking parameters and data

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

Main Challenges for a computational scientist

```
1 my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

Tracking parameters and data

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

Tracking the process (on short/long term)

- Why did I run this? What did I learn from it? I remember doing this but when?

Main Challenges for a computational scientist

```
1 my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

Tracking parameters and data

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

Tracking the process (on short/long term)

- Why did I run this? What did I learn from it? I remember doing this but when?

Handle complex sequences and reuse results (leverage supercomputers)

Tool 1: Computational Notebooks (Litterate Programming)

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

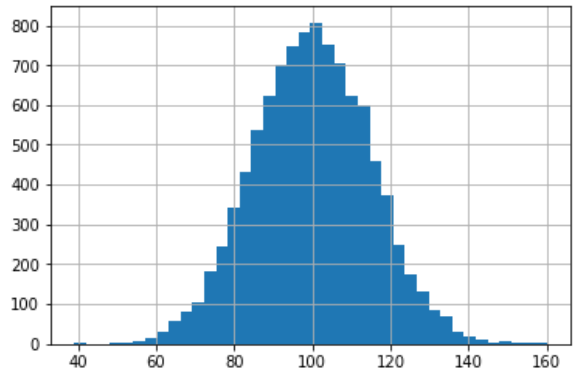
```
3.141592653589793
```

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... ☺).
```

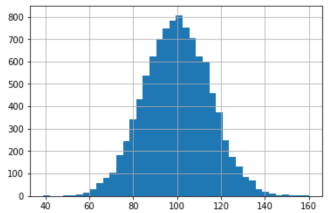
In [3]:

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*

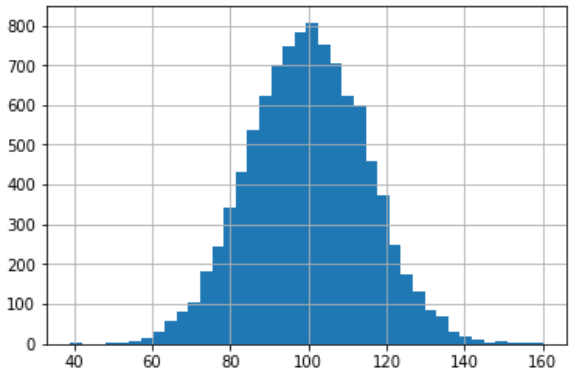
```
3.141592653589793
```

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
```

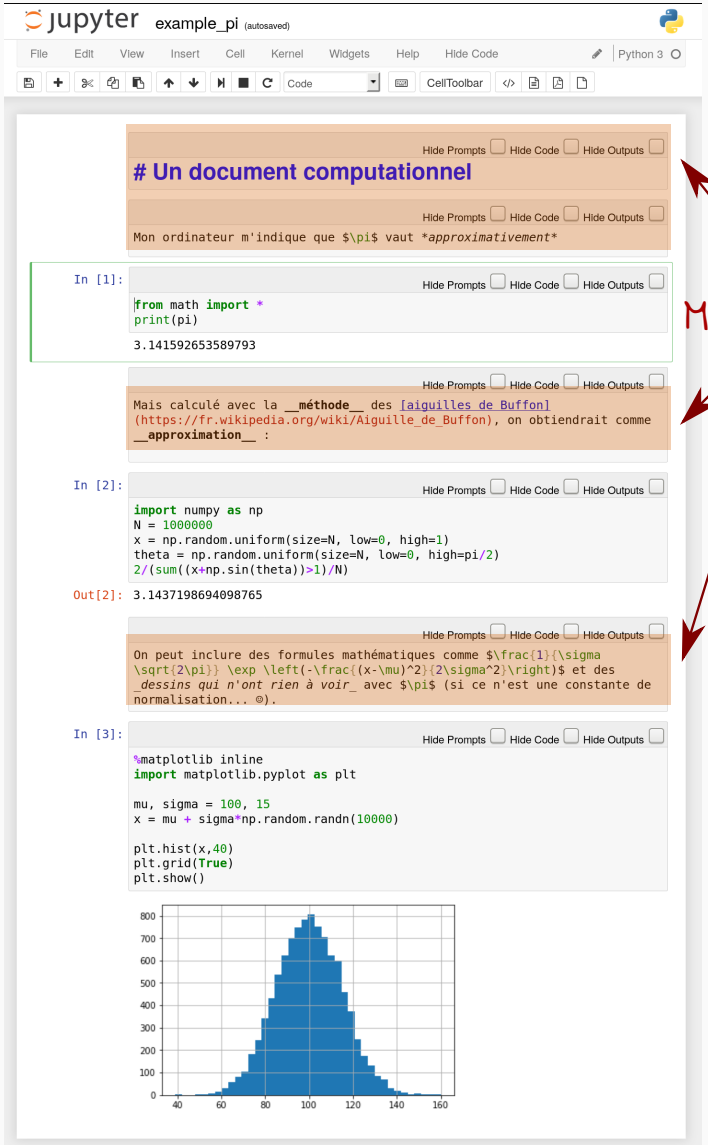
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



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*

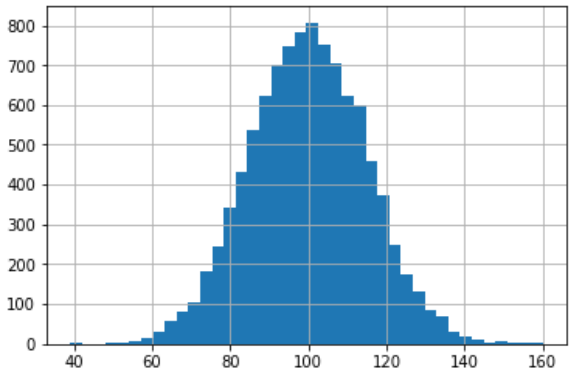
```
3.141592653589793
```

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)
```

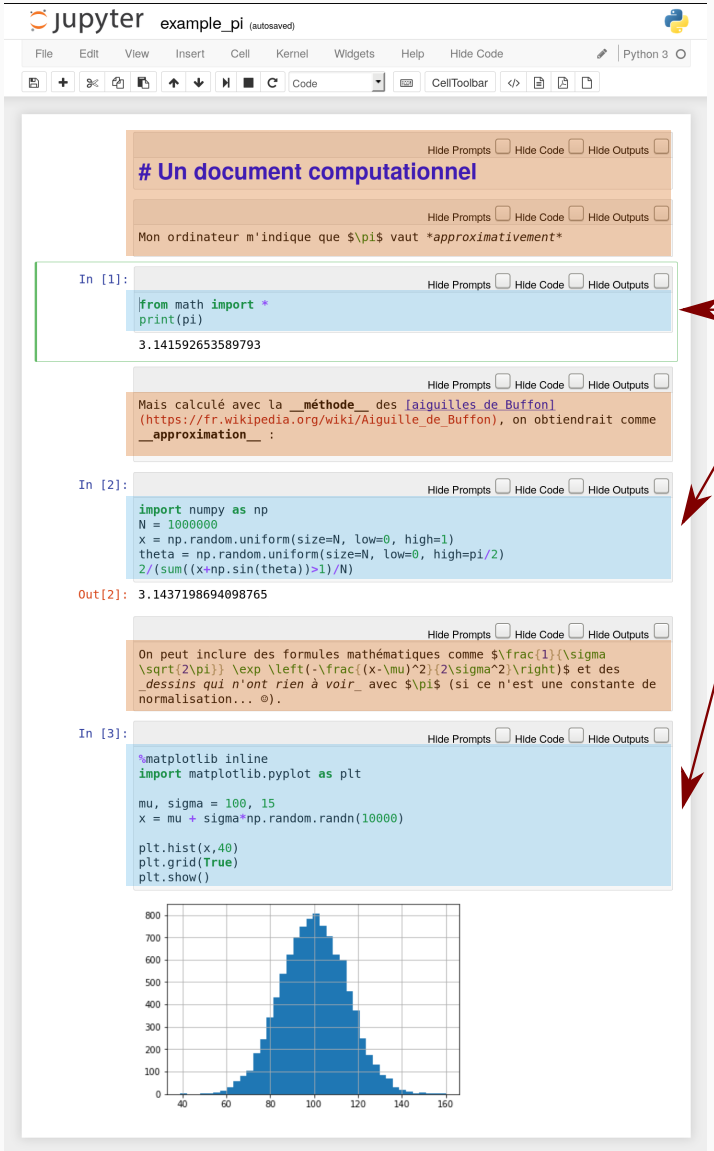
```
3.1437198694098765
```

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*

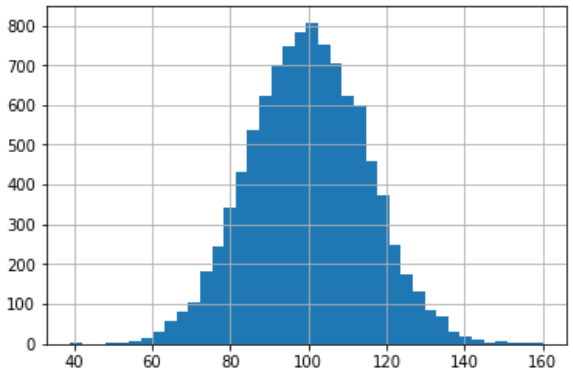
```
3.141592653589793
```

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
```

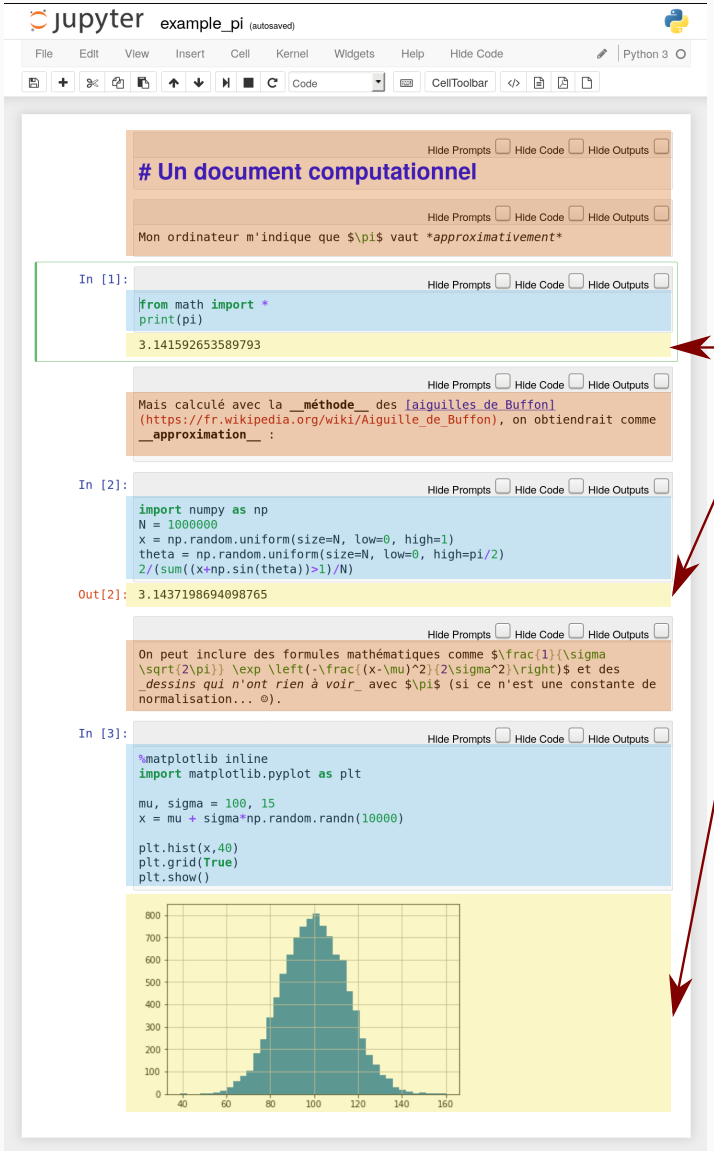
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks (Litterate Programming)

Document initial dans son environnement

Document final



Résultats

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

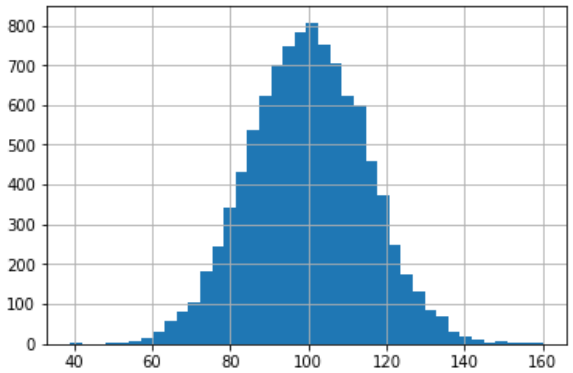
```
3.141592653589793
```

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... ☺).
```

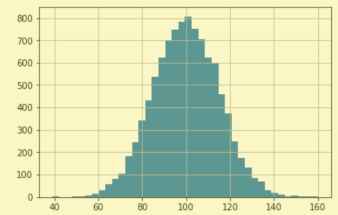
In [3]:

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()
```



Export

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

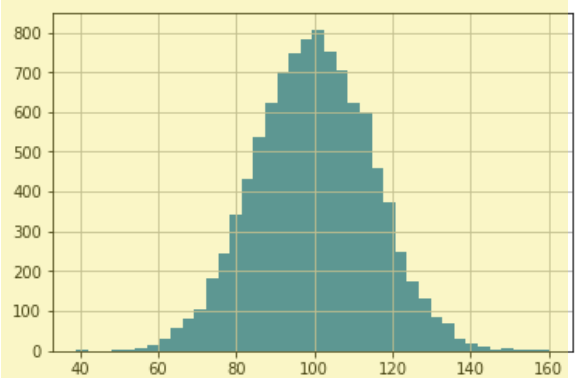
```
3.141592653589793
```

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)
```

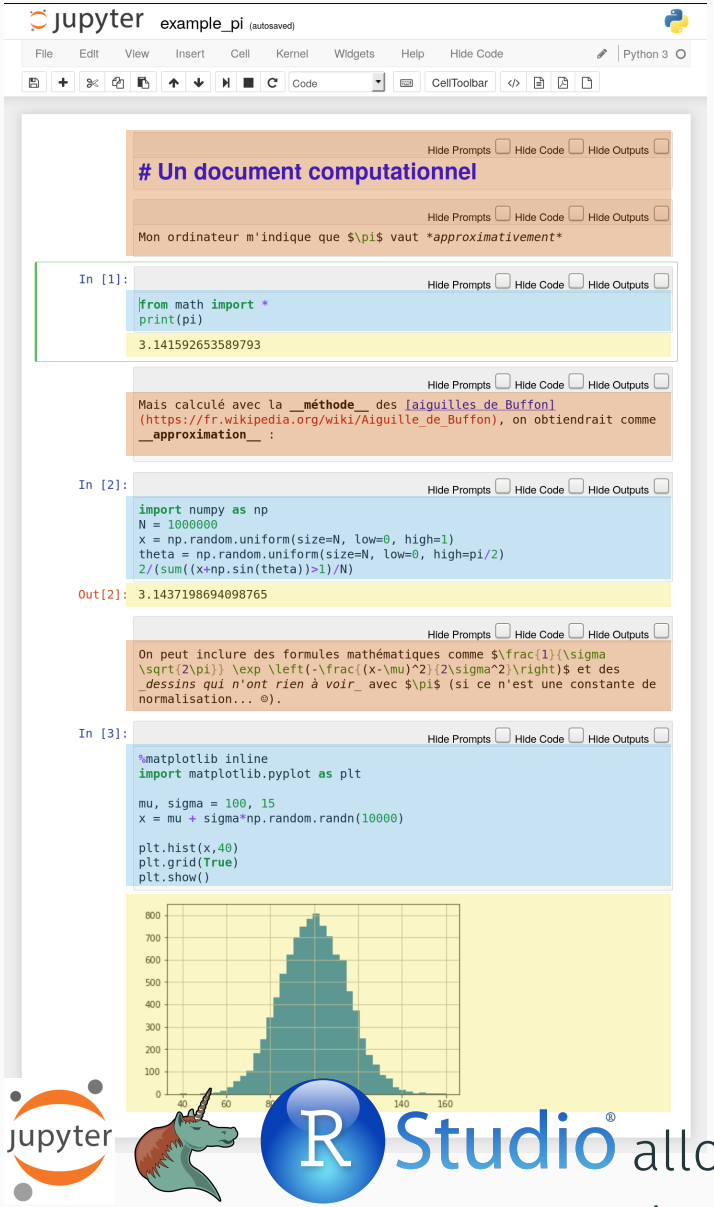
```
3.1437198694098765
```

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*

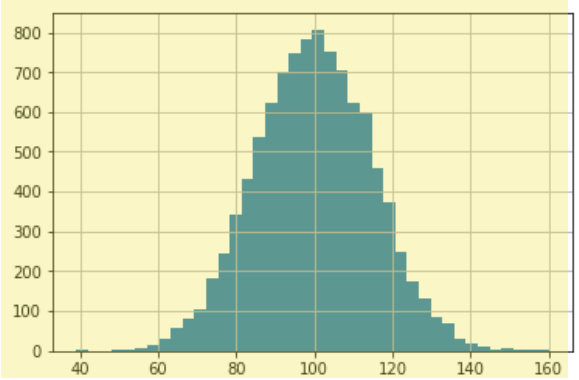
```
3.141592653589793
```

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)
```

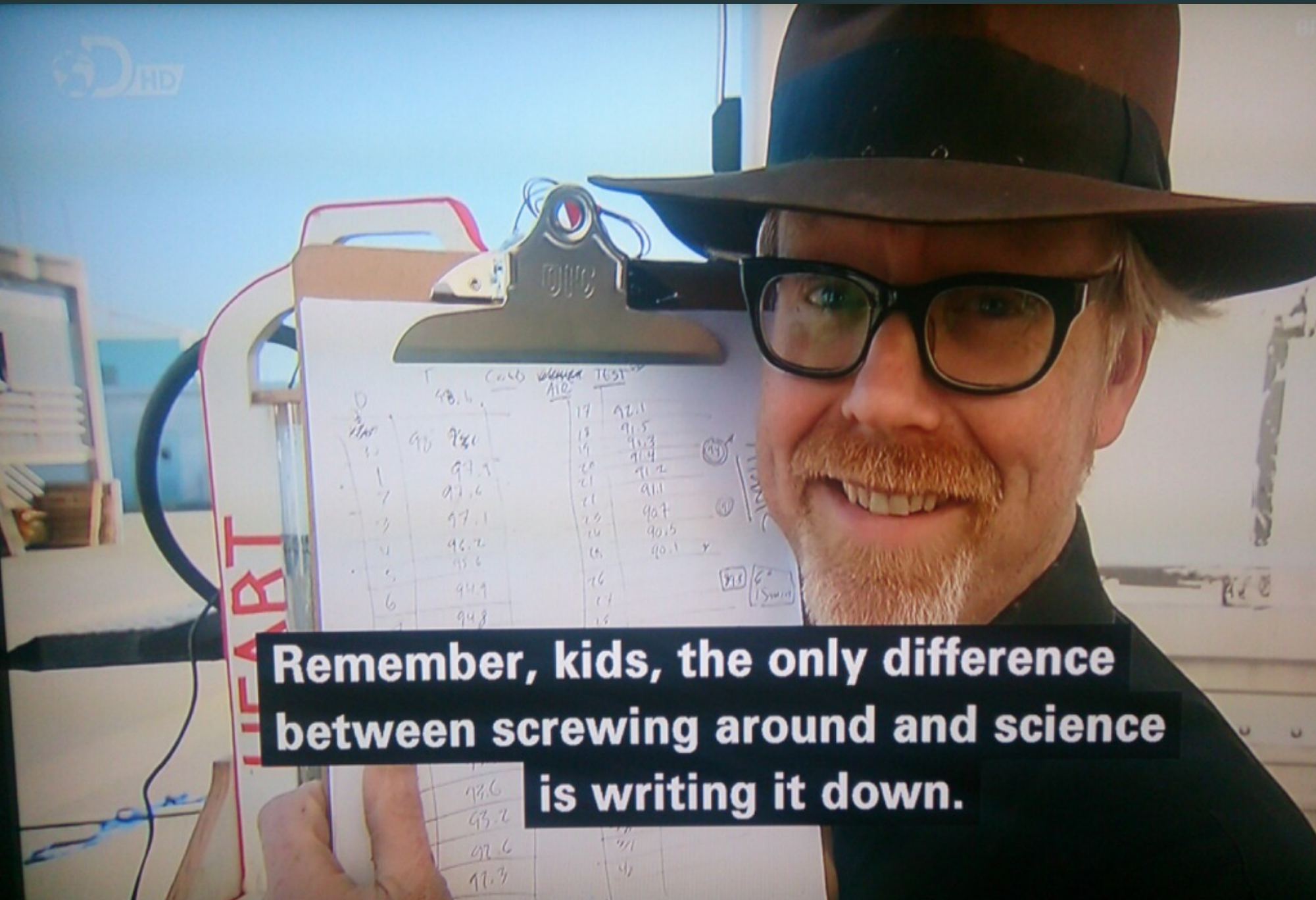
```
3.1437198694098765
```

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... ☺).



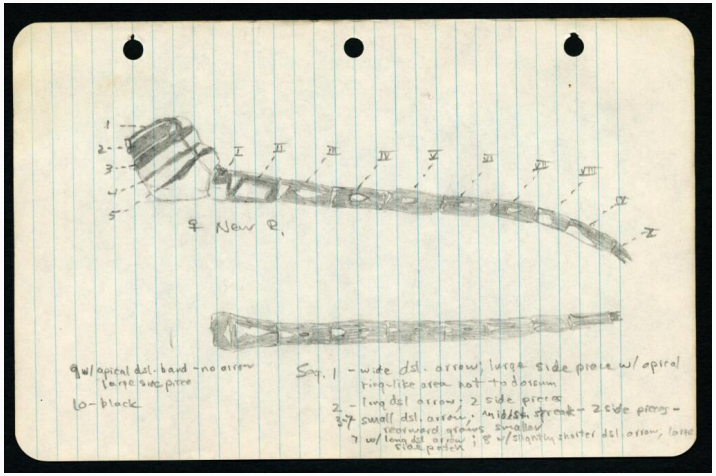
jupyter RStudio® allow to write a reproducible document that links raw data with final figures

Tool 2: Electronic Notebooks



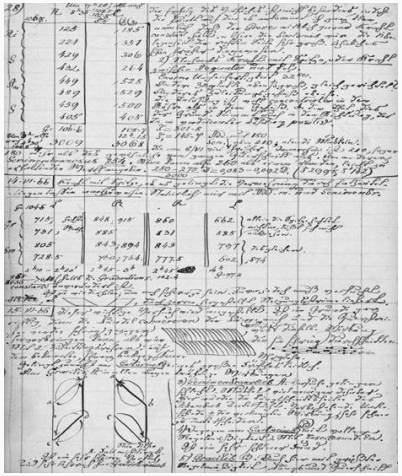
Field and Laboratory notebooks

Social Sciences, Ecology, Biology



Robert Henry Gibbs, Jr.,
ichthyologist (1929 – 1988)

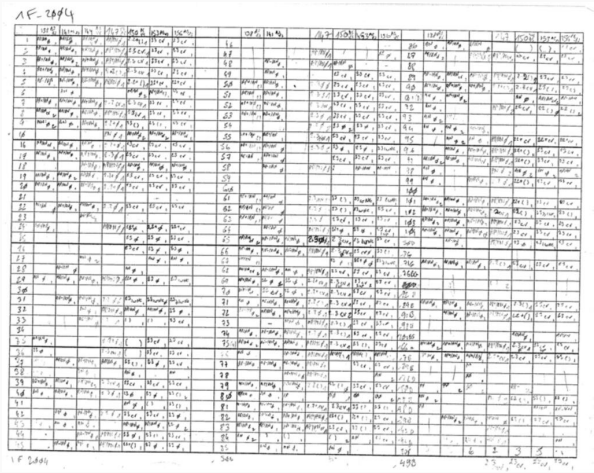
Chemistry, Physics, Biology



Emil Heinrich du Bois-Reymond,
electrophysiology (1818 – 1896)

Field and Laboratory notebooks

Social Sciences, Ecology, Biology



Marie-Claude Quidoz, Centre d'Ecologie Fonctionnelle et Evolutive

Chemistry, Physics, Biology

Contemporary laboratory notebook in Neurology

Essentially no evolution throughout the last century

Taking notes

Org-mode and Markdown two simple text formats

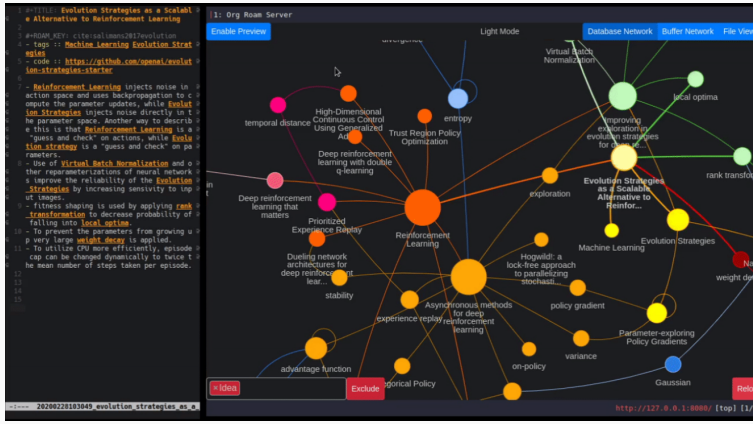
- simple formatting and export to more elaborate formats
- Hyperlinks, images, code, etc.

Journal structure

- My journal/notebook (single org file)
- Tom Cornebize's journal (single org file + Jupyter notebooks)

Zettelkasten possible structure

- Org-roam, Zettler, Roam, Obsidian, ...
- Architects, librarians, gardeners



Taking notes

Org-mode and Markdown two simple text formats

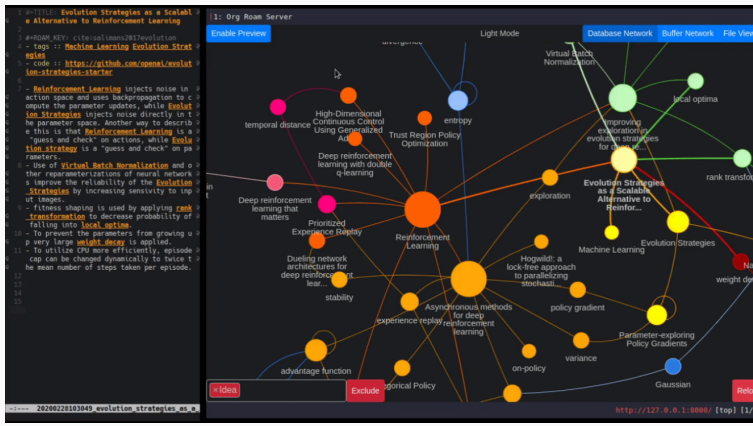
- simple formatting and export to more elaborate formats
- Hyperlinks, images, code, etc.

Journal structure

- My journal/notebook (single org file)
- Tom Cornebize's journal (single org file + Jupyter notebooks)

Zettelkasten possible structure

- Org-roam, Zettler, Roam, Obsidian, ...
- Architects, librarians, gardeners



My recommendations: Do not use a fancy cloud-based proprietary tool

- Simple open source text-based format
- Control versions and backup yourself (e.g., using gitlab, github)
- Single location if possible
- Annotate (tags in a journal, links in a Zettelkasten)

Good Practice #2
Controling Software Environment

Nightmare 1: Fighting Software Environments Nightmare

What is hiding behind a simple

```
1 import matplotlib
```

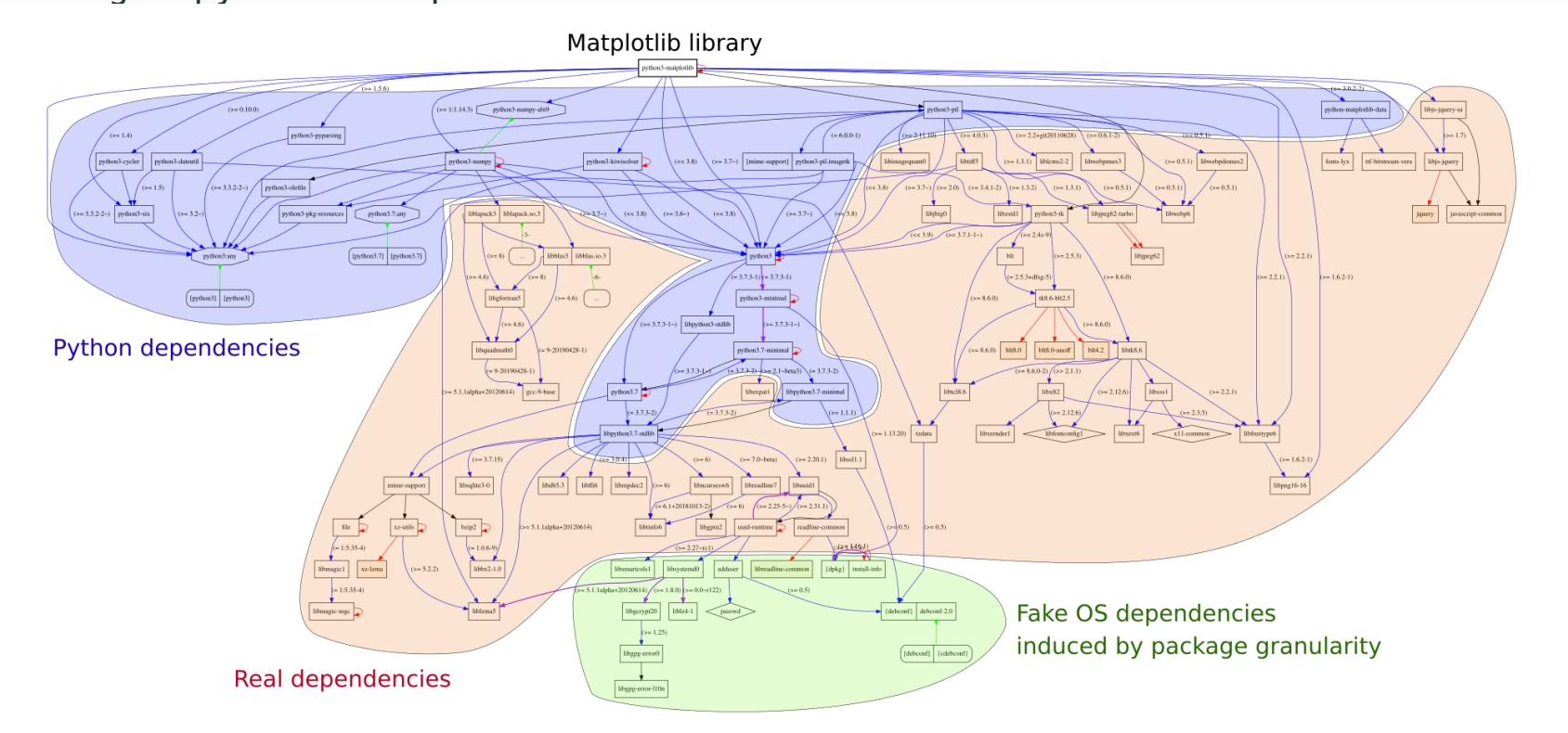
```
Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2),
python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz,
libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1),
python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~),
python3-cycler (>= 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)
```

Nightmare 1: Fighting Software Environments Nightmare

What is hiding behind a simple

```
1 import matplotlib
```

`Package: python3-matplotlib`



Nightmare 1: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

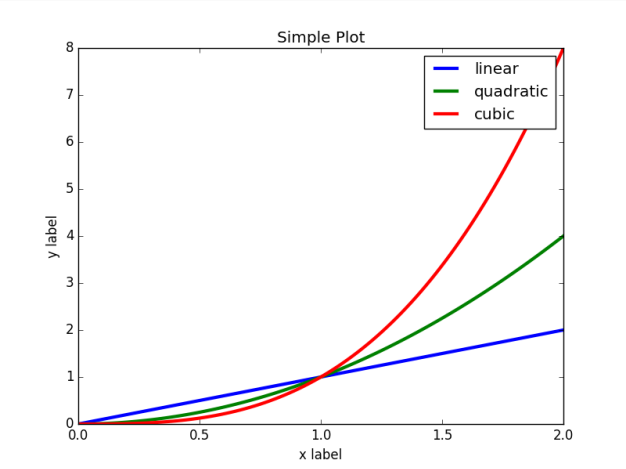
```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

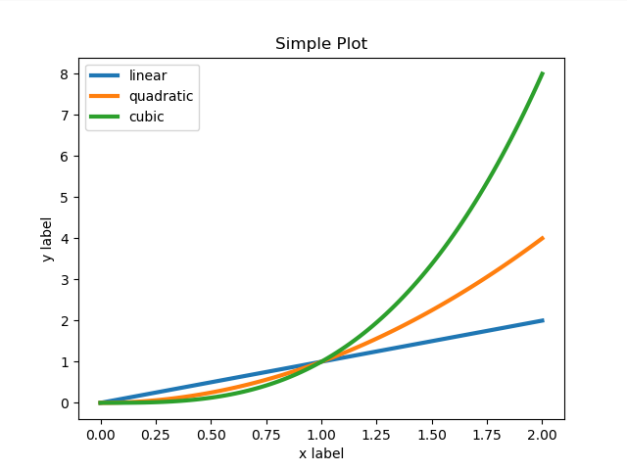
```
3
3.3333333333333335
```

Nightmare 1: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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

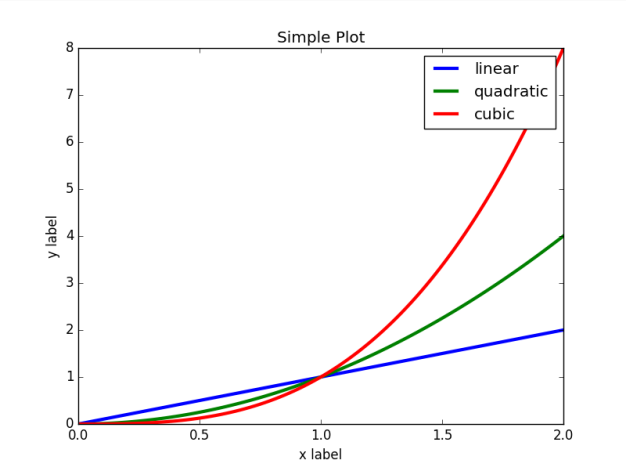


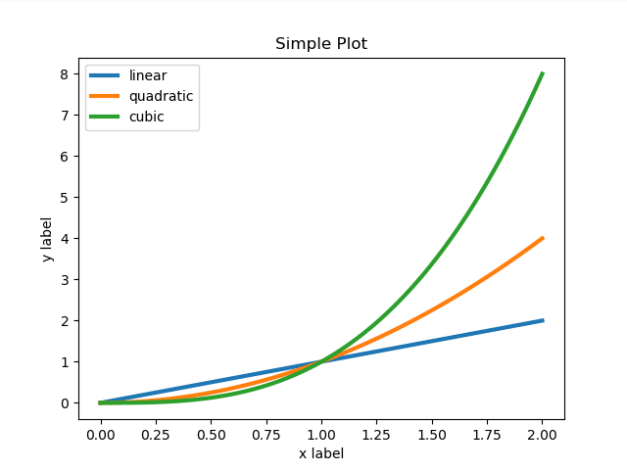


Nightmare 1: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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





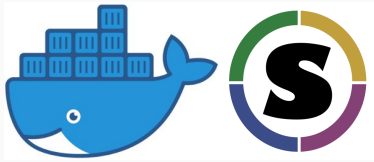
Cortical Thickness Measurements (PLOS ONE, June 2012) in *FreeSurfer*

> *About a* ***factor two smaller differences*** *were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

Tool 3: Containers and Package Managers

The good

The bad

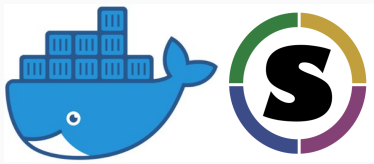


The uggly

Automatic tracking

Tool 3: Containers and Package Managers

The good The bad The uggly



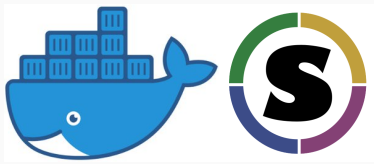

Automatic tracking

Containers

- Pros: Lightweight, Good isolation, Easy to use
  - Running as easy as `docker run <img> <cmd>`
  - Building images: `docker build -f <Dockerfile>`
  - Sharing through the Docker Hub: `docker pull/push <img>`

Tool 3: Containers and Package Managers

The good The bad The uggly




Automatic tracking

Containers

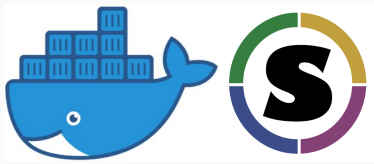
- Pros: Lightweight, Good isolation, Easy to use
- Cons: Opaque, Container build is generally not reproducible
  - Recipes rarely follow *reproducibility good practices*

```
1  FROM ubuntu:20.04
2  RUN apt-get update
3       && apt-get upgrade -y
4       && apt-get install -y ...
```

  - Choose a stable image (and the smallest possible)
  - Include only the necessary libraries (e.g. no graphics libs)
  - Avoid system updates (instead freeze sources)

Tool 3: Containers and Package Managers

The good The bad The uggly




Automatic tracking

Containers

- Pros: Lightweight, Good isolation, Easy to use
- Cons: Opaque, Container build is generally not reproducible

Package managers (the uggly and the good)

- Language specific: `pip/pipenv/virtualenv`, `conda`, `CRAN/Bioconductor`
  - Limits: version management, durability, permeable, language centric
- GUIX/NiX = Full-fledged functional package manager
  - Native support for environment (*à la git*)
  - Isolation through `--pure`
  - Recompile from source (cache recommended)

Other Computer Related Nightmares

Nightmare 2: 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*

Nightmare 2: 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

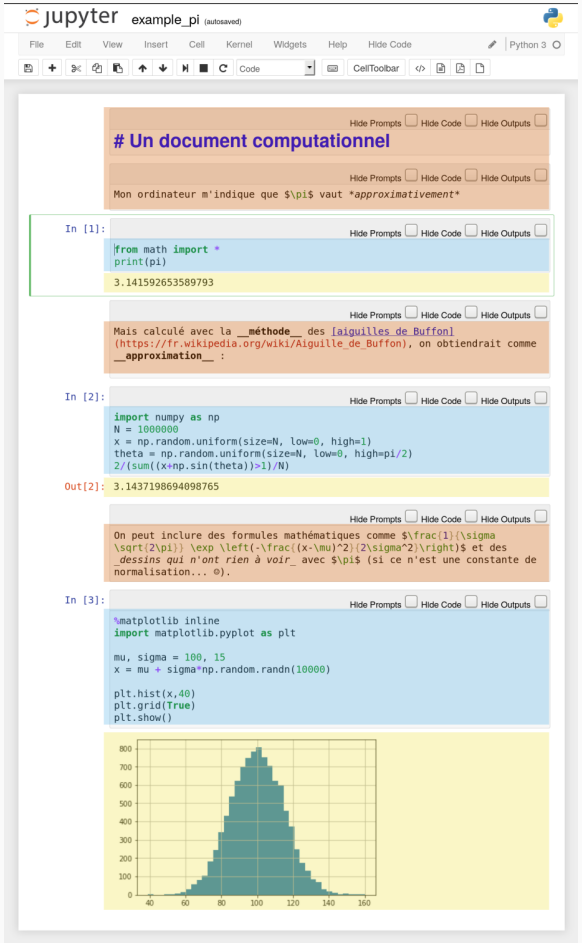
Software Archive Software Heritage

or = awesome but:

- Inadequate for **large** data
- ≠ Archive

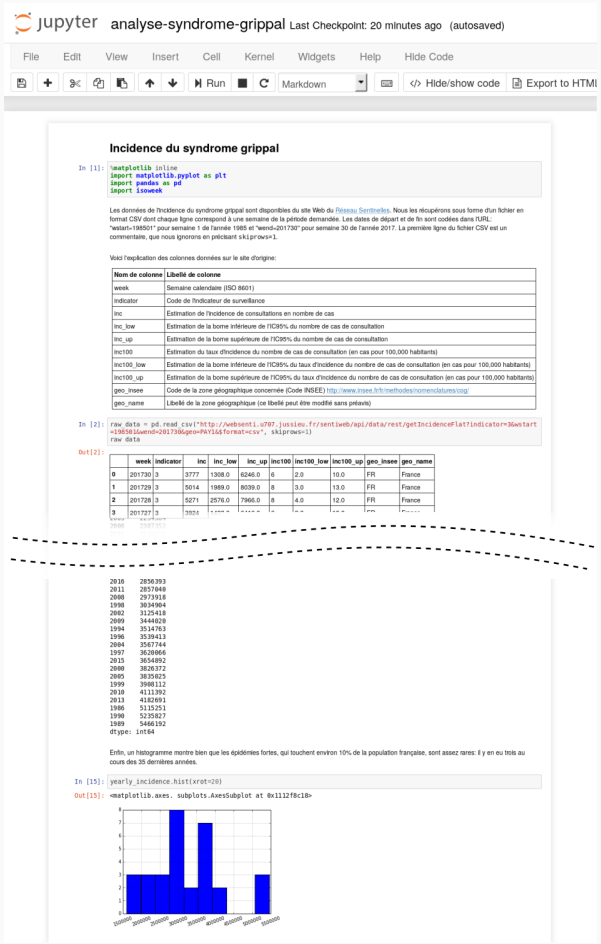
Tool 5: Automating computations with workflows

Notebooks are no panacea and do not help developing clean code



Tool 5: Automating computations with workflows

Notebooks are no panacea and do not help developing clean code


jupyter analyse-syndrome-grippal Last Checkpoint: 20 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code

Run Markdown Hide/show code Export to HTML

Incidence du syndrome grippal

```
In [1]: %matplotlib inline
        import matplotlib.pyplot as plt
        import pandas as pd
        import isoweek
```

Les données de l'incidence du syndrome grippal sont disponibles du site Web du Réseau Sentinelles. Nous les récupérons sous forme d'un fichier en format CSV dont chaque ligne correspond à une semaine de la période demandée. Les dates de départ et de fin sont codées dans l'URL : "sstart=198501" pour semaine 1 de l'année 1985 et "send=201730" pour semaine 30 de l'année 2017. La première ligne du fichier CSV est un commentaire, que nous ignorons en précisant `skiprows=1`.

Voici l'explication des colonnes données sur le site d'origine:

Nom de colonne	Libellé de colonne
week	Semaine calendaire (ISO 8601)
indicator	Code de l'indicateur de surveillance
inc	Estimation de l'incidence de consultations en nombre de cas
inc_low	Estimation de la borne inférieure de l'IC95% du nombre de cas de consultation
inc_up	Estimation de la borne supérieure de l'IC95% du nombre de cas de consultation
inc100	Estimation du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants)
inc100_low	Estimation de la borne inférieure de l'IC95% du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants)
inc100_up	Estimation de la borne supérieure de l'IC95% du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants)
geo_insee	Code de la zone géographique concernée (Code INSEE) http://www.insee.fr/fr/methodes/nomenclatures/cog/
geo_name	Libellé de la zone géographique (ce libellé peut être modifié sans préavis)

```
In [2]: raw_data = pd.read_csv("http://websenti.u707.jussieu.fr/sentiweb/api/data/rest/getIncidenceFlat?indicator=3&sstart
        =198501&send=201730&geo=PAYS1&format=csv", skiprows=1)
        raw_data
Out[2]:
```

	week	indicator	inc	inc_low	inc_up	inc100	inc100_low	inc100_up	geo_insee	geo_name
0	201730	3	3777	1308.0	6246.0	6	2.0	10.0	FR	France
1	201729	3	5014	1989.0	8039.0	8	3.0	13.0	FR	France
2	201728	3	5271	2576.0	7966.0	8	4.0	12.0	FR	France
3	201727	3	3924							

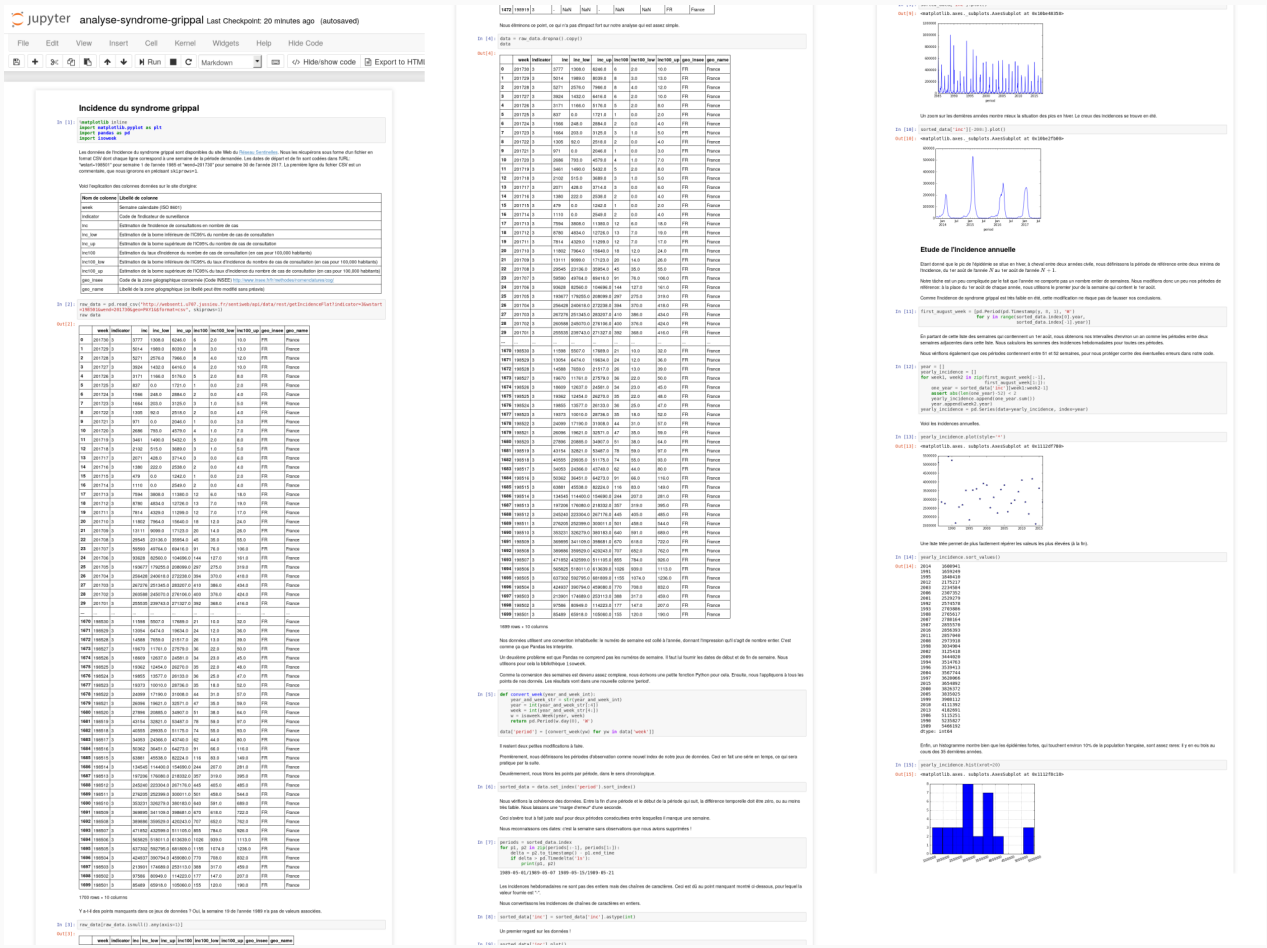
```
2016    2856393
2011    2857040
2008    2973918
1998    3034904
2002    3125418
2009    3444028
1994    3514763
1996    3539413
2004    3567744
1997    3620066
2015    3654892
2000    3826372
2005    3835025
1999    3908112
2010    4111392
2013    4182691
1986    5115251
1990    5235827
1989    5466192
dtype: int64
```

Enfin, un histogramme montre bien que les épidémies fortes, qui touchent environ 10% de la population française, sont assez rares: il y en eu trois au cours des 35 dernières années.

```
In [15]: yearly_incidence.hist(xrot=20)
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1112f8c18>
```

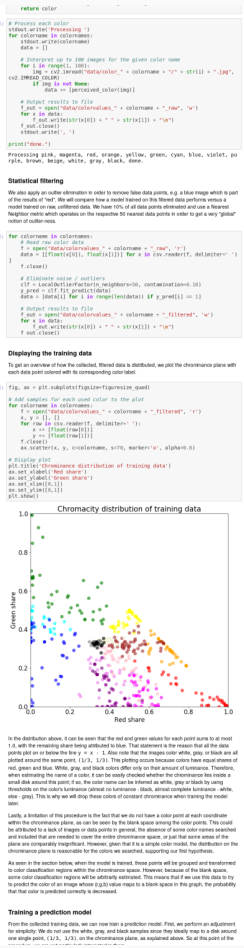
Tool 5: Automating computations with workflows

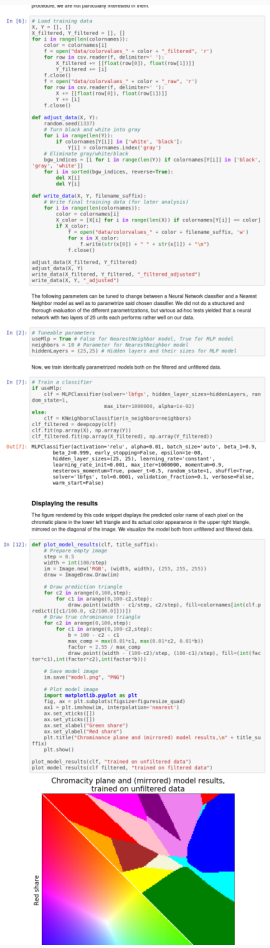
Notebooks are no panacea and do not help developing clean code

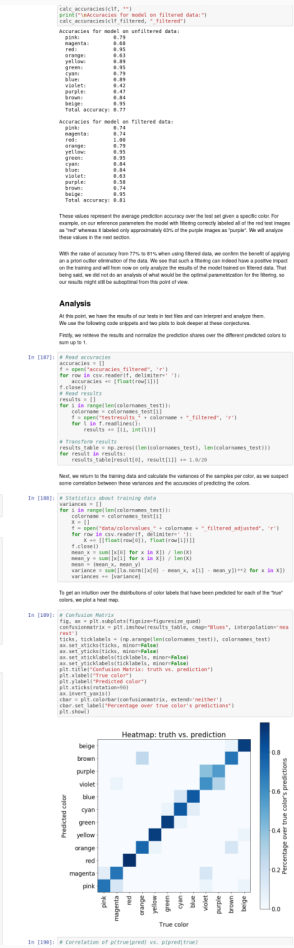


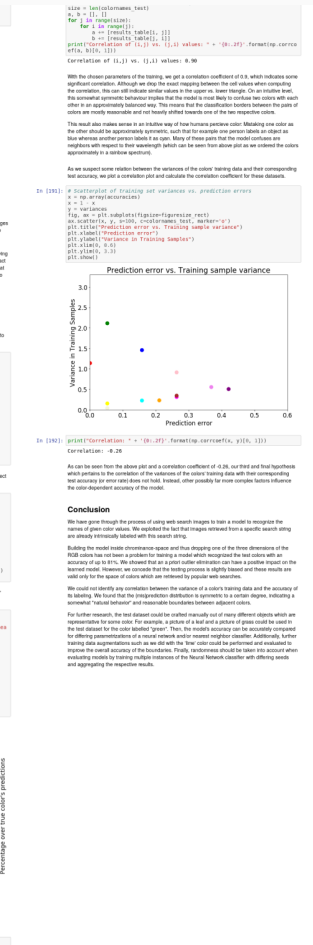
Tool 5: Automating computations with workflows

Notebooks are no panacea and do not help developing clean code

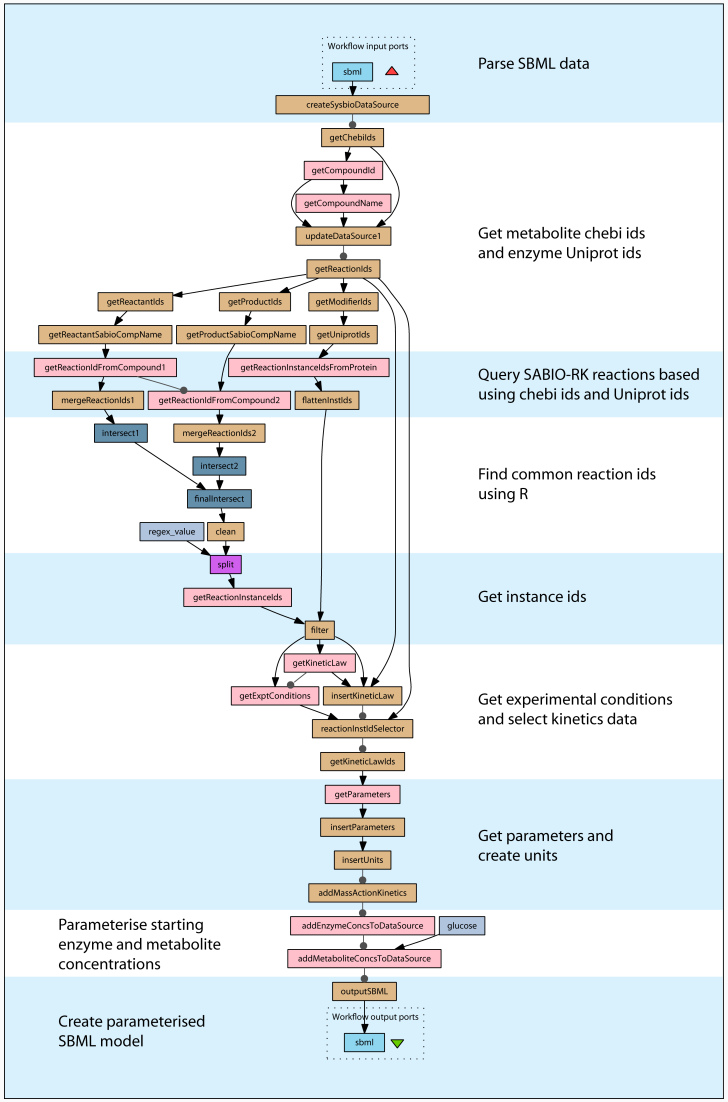


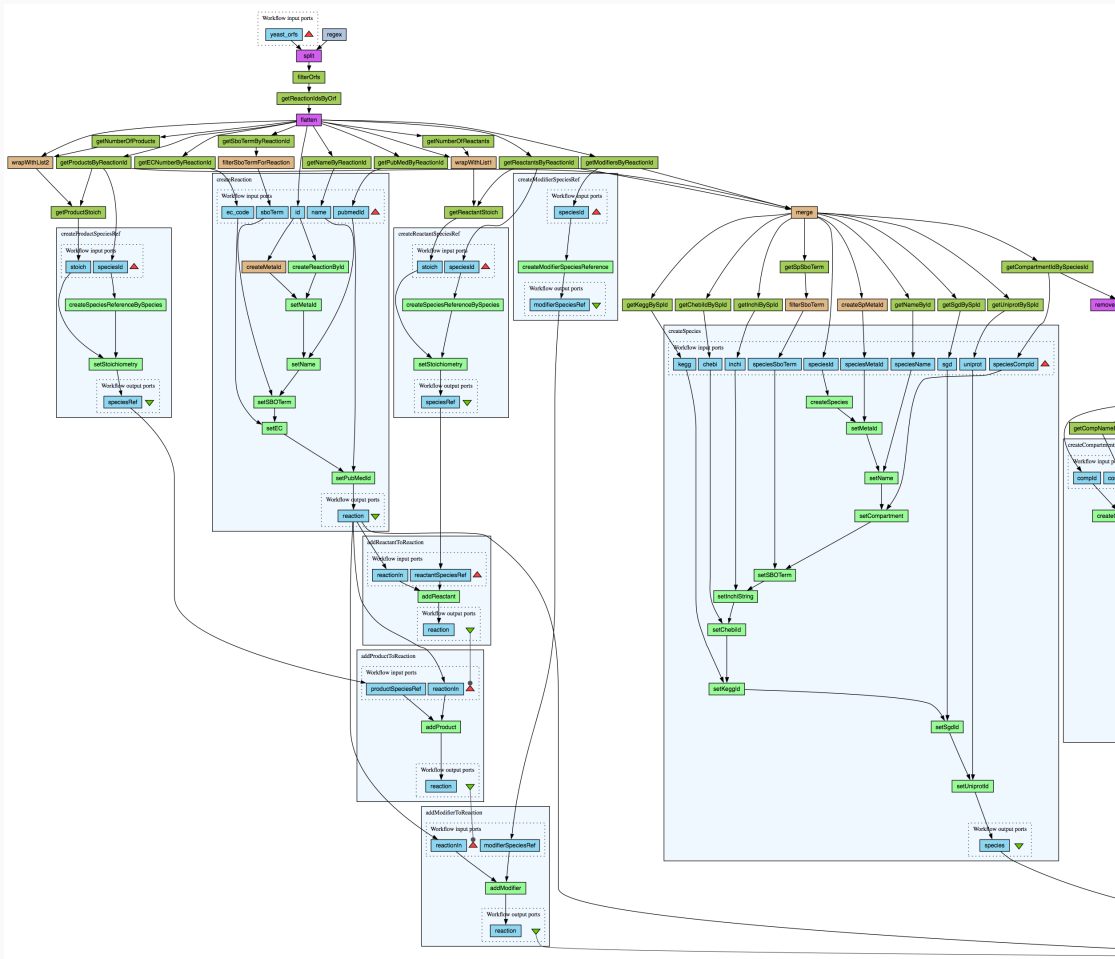






Tool 5: Automating computations with workflows





Tool 5: Automating computations with workflows

Workflows:

- Clearer high-level view
- Explicit composition of codes and data movement
- Safer sharing, reusing, and execution
- Notebooks are a variant that is both impoverished and richer
  - No simple/mature path from a notebook to a workflow

Examples:

- Galaxy, Kepler, Taverna, Pegasus, Collective Knowledge, VisTrails
- Light-weight: `make`, dask, drake, swift, `snakemake`, ...
- Hybrids: SOS-notebook, ...

Floating Points Rounding: the other root of all evil ?

- Every operation includes implicit rounding.

```
1  print(2.1-2.0 == 0.1)
```

```
False
```

Floating Points Rounding: the other root of all evil ?

- Every operation includes implicit rounding.

```
1  print(2.1-2.0 == 0.1)
```

```
False
```

- Unfortunately: round(round(a+b)+c) $\neq$ round(a+round(b+c))
  Hence, operation order matters. For a reproducible computation, operation order should be preserved!!! Which order is more relevant is an other debate 😉

Floating Points Rounding: the other root of all evil ?

- Every operation includes implicit rounding.

```
1  print(2.1-2.0 == 0.1)
```

```
False
```

- Unfortunately: round(round(a+b)+c) $\neq$ round(a+round(b+c))
  Hence, operation order matters. For a reproducible computation, operation order should be preserved!!! Which order is more relevant is an other debate 😉
- Numerical instability may be closer than you think [Rump, 1988]

$$f(x,y) = 333.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2)2 + 5.5y^8 + \frac{x}{2y}$$

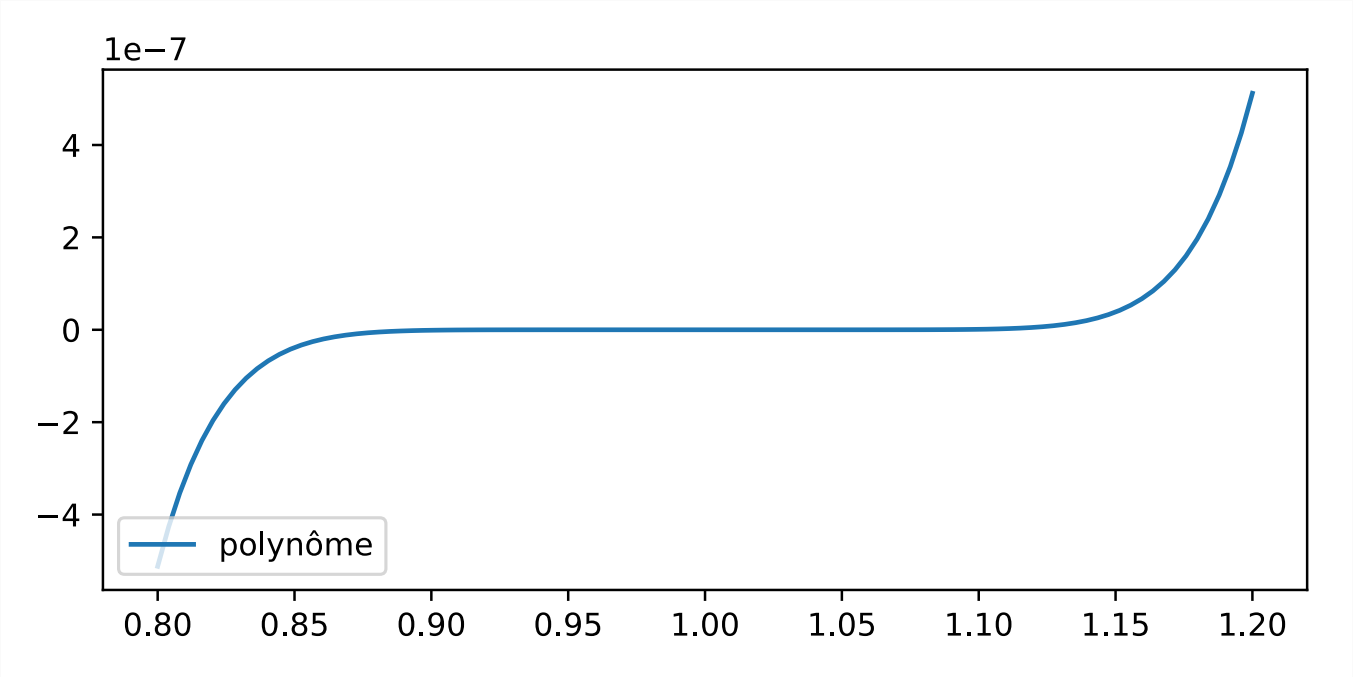
Evaluation of $f(77617.0, 33096.0)$

Single precision	1.172603
Double precision	1.1726039400531
Extended precision	1.172603940053178
MPFI (multiple precision interval arithmetic)	[-0.827396059946821368141165..., -0.827396059946821368141165...]

Courtesy of Christophe Denis

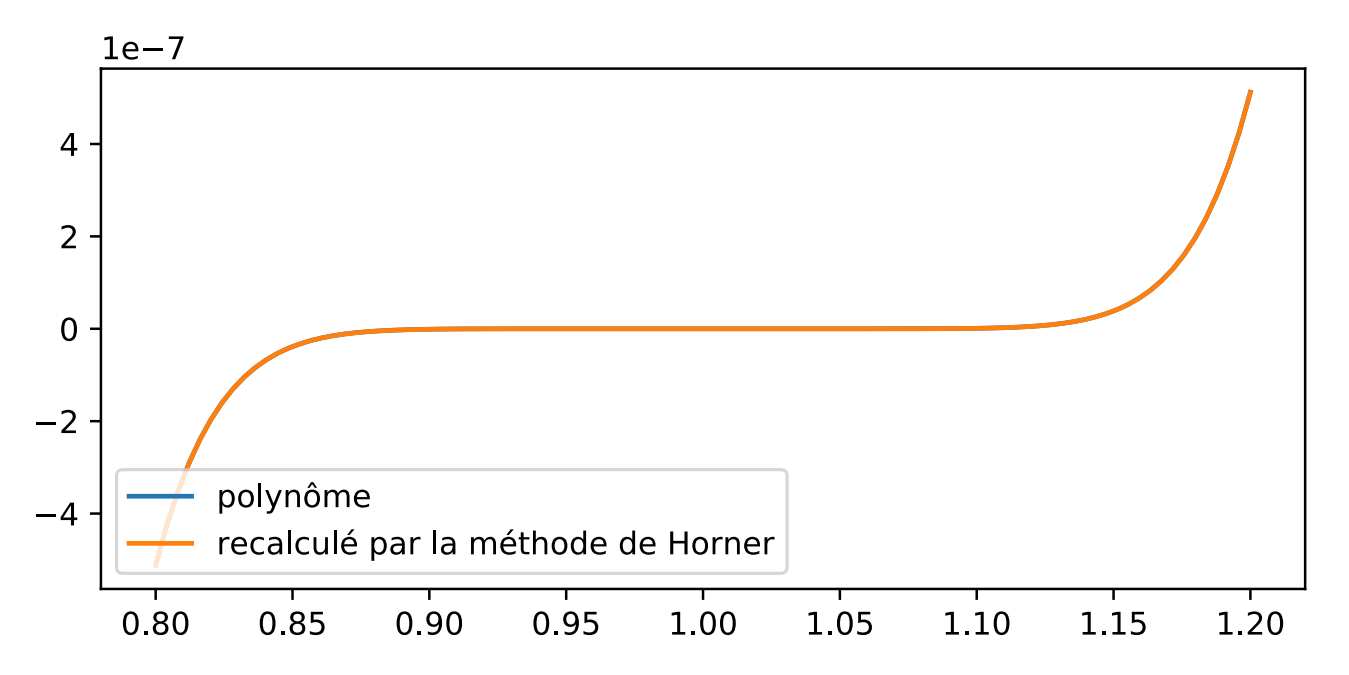
At scale (complex code + non-determinism), all this can become particularly harmful and painful.

All I care about is the algorithm output (FP)



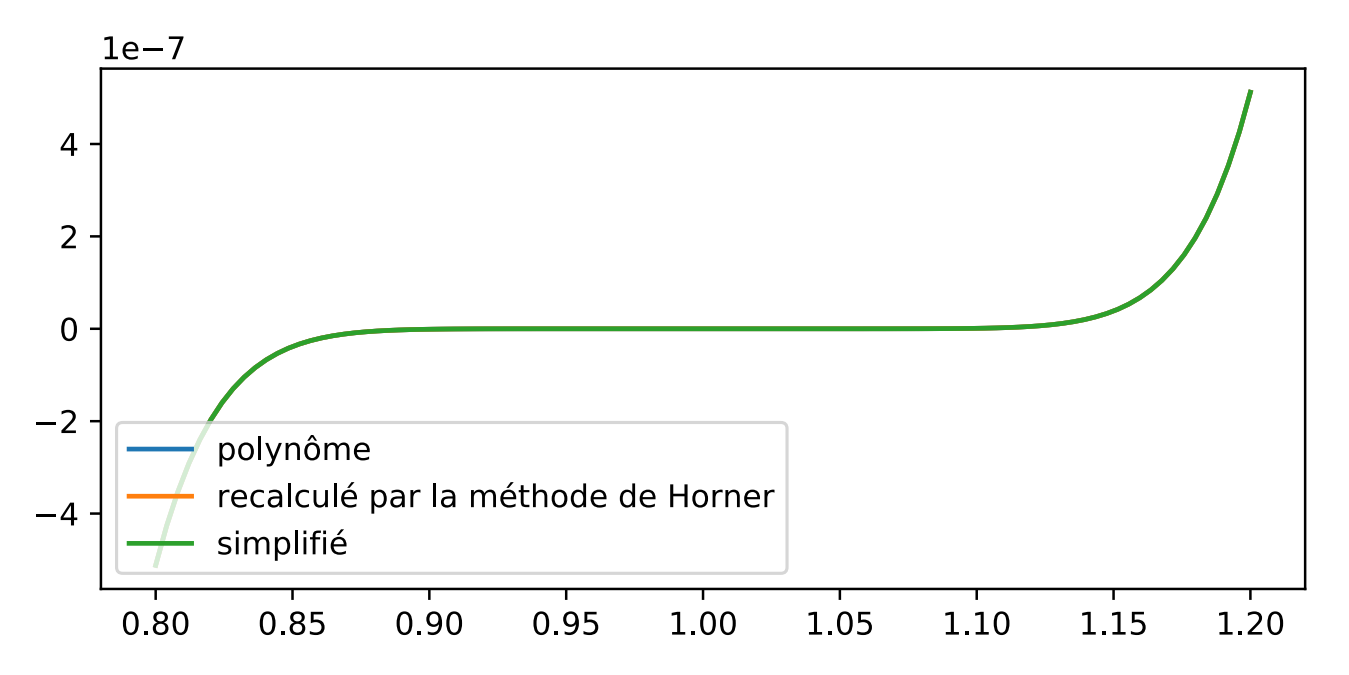

```
1  def polynome(x):
2      return x**9 - 9.*x**8 + 36.*x**7 - 84.*x**6 + 126.*x**5 \
3             - 126.*x**4 + 84.*x**3 - 36.*x**2 + 9.*x - 1.
```

Floating-point arithmetic



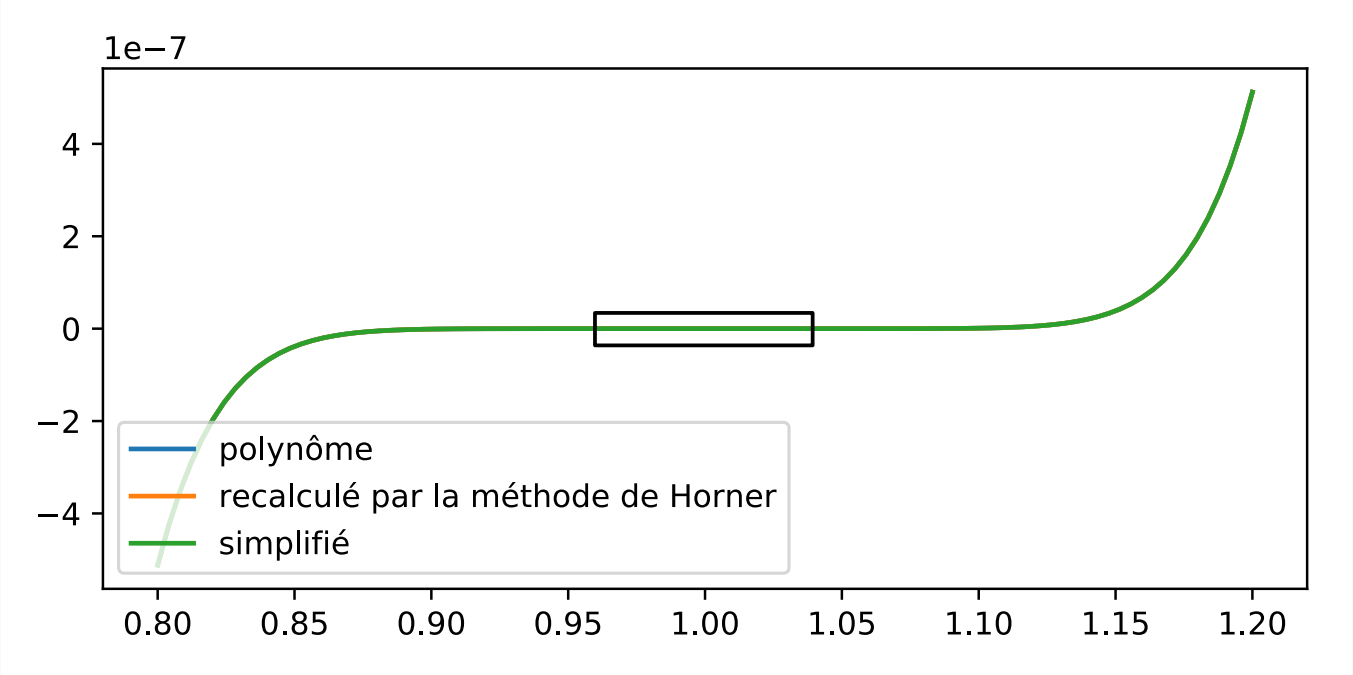

```
1  def horner(x):
2      return x*(x*(x*(x*(x*(x*(x*(x*(x - 9.) + 36.) - 84.) + 126.) \
3             - 126.) + 84.) - 36.) + 9.) - 1.
```

Floating-point arithmetic

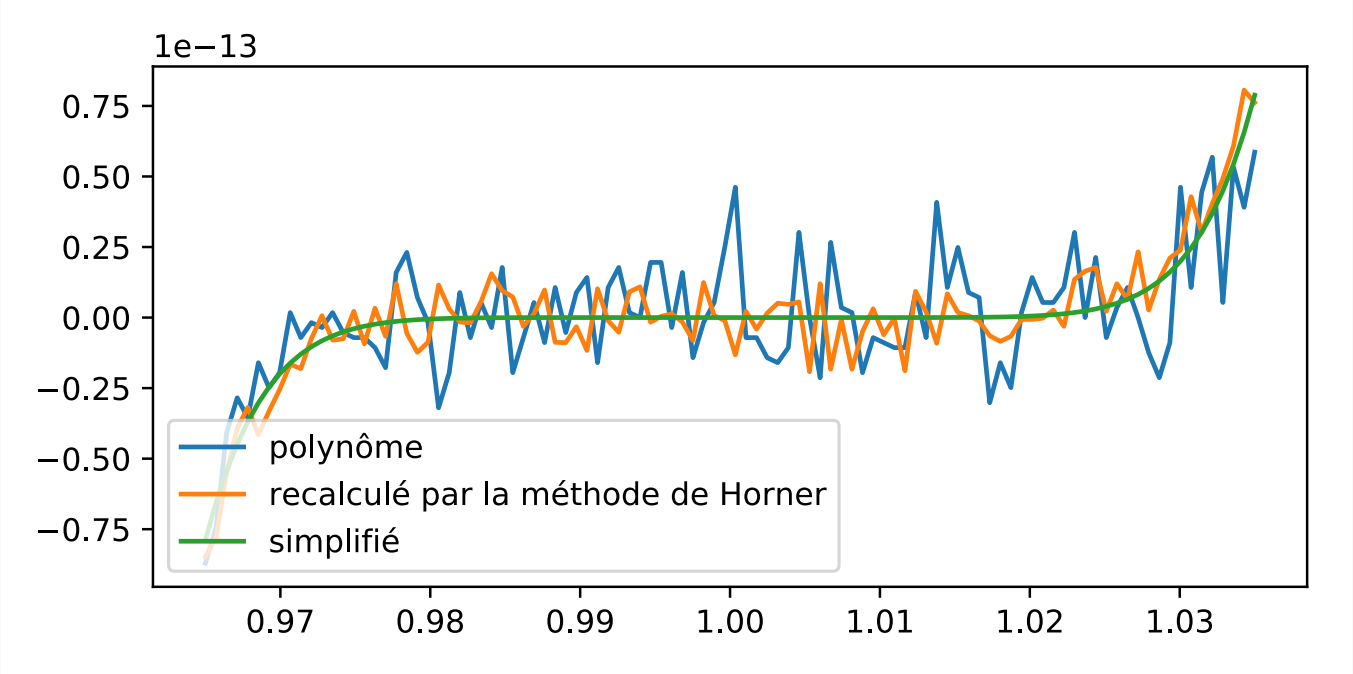



```
1  def simple(x):
2      return (x-1.)**9
3  # Easy! ;)
```

Floating-point arithmetic



Floating-point arithmetic

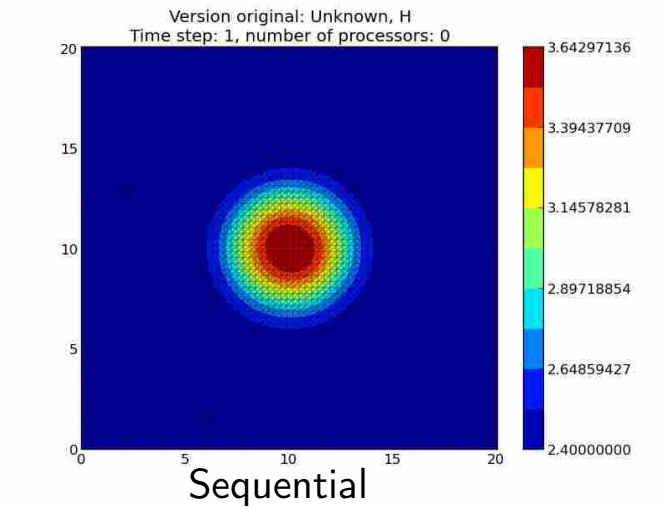


Telemac2D: the simplest gouttedo simulation

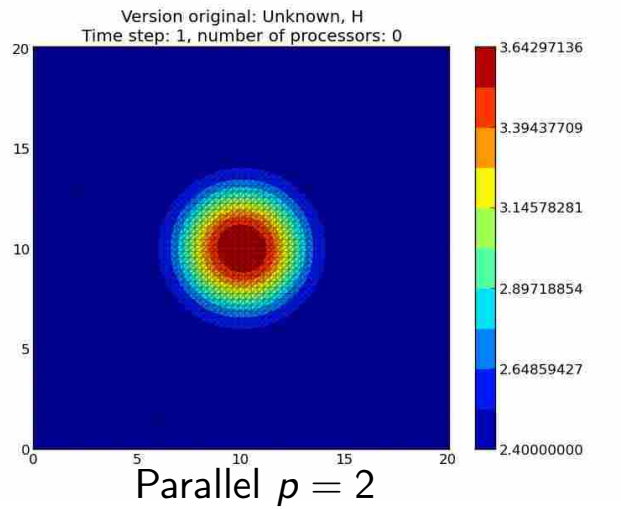
The gouttedo test case

- 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



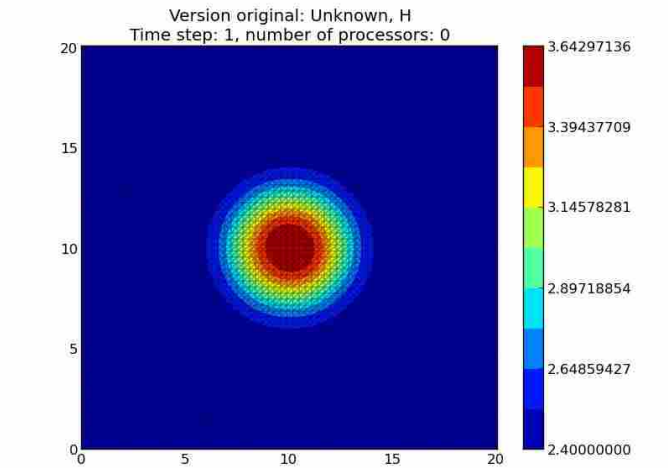
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

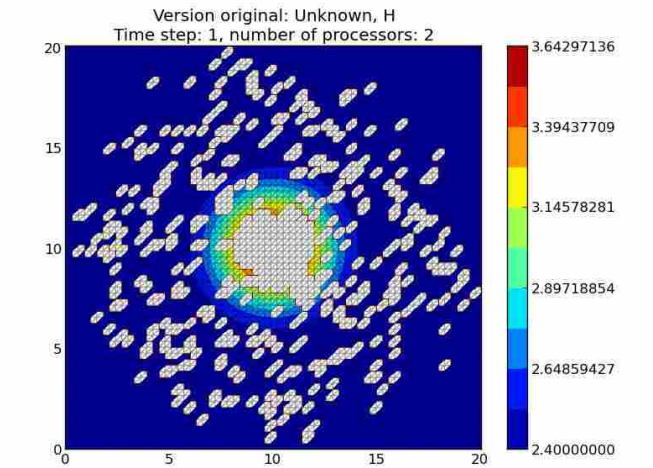
A white plot displays a non-reproducible value

Numerical reproducibility?

time step $= 1$



Sequential



Parallel $p = 2$

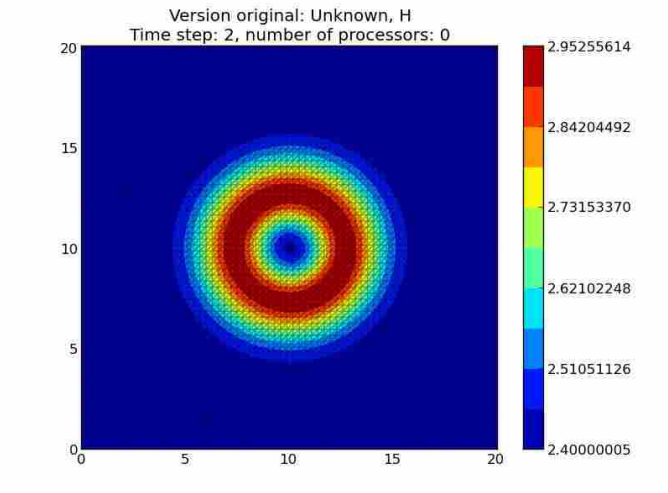
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

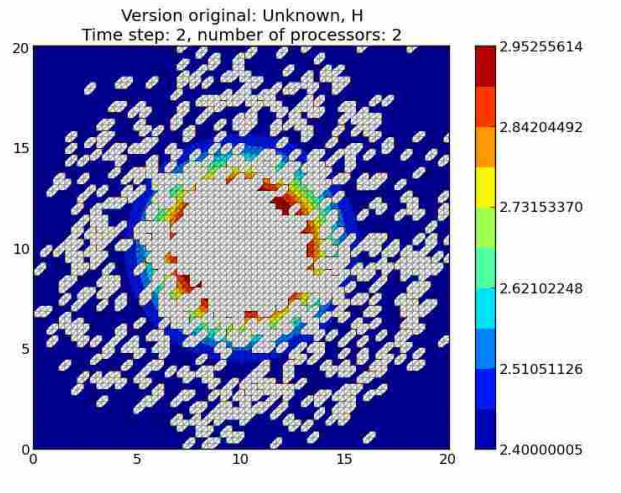
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 2



Sequential



Parallel $p = 2$

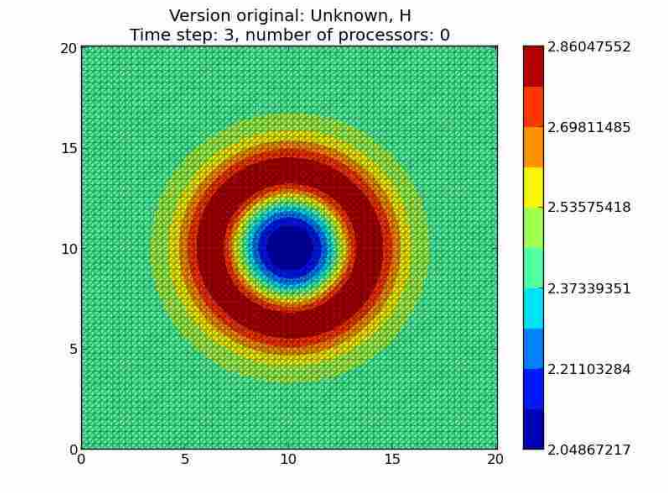
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

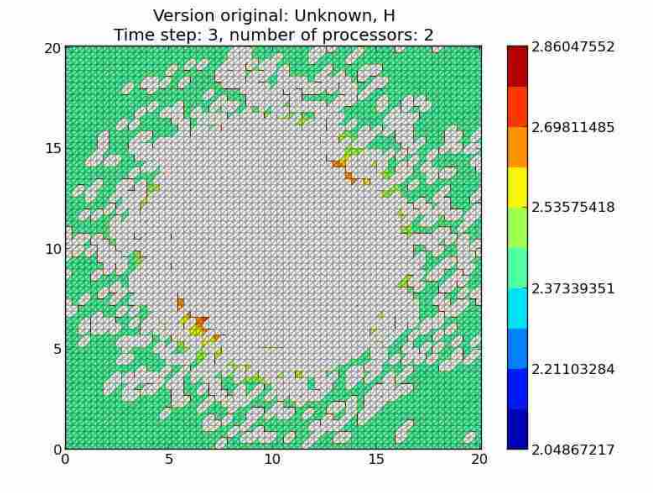
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 3



Sequential



Parallel $p = 2$

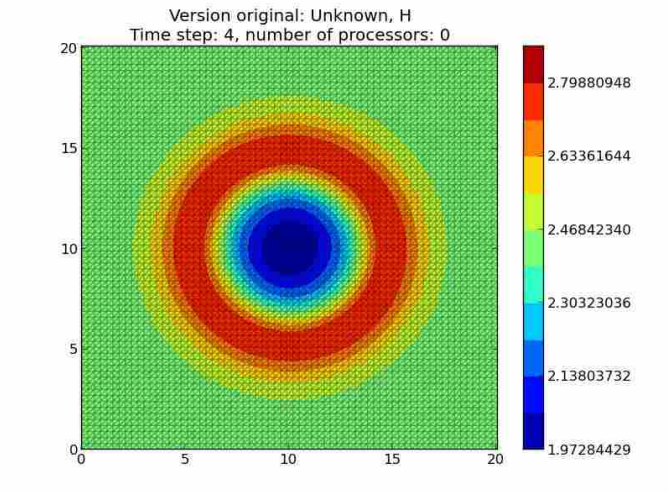
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

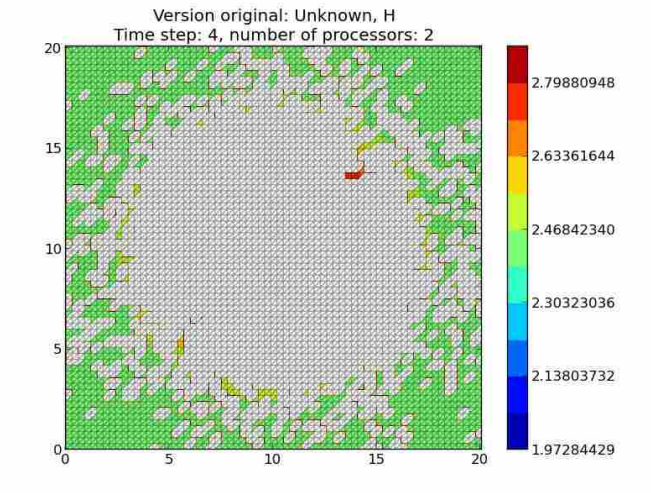
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 4



Sequential



Parallel $p = 2$

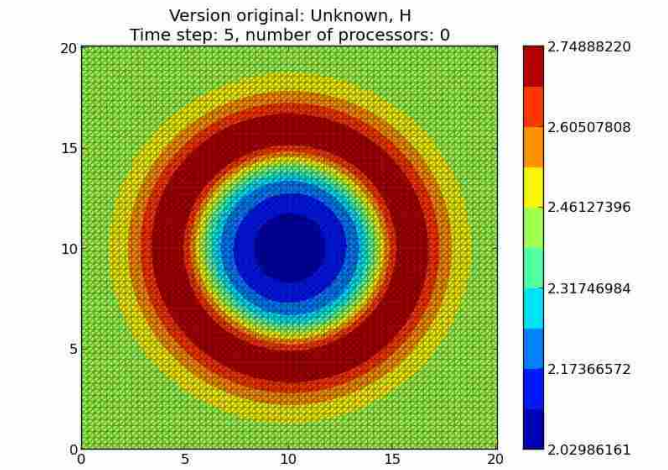
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

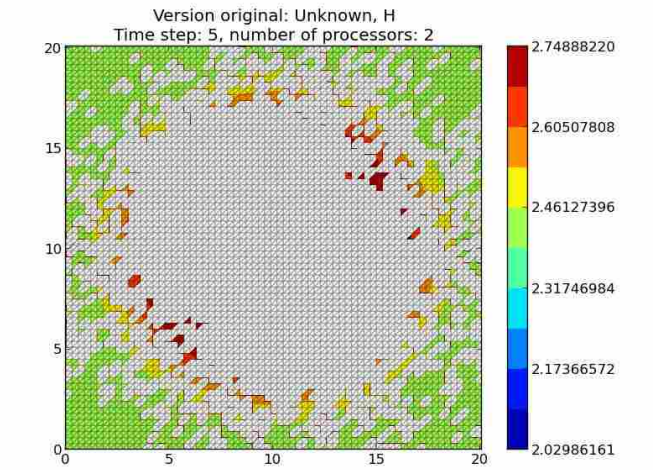
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 5



Sequential



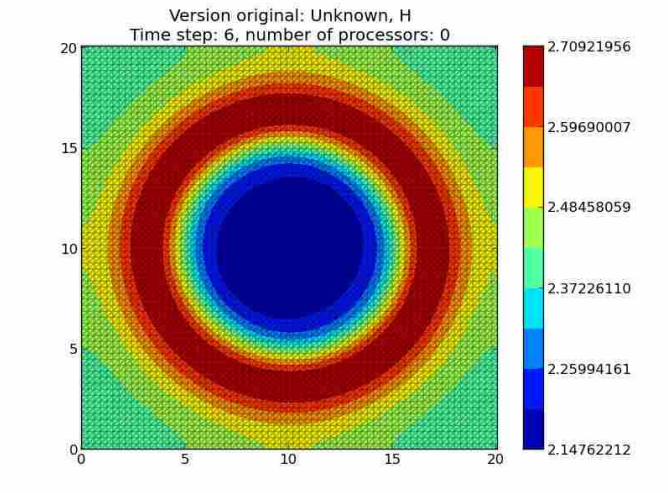
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

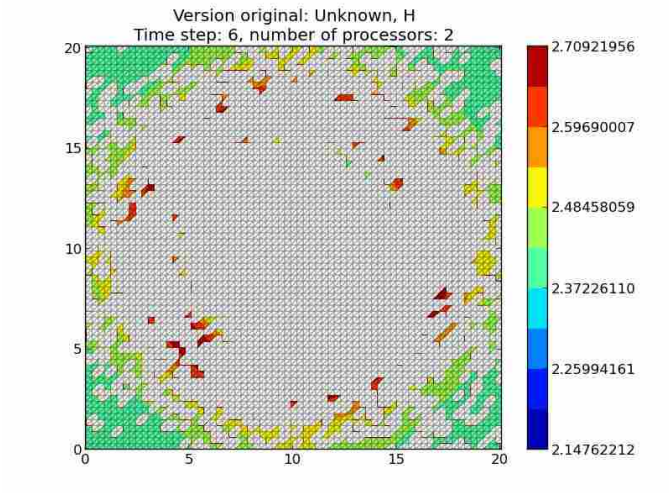
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 6



Sequential



Parallel $p = 2$

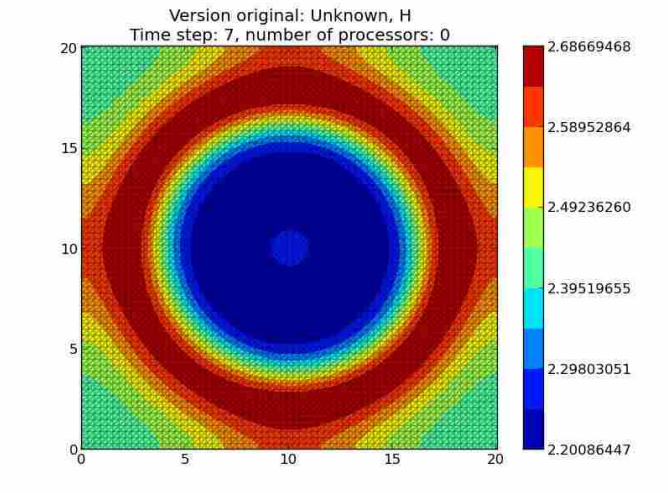
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

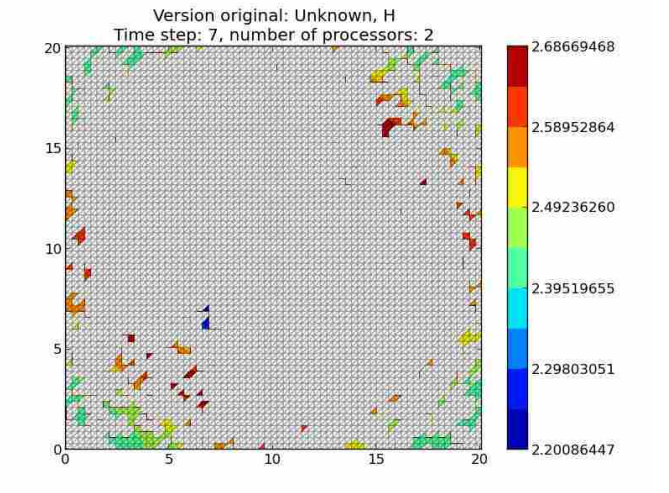
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 7



Sequential



Parallel $p = 2$

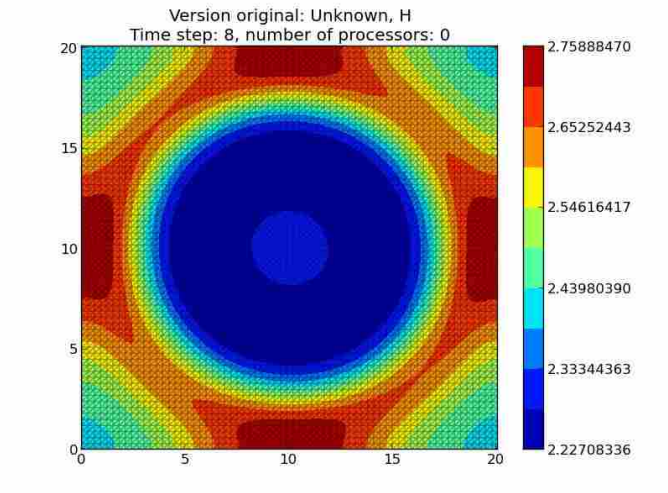
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

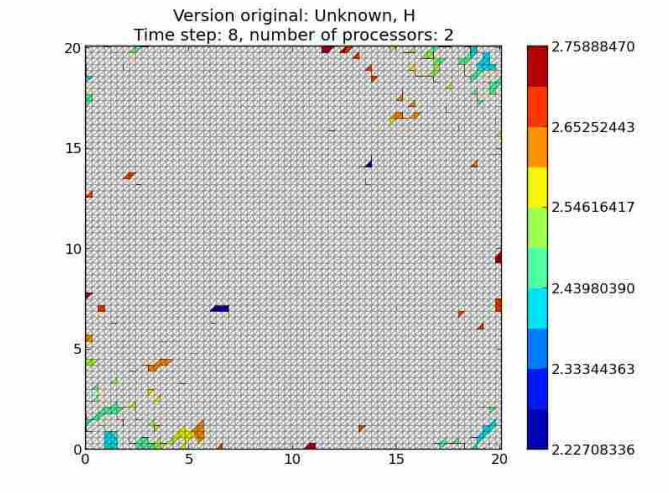
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 8



Sequential



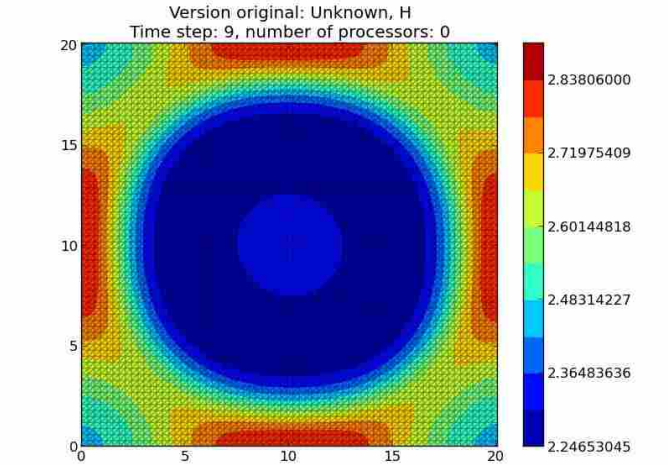
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

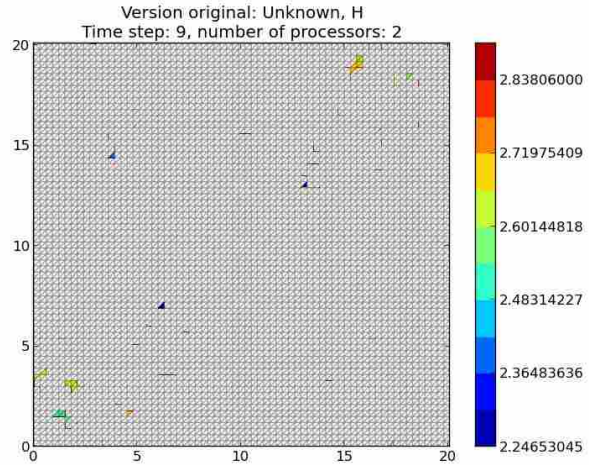
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 9



Sequential



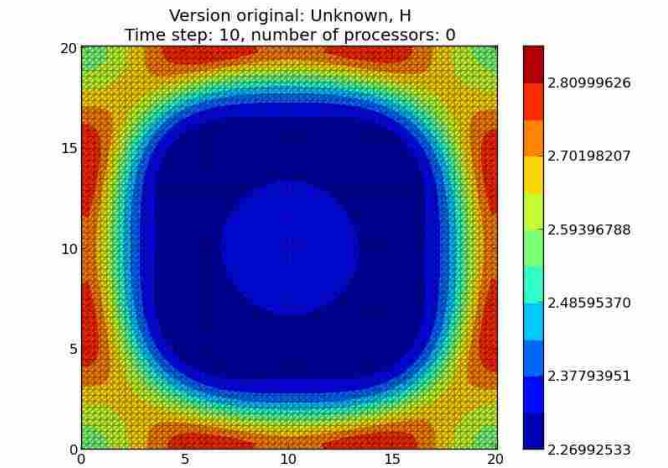
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

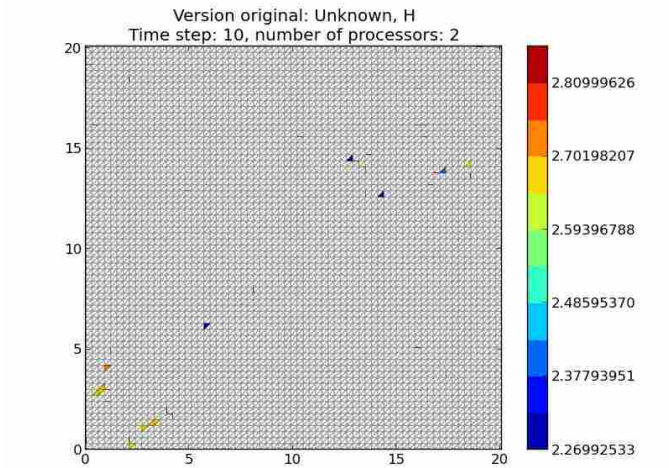
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 10



Sequential



Parallel $p = 2$

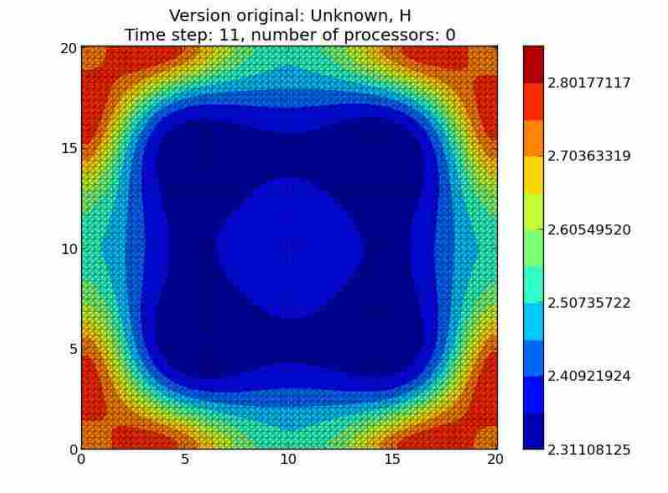
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

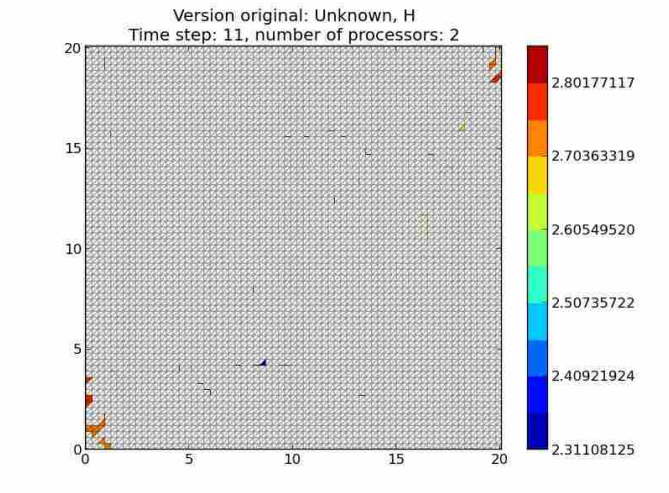
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 11



Sequential



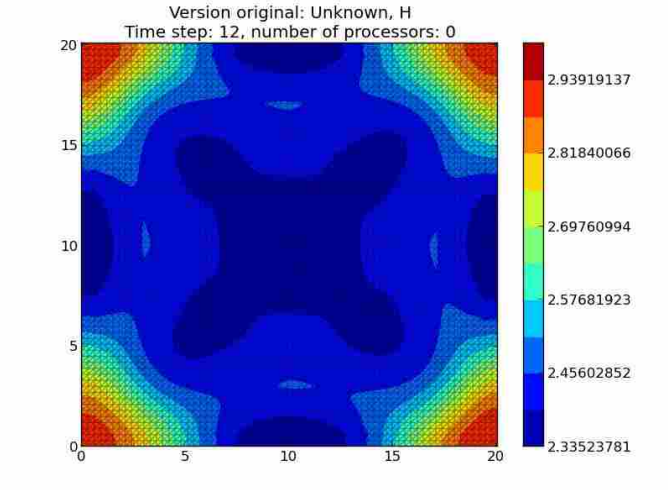
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

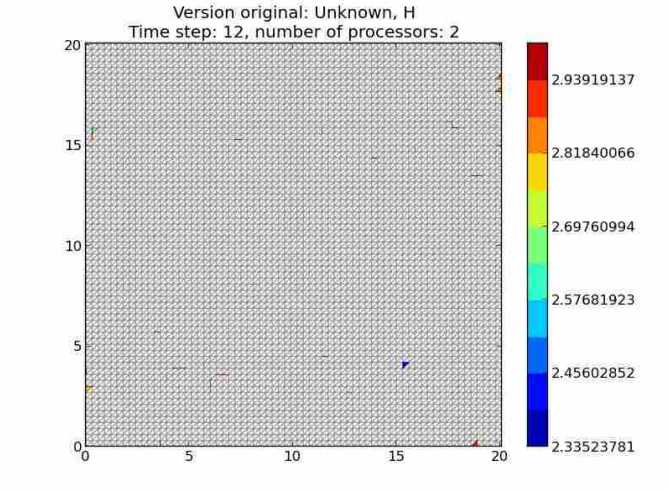
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 12



Sequential



Parallel $p = 2$

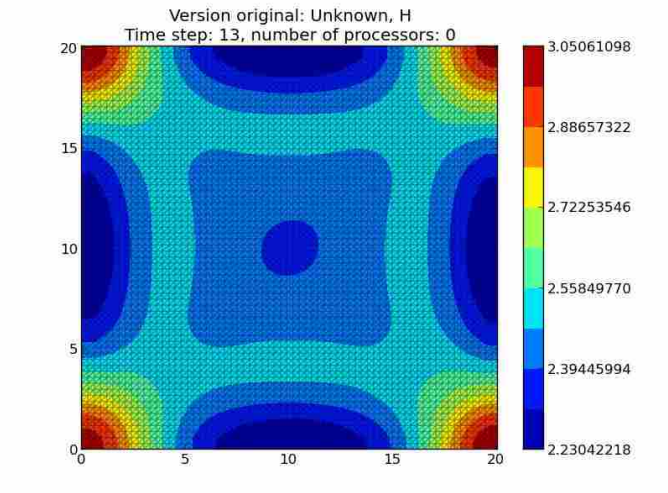
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

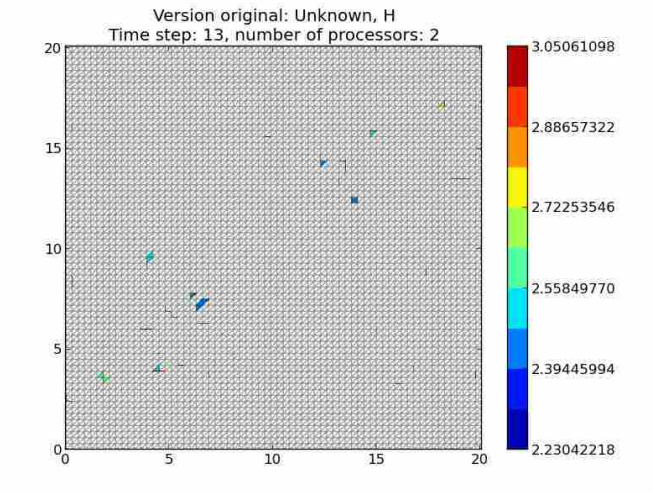
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 13



Sequential



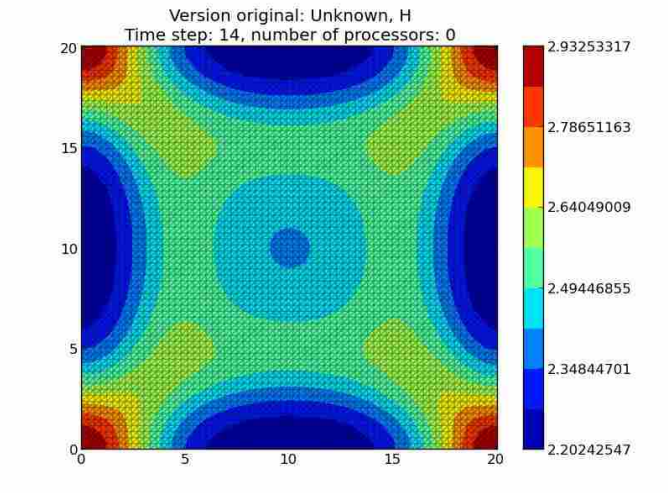
Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

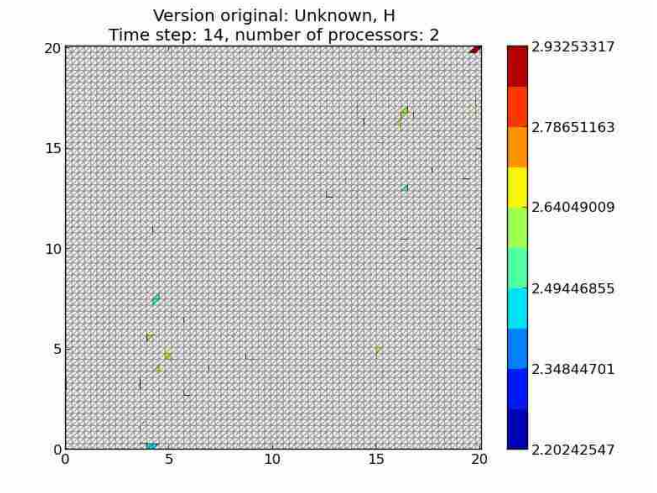
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 14



Sequential



Parallel $p = 2$

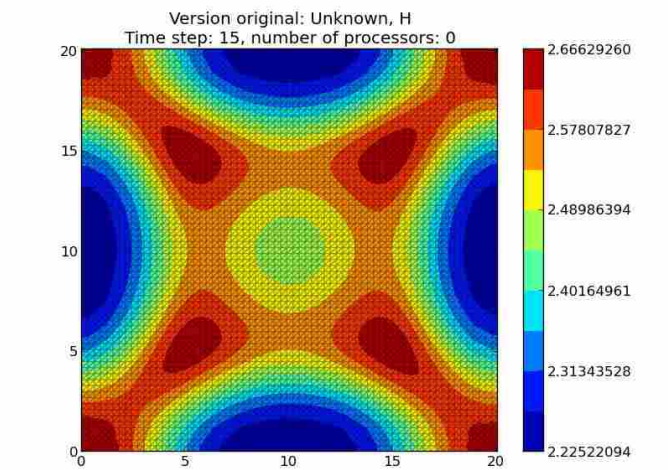
Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

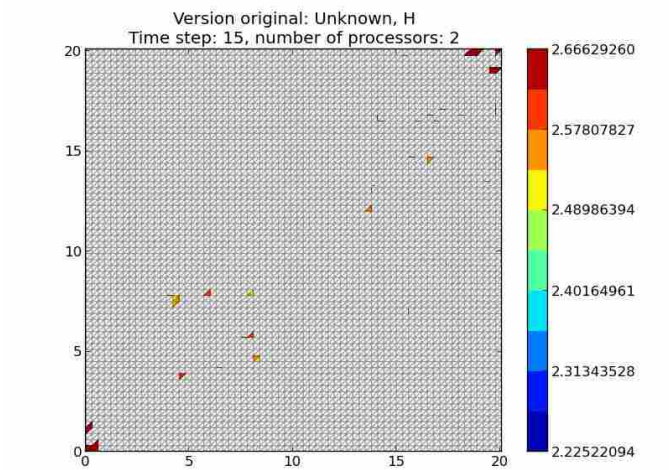
A white plot displays a non-reproducible value

NO numerical reproducibility!

time step = 15



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

Did I mention we have parallel machines nowadays?

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

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

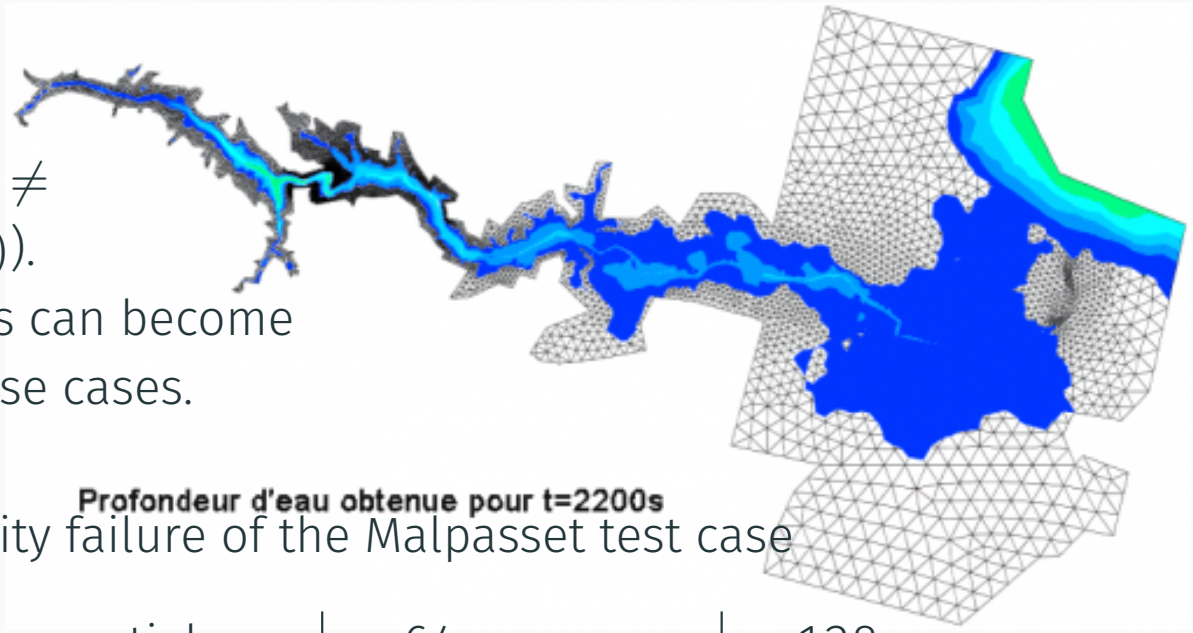


TABLE 1.1: Reproducibility failure of the Malpasset test case

	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.

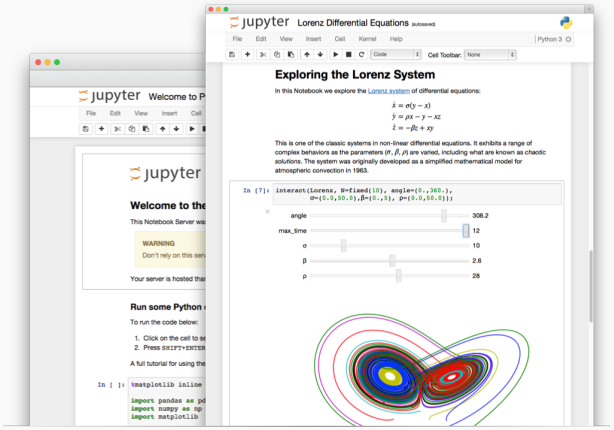
The whole chain needs to be revisited.

Courtesy of P. Langlois and R. Nheili

Conclusion and Advertising

Good research requires time and resources

Computation provenance: notebooks

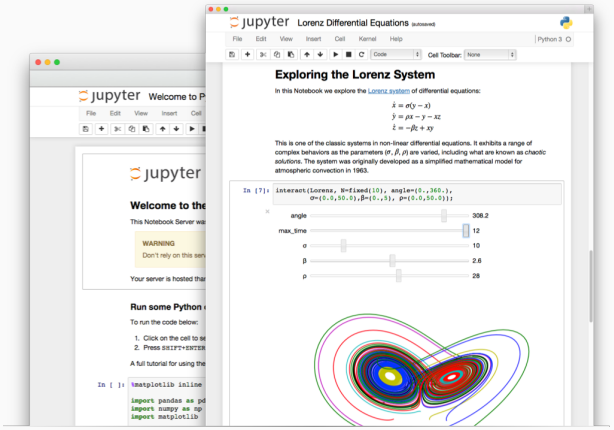


MOOC RR 1: Methodological principles for a transparent science

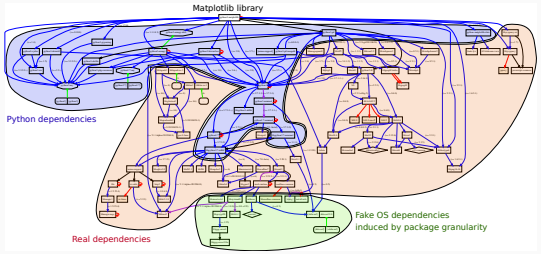
3rd Edition: March 2020 – … (22,800+)

Good research requires time and resources

Computation provenance: notebooks and workflows



Software environments



Sharing and Archiving

arXiv.org HAL archives-ouvertes.fr

zenodo

Software Heritage

GNU Make

snakemake

MOOC RR 1: Methodological principles for a transparent science

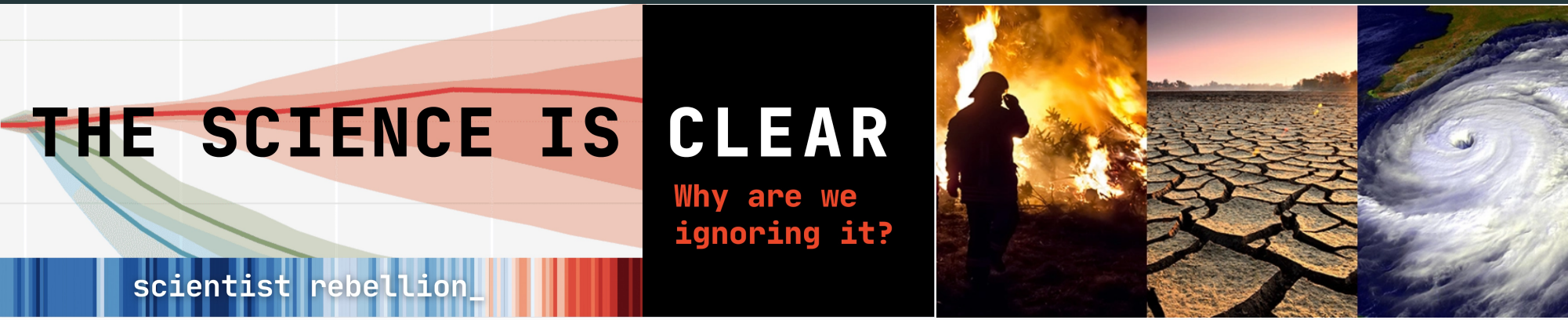
3rd Edition: March 2020 – … (22,800+)

MOOC RR 2: Practices and tools for managing computations and data

1st Edition: May 2024–Oct 2024

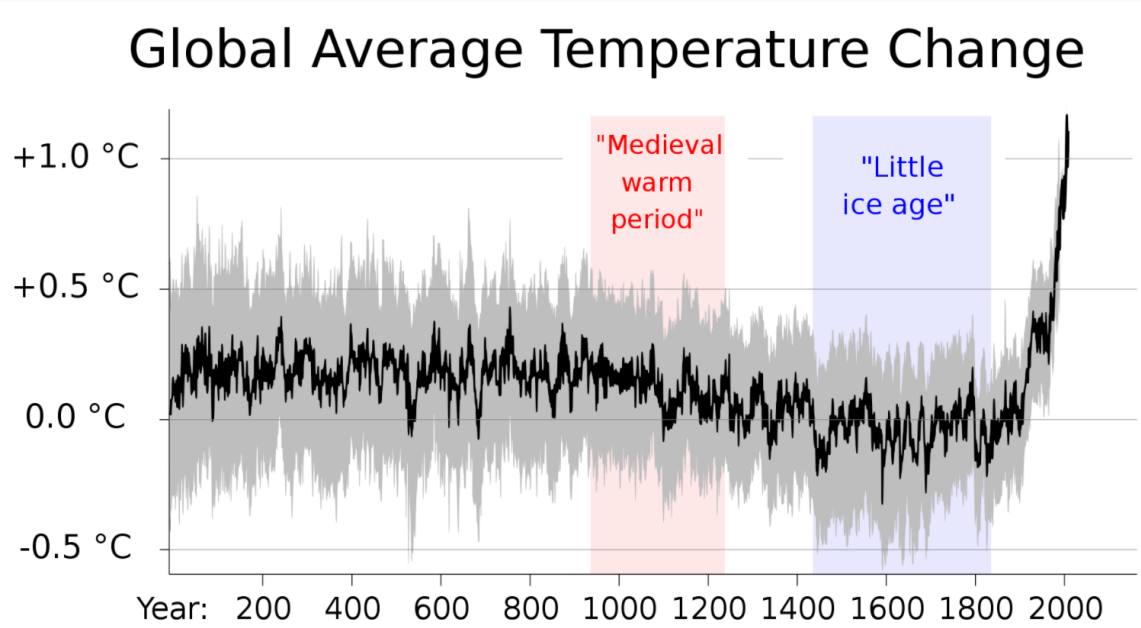
What's the point ?

The Elephant in the Room: Climate Change

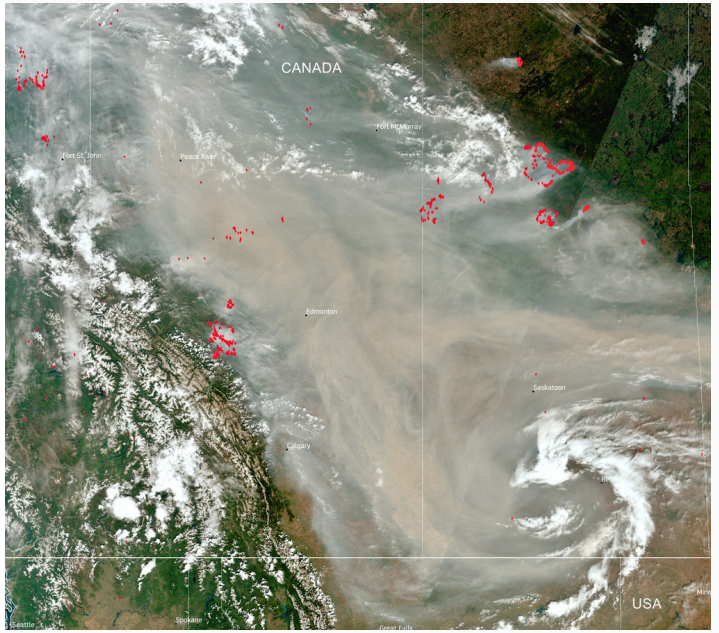


IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem

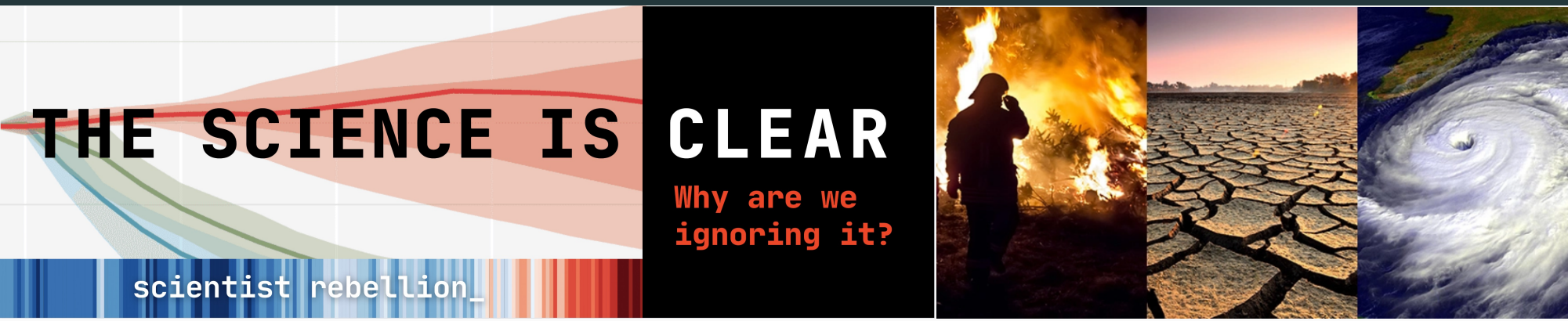


https://en.wikipedia.org/wiki/Global_temperature_record



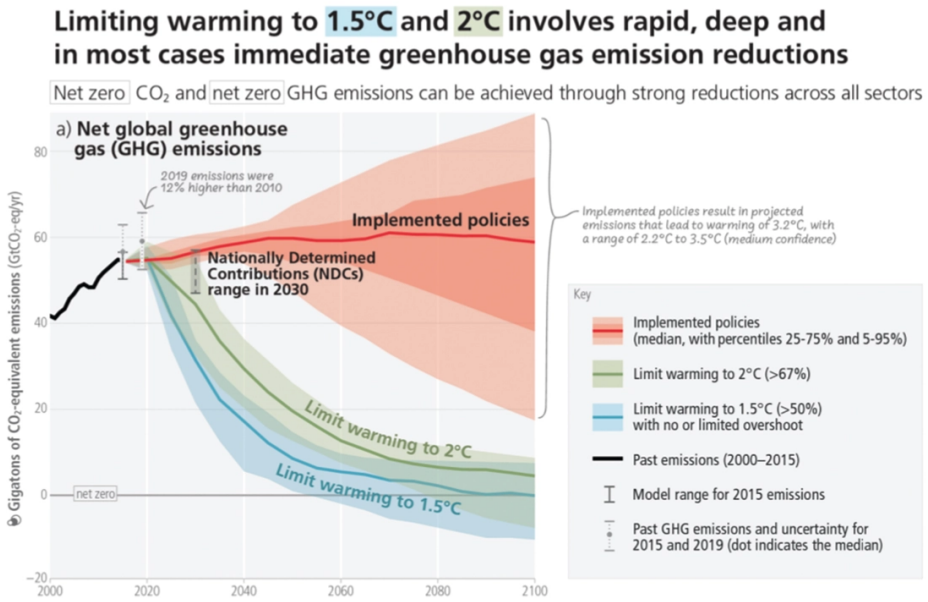
2023 Alberta wildfires (> 1 Mha)

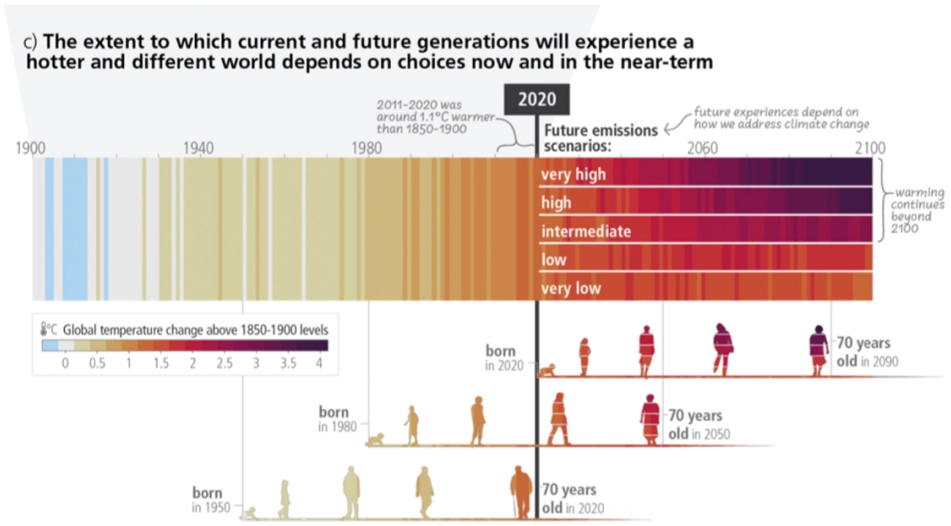
The Elephant in the Room: Climate Change



IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem
2. It is **entirely** due to human activity

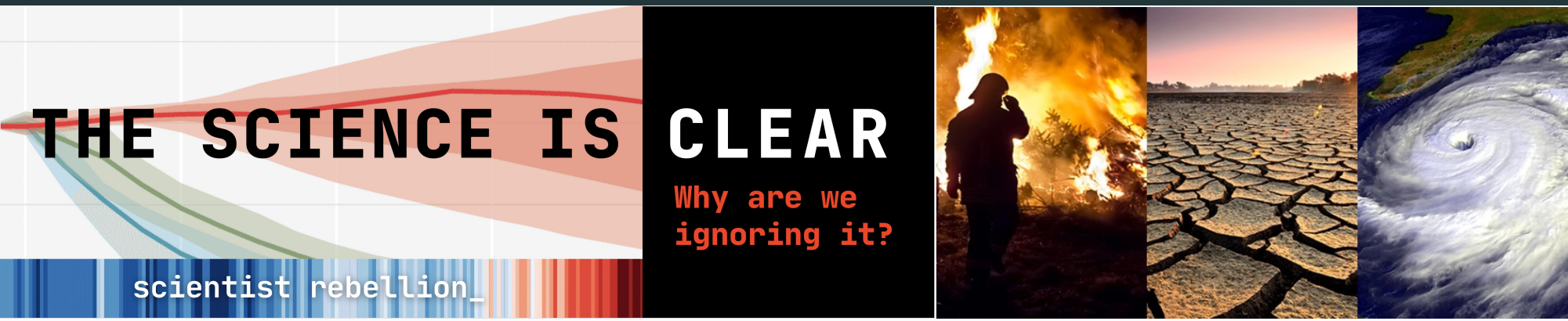




Paris Agreement'15 ∼ Net Zero by 2050

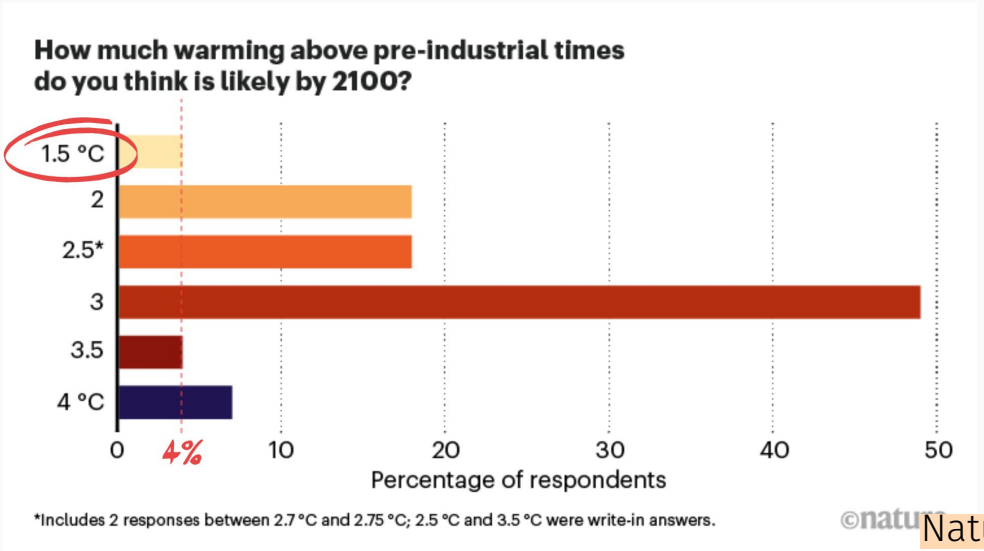
Latest IPCC report

The Elephant in the Room: Climate Change



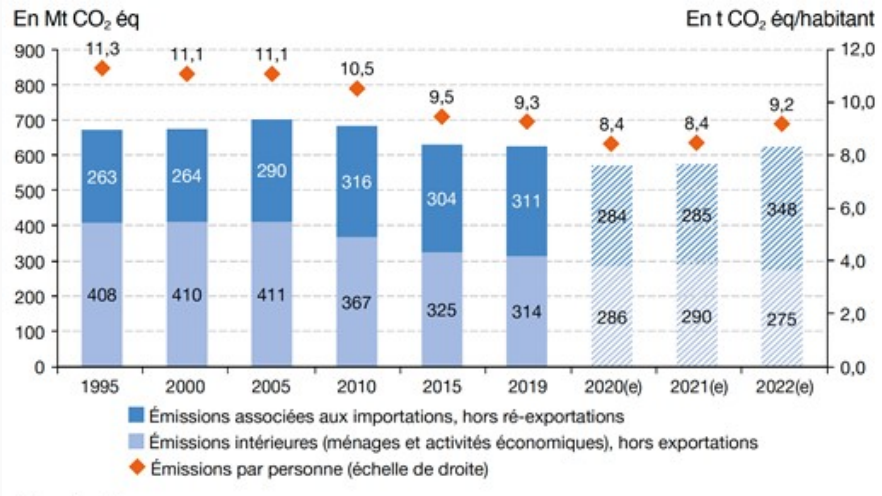
IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem
2. It is **entirely** due to human activity
3. *9 out of 10 IPCC scientists believe overshoot is likely*

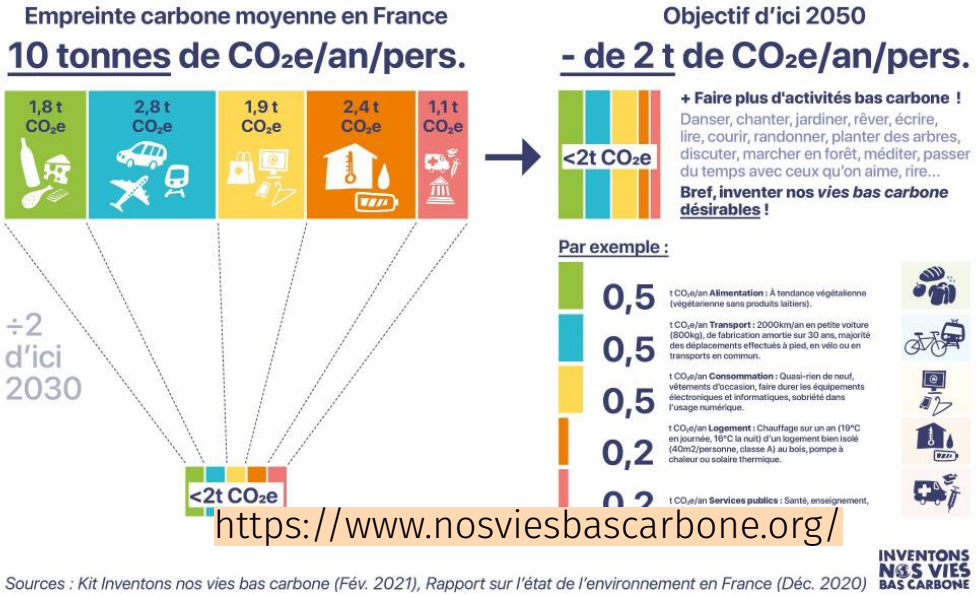


Nature survey, Nov. 2021

Put aside biodiversity loss, pollution, freshwater, land system change...

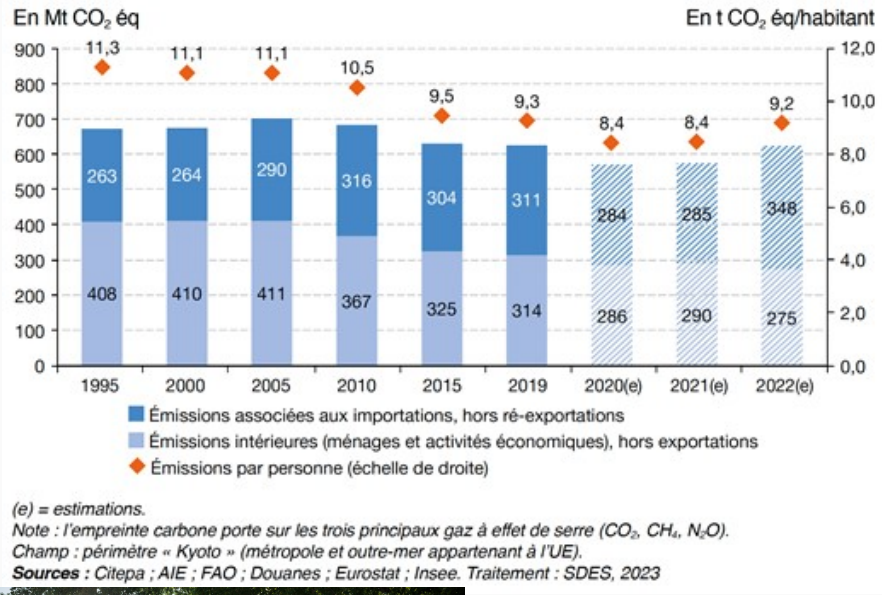


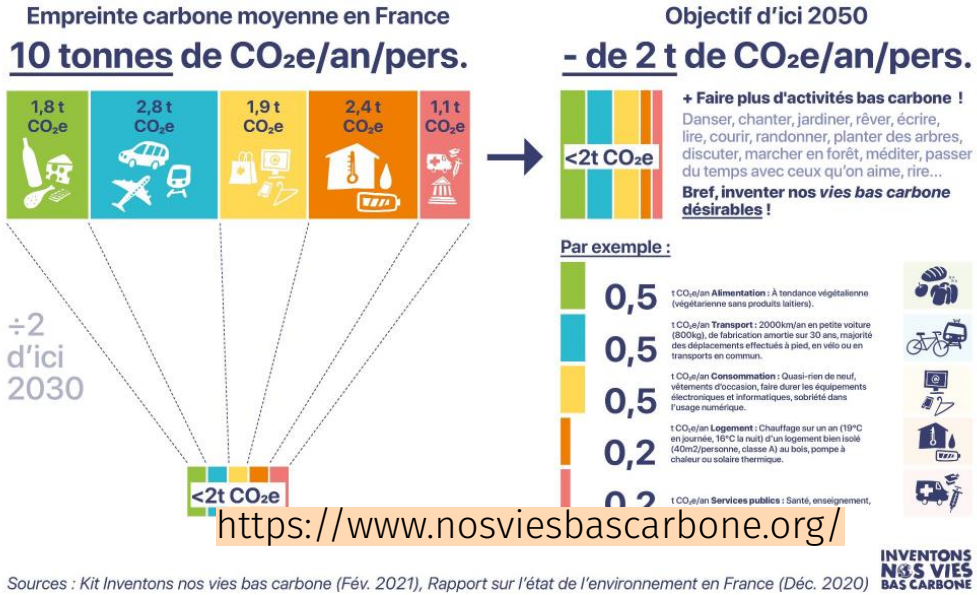
(e) = estimations.
Note : l'empreinte carbone porte sur les trois principaux gaz à effet de serre (CO_2, CH_4, N_2O).
Champ : périmètre « Kyoto » (métropole et outre-mer appartenant à l'UE).
***Sources :** Citepa ; AIE ; FAO ; Douanes ; Eurostat ; Insee. Traitement : SDES, 2023*



Sources : Kit Inventons nos vies bas carbone (Fév. 2021), Rapport sur l'état de l'environnement en France (Déc. 2020)

Put aside biodiversity loss, pollution, freshwater, land system change...



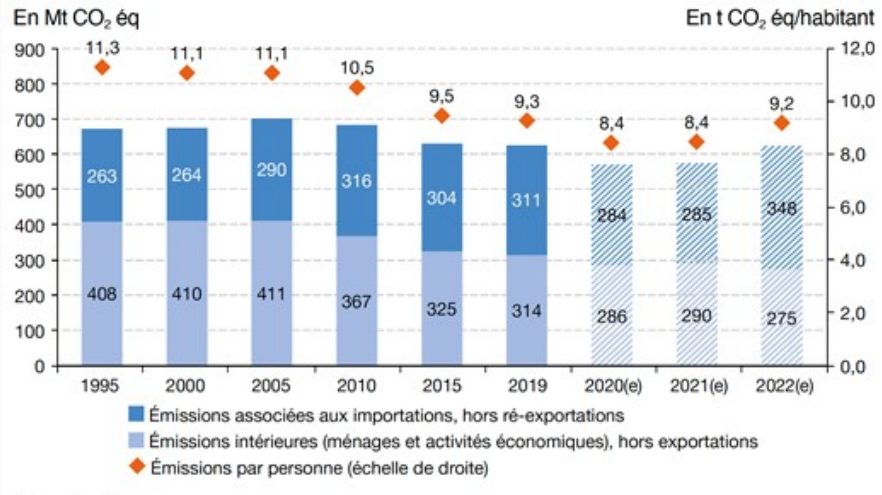


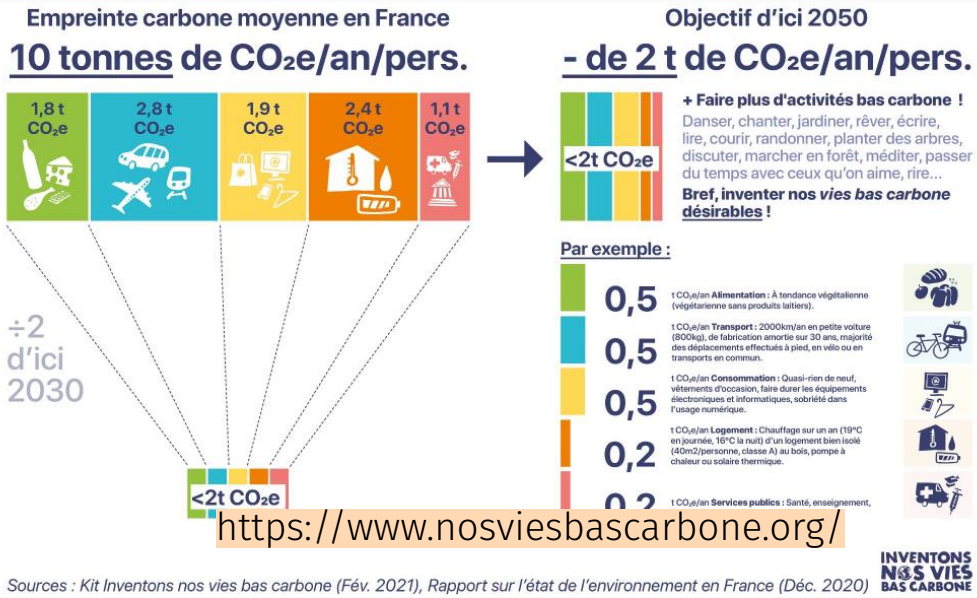
https://www.nosviesbascarbone.org/

French government response

- *Verdissement de l'industrie: « pause » sur les normes environnementales*
- *Loi de programmation militaire (+41%)*
- *Nous devons préparer la France à une élévation de la température de 4 °C*
- Academia ? PEPR 5G, Cloud, NUMPEX, Quantique, IA, Agroécologie et numérique

Put aside biodiversity loss, pollution, freshwater, land system change...





https://www.nosviesbascarbone.org/

French government response

- *Verdissement de l'industrie: « pause » sur les normes environnementales*
- *Loi de programmation militaire (+41%)*
- *Nous devons préparer la France à une élévation de la température de 4 °C*
- Academia ? PEPR 5G, Cloud, NUMPEX, Quantique, IA, Agroécologie et numérique

~~Several~~ scenarios on the table

- Energy optimization/saving ≠ sobriety and frugality
- What will research/CS look like/be used for in such a world?