

Reproducible Research for Computer Scientists

Lucas M. Schnorr

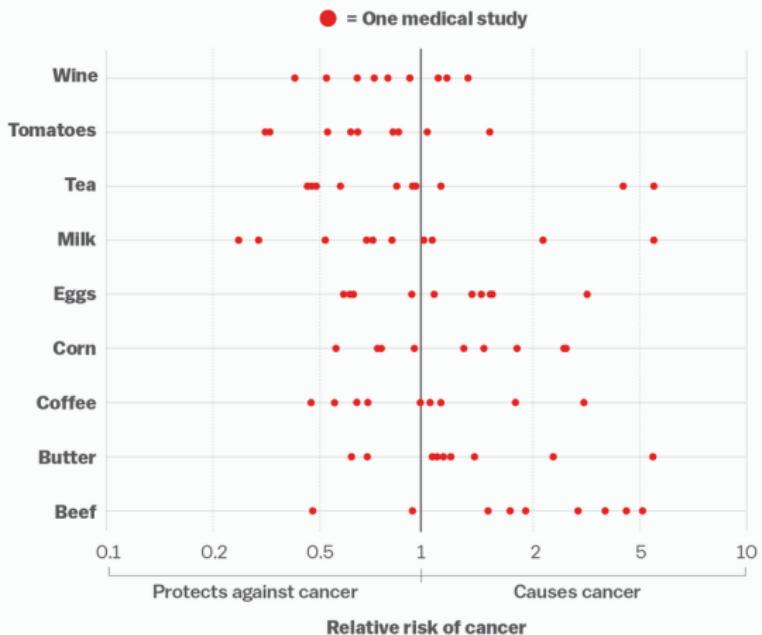
Comp. Syst. Perf. Analysis

Inconsistencies

Is everything we eat associated with cancer? A systematic cookbook review, Schoenfeld and Ioannidis, *Amer. Jour. of Clinical Nutrition*, 2013.

Inconsistencies

Everything we eat both causes and prevents cancer



SOURCE: Schoenfeld and Ioannidis, *American Journal of Clinical Nutrition*

Vox

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

Los Angeles Times | BUSINESS

LOCAL U.S. WORLD BUSINESS SPORTS ENTERTAINMENT HEALTH STYLE TRAVEL

Science has lost its way, at a big cost to humanity

Researchers are rewarded for splashy findings, not for double-checking accuracy. So many scientists looking for cures to diseases have been building on ideas that aren't even true.

Science AAAS.org FEEDBACK HELP LIBRARIANS All Science Journals Guest Search Type...
AAAS NEWS SCIENCE JOURNALS CAREERS MULTIMEDIA COLLECTIONS GUEST ALERTS ACCESS REGISTRATION

Science Home Current Issue Previous Issues Reference Express Science Products My Science About the Journal

Science Magazine > 12 January 2013 > McNutt, 343 (6168): 229

Science Views - Summary Full Text Full Text (PDF)

EDITORIAL Reproducibility Marcia McNutt

Science advances on a foundation of trusted approach that scientists use to gain confidence community was shaken by reports that a result not reproducible. Because confidence in results is fundamental to the scientific method. Science. For preclinical studies (one of the top recommendations of the U.S. National Institute increasing transparency.* Authors will indicate handling (such as how to deal with outliers), will ensure a sufficient signal-to-noise ratio, when experimenter we blind to the conduct of the e-guidelines.

TheScientist EXPLORING LIFE. INSPIRING INNOVATION

NIH Tackles Irreproducibility

The federal agency speaks out about how to improve the quality of scientific research.

By Jef Akst | January 28, 2014

Announcement: Reducing our Irreproducibility | Nature News & Comment

nature.com Sitemap Login Register

nature International weekly journal of science

Home News & Comment Research Careers & Jobs Current Issue Archive
Archive > Volume 496 > issue 7448 > Editorial > Article

NATURE | EDITORIAL

Announcement: Reducing our Irreproducibility

24 April 2013

PDF Rights & Permissions

Over the past year, Nature has published a string of articles that reliability and reproducibility of published research (collected an

NATURE | EDITORIAL

Must try harder

Nature 483, 509 (29 March 2012) | doi:10.1038/483509a
Published online 28 March 2012

PDF Citation Reprints Rights & permissions Article metrics

Too many sloppy mistakes are creeping into scientific papers. Lab heads must look more rigorously at the data — and at themselves.

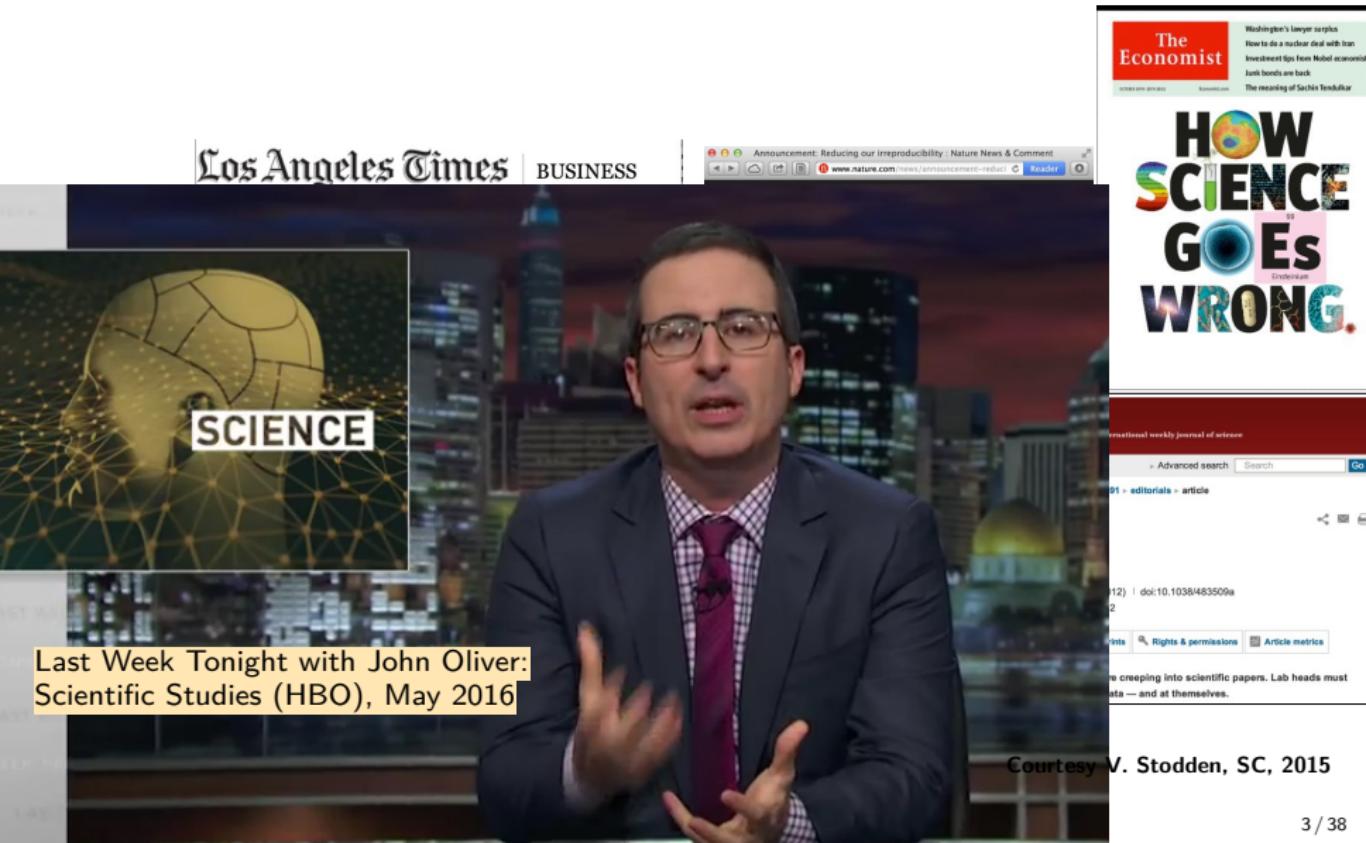
The Economist Washington's lawyer serpits
How to do a nuclear deal with Iran
Investment tips from Nobel economist
Junk bonds are back
The reasoning of Sachin Tendulkar

HOW SCIENCE GOES WRONG.

Courtesy V. Stodden, SC, 2015

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



Austerity in Fiscal Policy

2010 "gross debt [...] exceeding 90 percent of the economy has a significant negative effect on economic growth"

– Reinhart et Rogoff: *Growth in a Time of Debt*

2013 While using RR's working spreadsheet, we identified coding errors, selective exclusion of available data, and unconventional weighting of summary statistics.

– Herndon, Ash and Pollin

combining data across centuries, exchange rate regimes, public and private debt, and debt denominated in foreign currency as well as domestic currency

– Wray

For 3 years, austerity was not presented as an option but as a necessity.

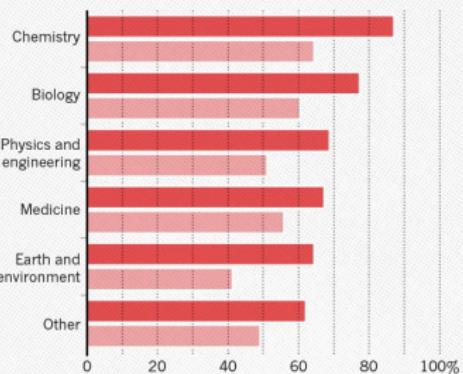
Yet, a scientific debate has at least been possible.

Why are scientific studies so difficult to reproduce?

HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.

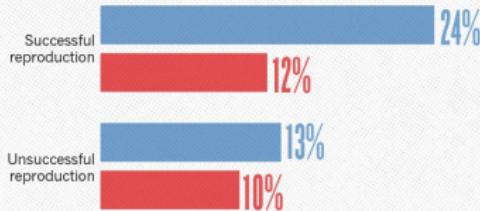
- Someone else's
- My own



HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

Although only a small proportion of respondents tried to publish replication attempts, many had their papers accepted.

- Published
- Failed to publish



Number of respondents from each discipline:

Biology 703, Chemistry 106, Earth and environmental 95,
Medicine 203, Physics and engineering 236, Other 233

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: 1+ million articles per year!

Methodological or 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

Wrap-up

- Oncology : "more than half studies published in prestigious journals cannot be reproduced in industrial labs"
- Psychology : "replicating a hundred of major articles: only one third of coherent results"



Whistle blowers, sick institutions, broken system, ?..

No. Questionning previous work is part of the scientific process

Just like honesty, rigor and transparency...

Risks science credibility put into question. No more difference with crooks!

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

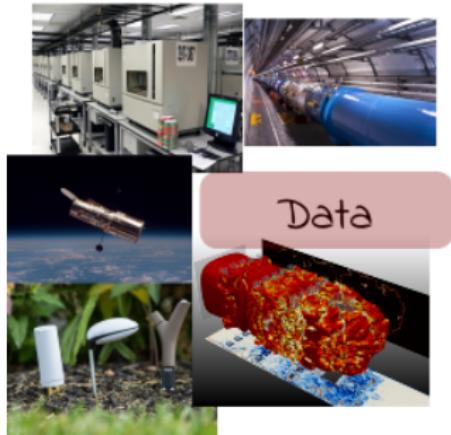
Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Computational science!

Authors



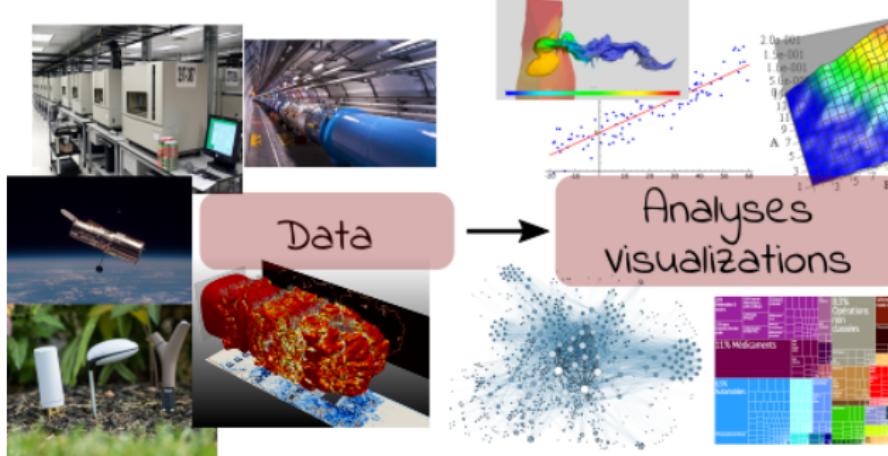
Data

Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...

– Nobel Comity (Chemistry), 2013

Computational science!

Authors

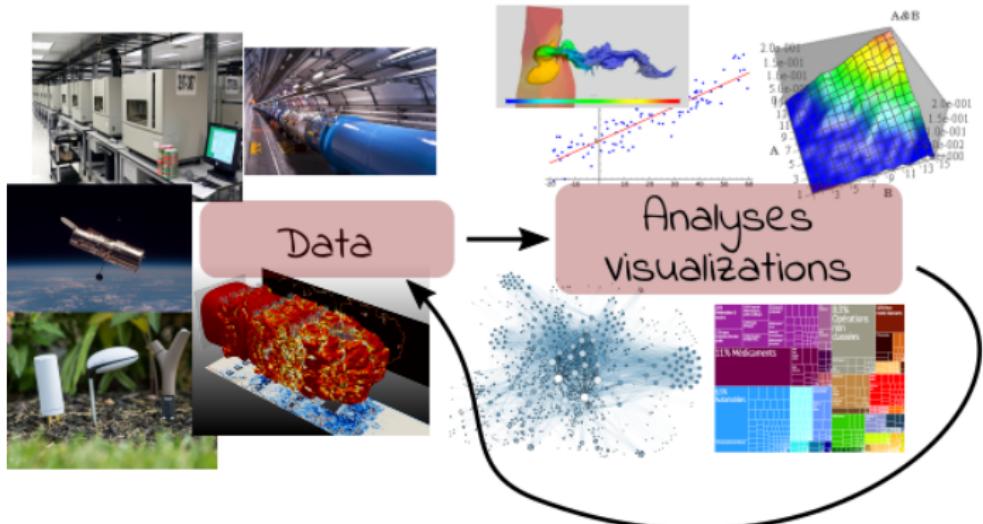


Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...

– Nobel Comity (Chemistry), 2013

Computational science!

Authors

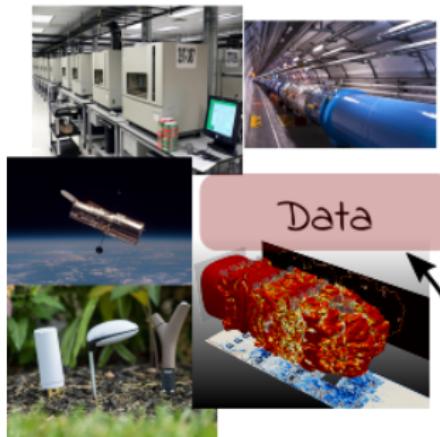


Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...

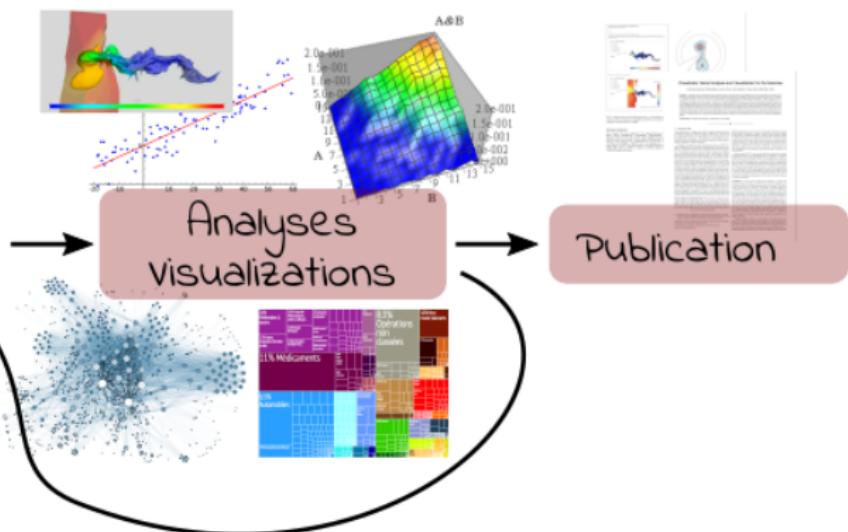
– Nobel Comity (Chemistry), 2013

Computational science!

Authors



Data

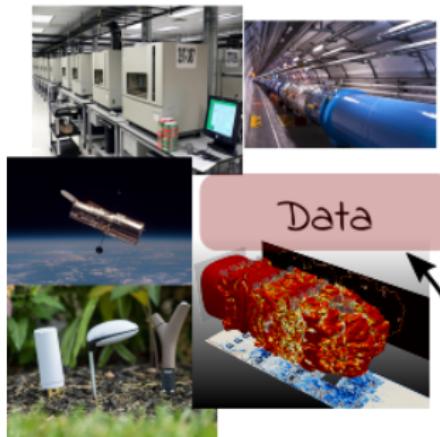


Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...

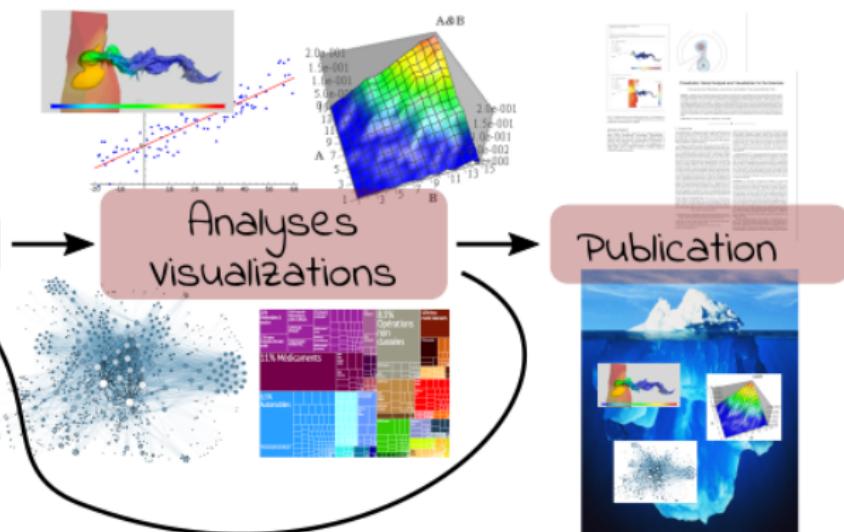
– Nobel Comity (Chemistry), 2013

Computational science!

Authors



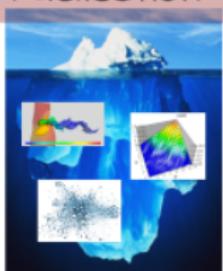
Data



Readers



Publication



Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...

– Nobel Comity (Chemistry), 2013

Aren't Computers Good for Science ?

How computers broke science – and what we can do to fix it.

- Point and click
- Spreadsheets : programming and data manipulation mistakes
 - Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase → MARCH1 → 2016-03-01 → 1456786800
 - 2310009E13 → 2.31E+19
- Complex software stacks : avoid proprietary software as much as possible
- Bugs : *Programming is difficult !*

All this is about Natural 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 Natural 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 ≠ Reality. Although designed and built by human beings, computer systems are **so complex** that mistakes easily slip in...

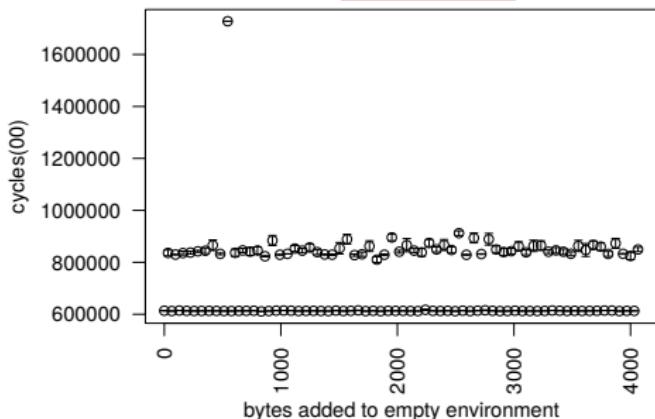
Experimenting with computers

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

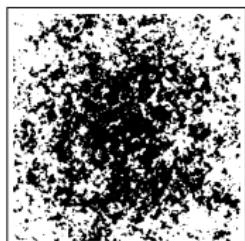
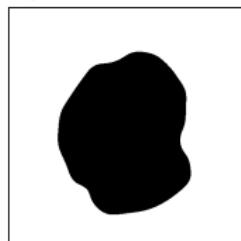
Computer performance ? Well, I design algorithms!

- "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





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

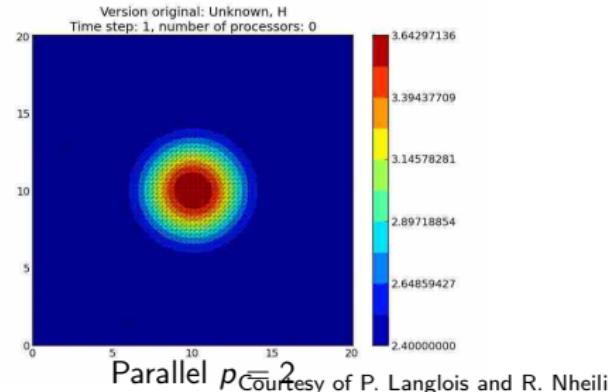
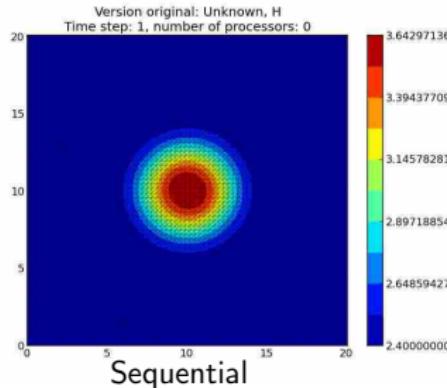
Did I mention we have parallel machines nowadays? 😊

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, ...)



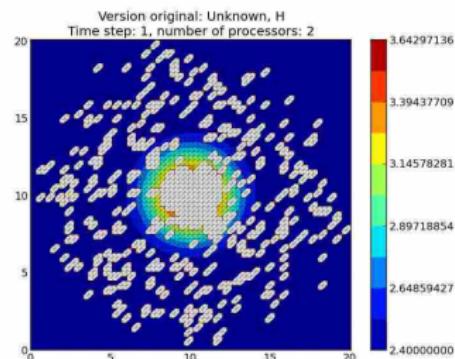
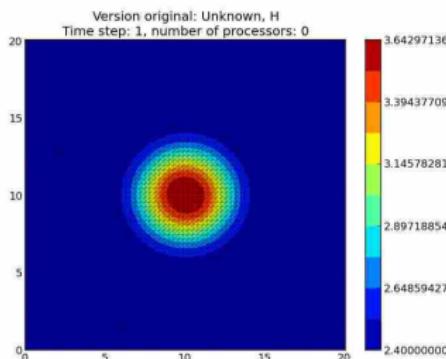
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 1



Courtesy of P. Langlois and R. Nheili

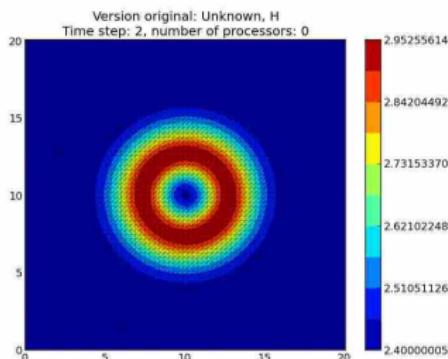
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

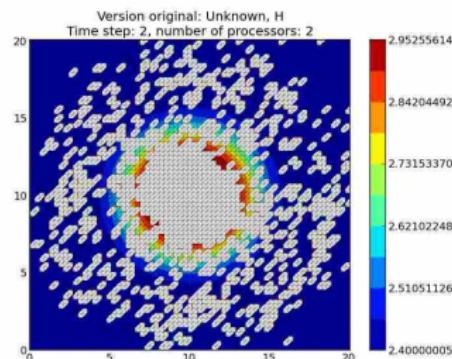
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 2



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

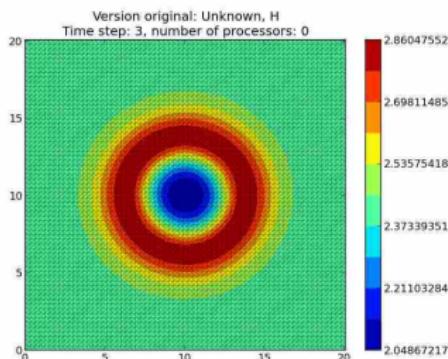
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

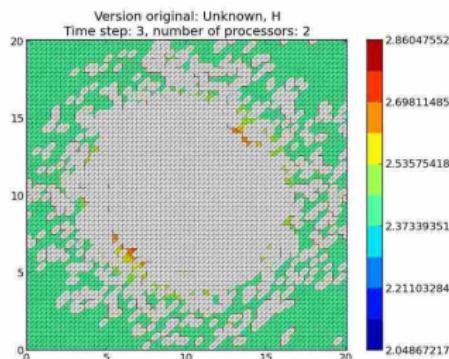
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 3



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

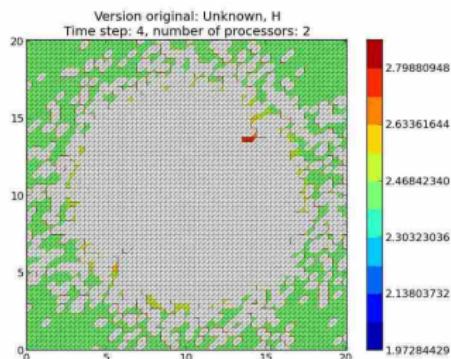
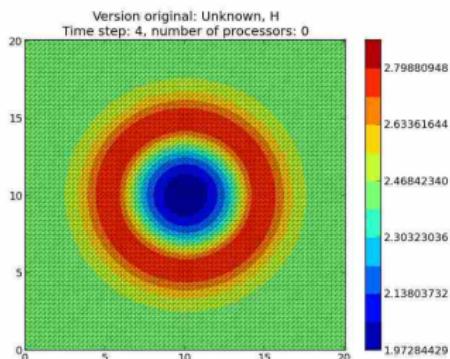
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 4



Courtesy of P. Langlois and R. Nheili

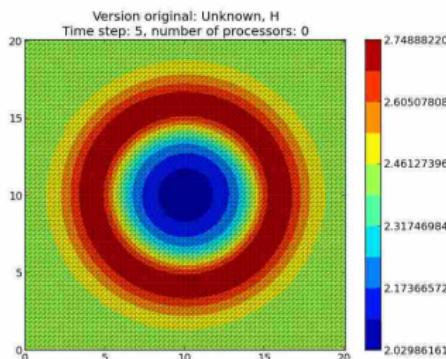
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

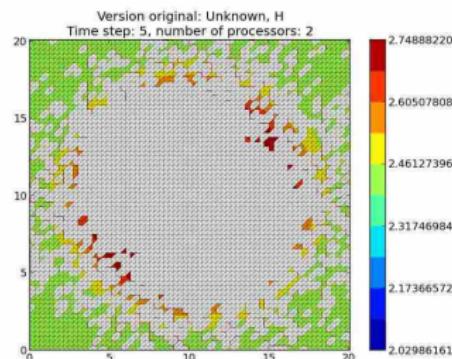
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 5



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

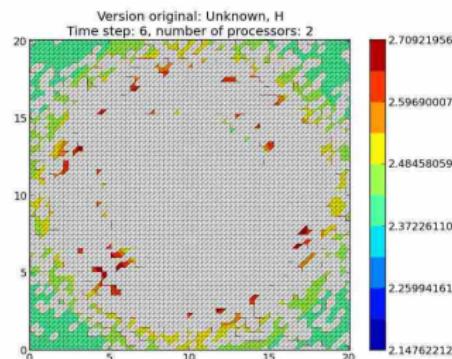
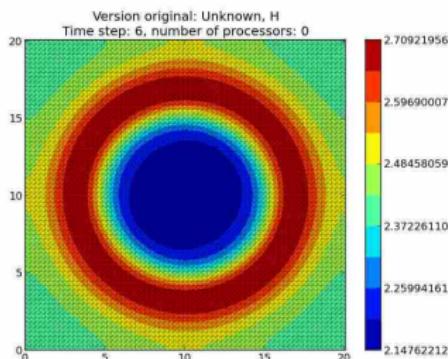
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 6



Courtesy of P. Langlois and R. Nheili

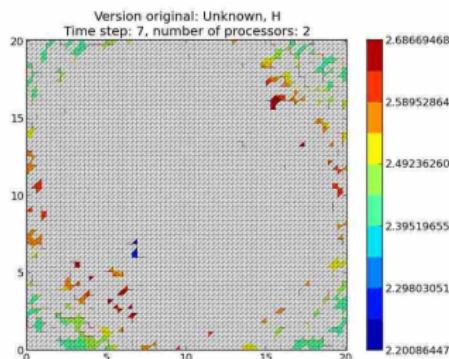
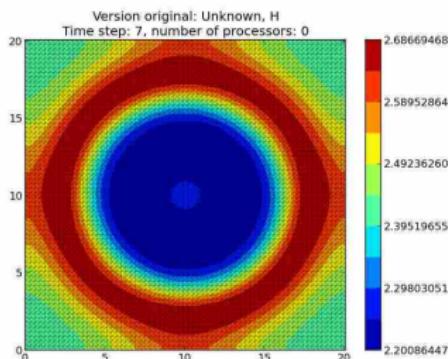
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 7



Courtesy of P. Langlois and R. Nheili

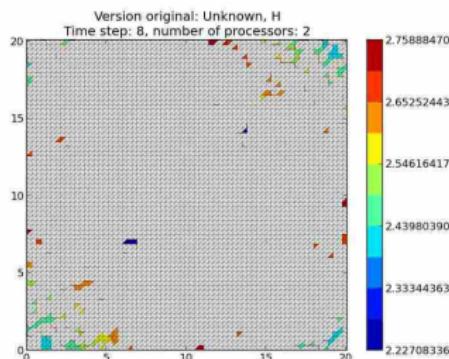
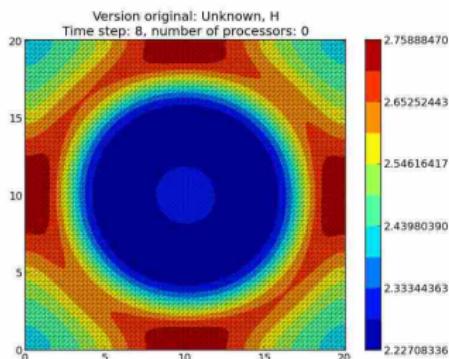
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 8



Courtesy of P. Langlois and R. Nheili

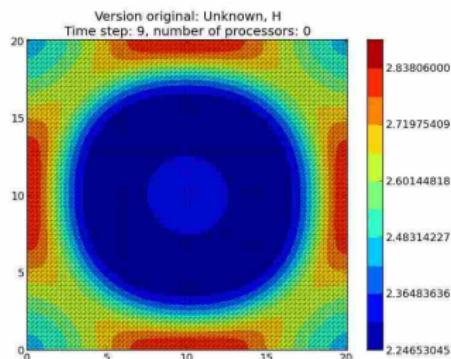
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

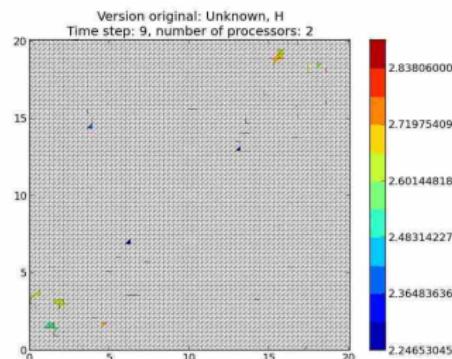
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 9



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

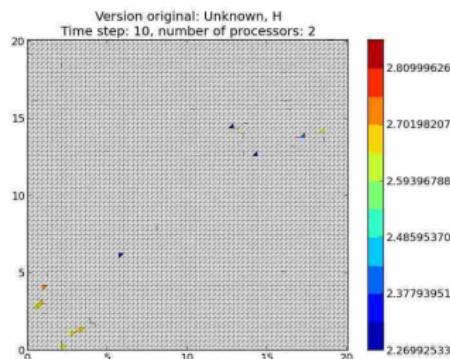
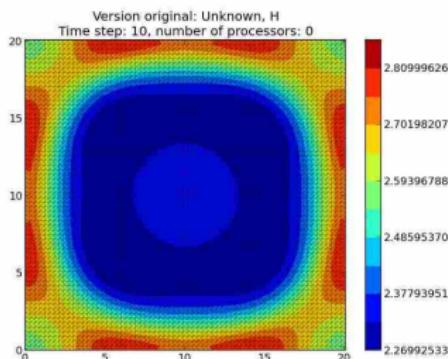
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 10



Courtesy of P. Langlois and R. Nheili

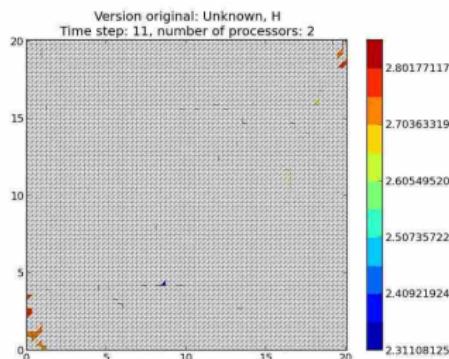
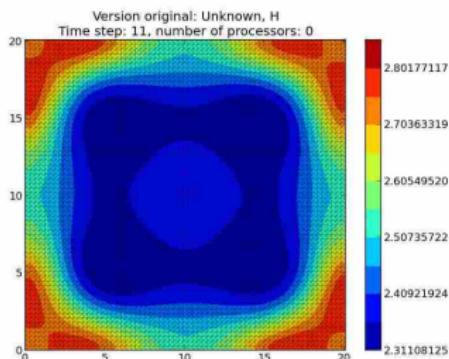
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 11



Courtesy of P. Langlois and R. Nheili

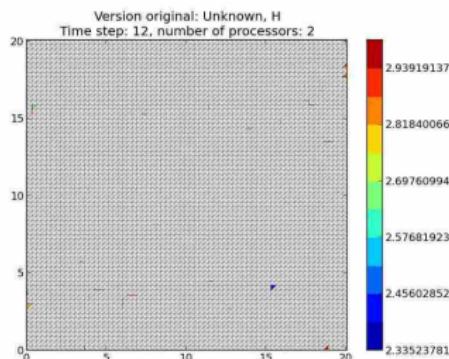
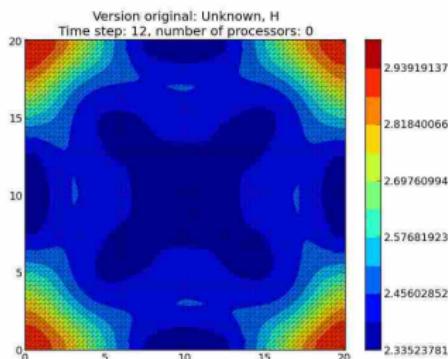
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 12



Courtesy of P. Langlois and R. Nheili

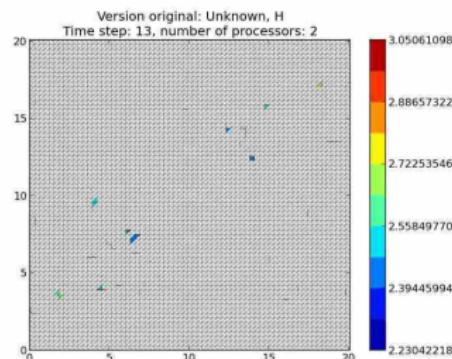
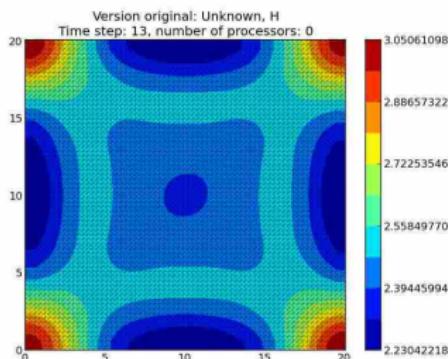
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 13



Courtesy of P. Langlois and R. Nheili

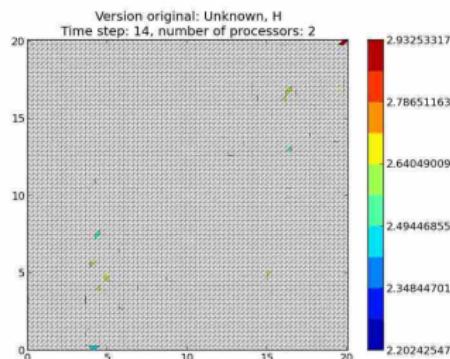
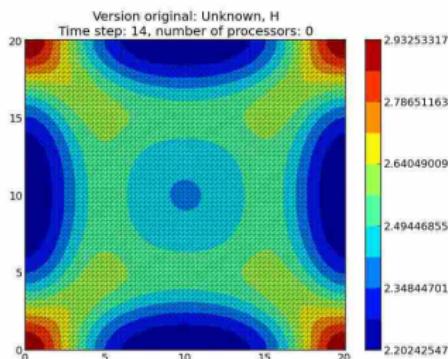
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 14



Courtesy of P. Langlois and R. Nheili

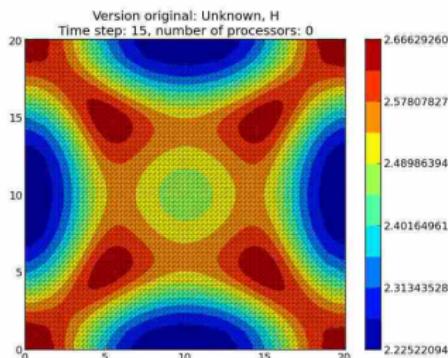
All I care about is the algorithm output

Did I mention we have parallel machines nowadays? 😊

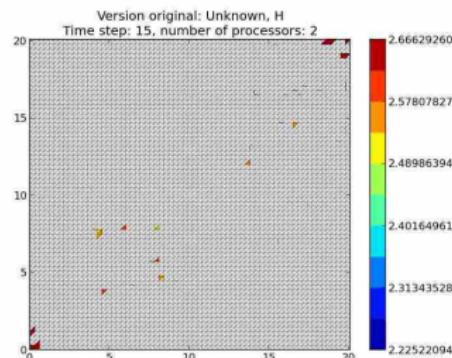
A white plot displays a non-reproducible value

NO numerical reproducibility!

time step = 15



Sequential



Parallel $p = 2$

Courtesy of P. Langlois and R. Nheili

All I care about is the algorithm output

Did I mention we have **parallel machines** nowadays? 😊

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

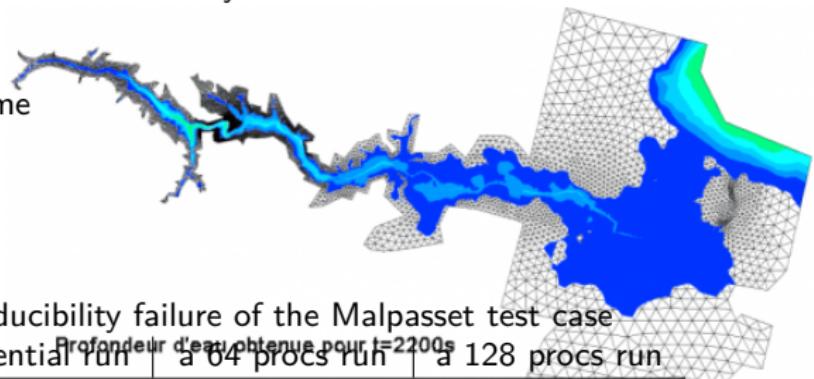


TABLE 1.1: Reproducibility failure of the Malpasset test case

	The sequential run	Profondeur d'eau obtenue pour t=2200s à 64 procs run	à 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.

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

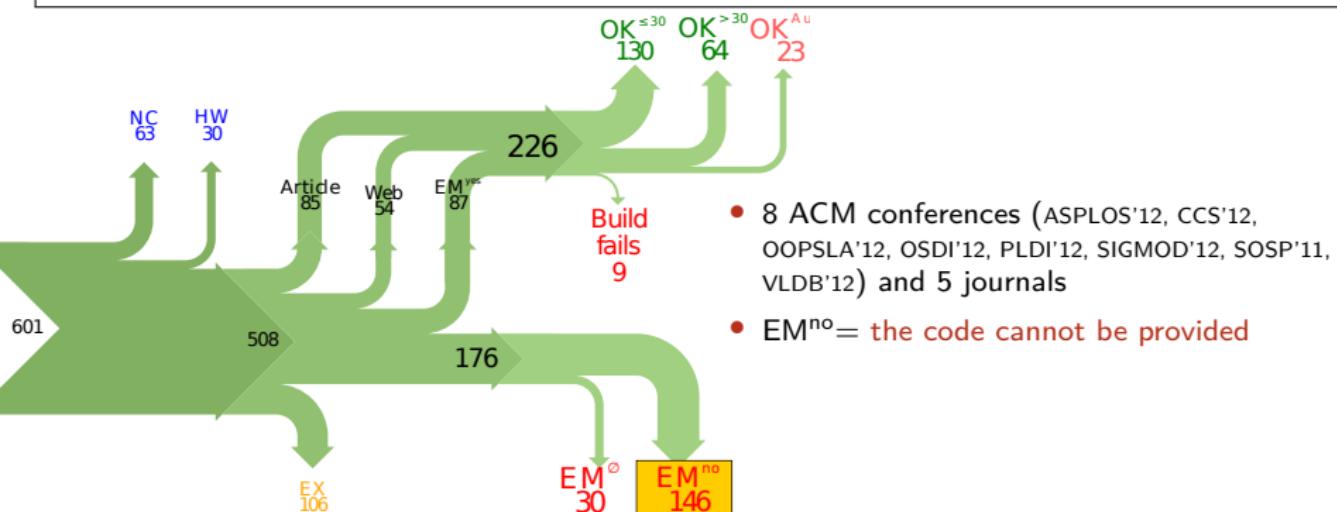
Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Lack of reproducibility is foremost a cultural matter

Collberg, Christian et Al., *Measuring Reproducibility in Computer Systems Research*,
<http://reproducibility.cs.arizona.edu/>

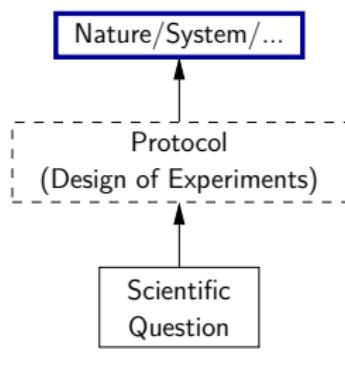


Reproducible Research: Trying to Bridge the Gap

Author



Published Article



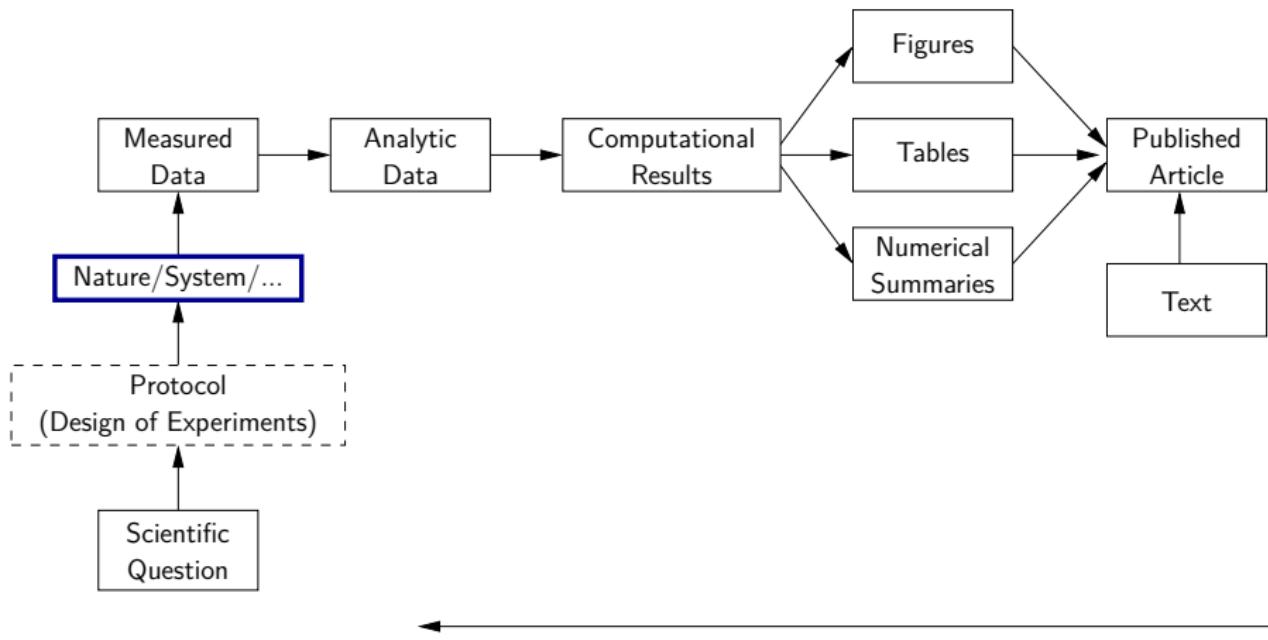
Reader

Inspired by Roger D. Peng's lecture on reproducible research, May 2014

In this series of lectures, we'll go from right to left and see how we can improve.

Reproducible Research: Trying to Bridge the Gap

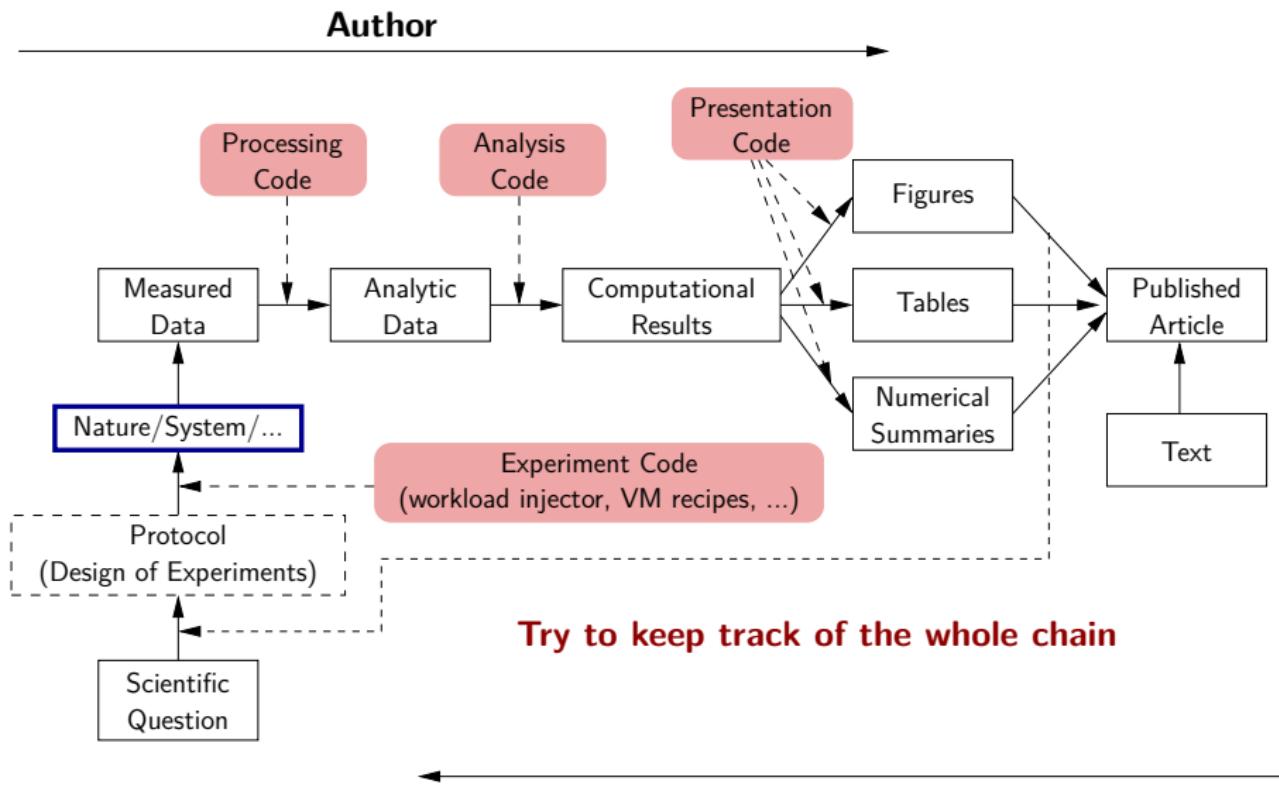
Author



Inspired by Roger D. Peng's lecture on reproducible research, May 2014

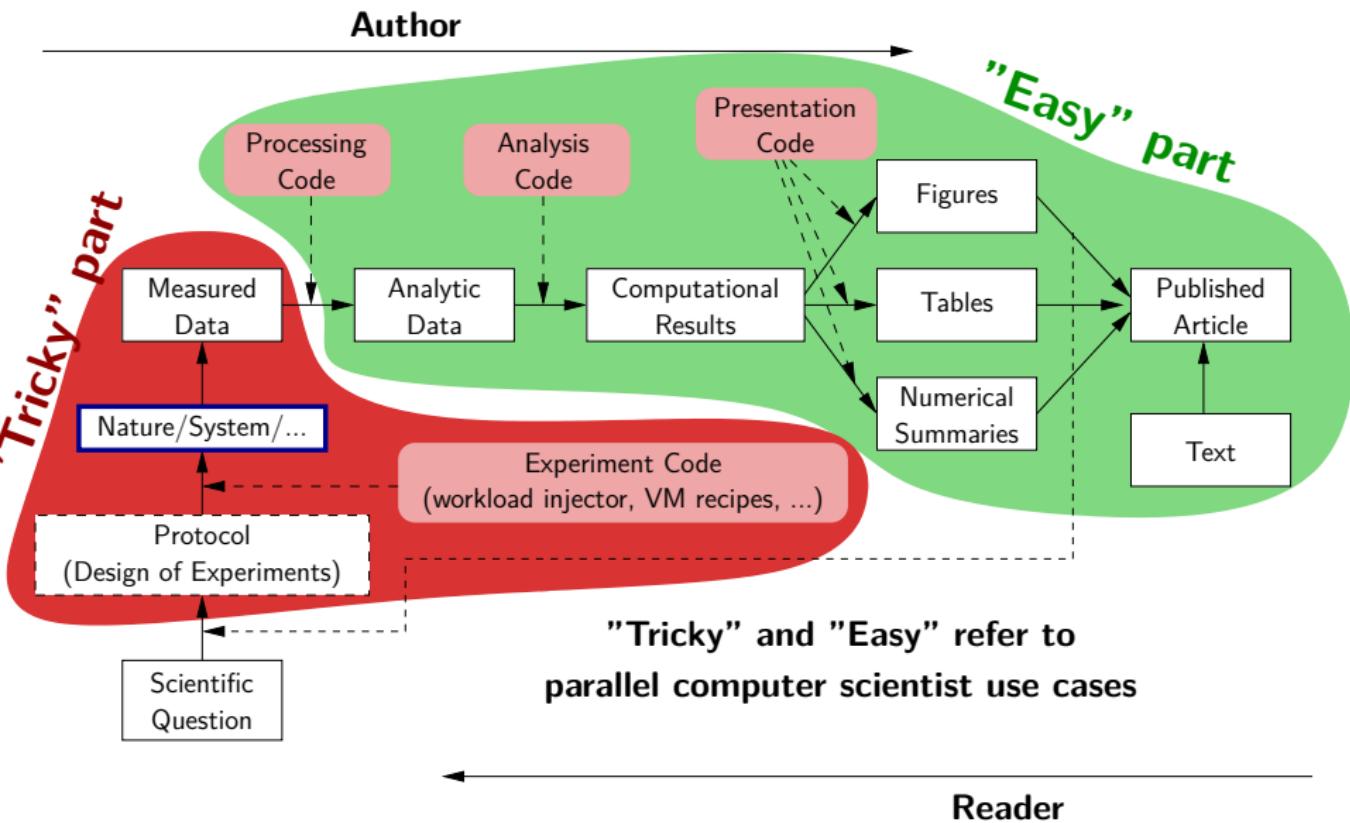
In this series of lectures, we'll go from right to left and see how we can improve.

Reproducible Research: Trying to Bridge the Gap



In this series of lectures, we'll go from right to left and see how we can improve.

Reproducible Research: Trying to Bridge the Gap



Inspired by Roger D. Peng's lecture on reproducible research, May 2014

In this series of lectures, we'll go from right to left and see how we can improve.

Paradigm Shift



in
c<>de
we
trust;

- ① Lack of information, data access
~~ *The right tracking/sharing tools*
- ② Lack of technical and scientific rigor
~~ *Better methodology (e.g., training in stats)*
- ③ Computation/programming mistakes
~~ *Continuous integration, code review, environment control, numerical analysis, ...*
- ④ Little incentive to care/share/...
~~ *The contribution evaluation processs (publication, recruiting, ...) needs to evolve*

Transparency increases the chances of finding mistakes
and getting rid of them

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Ipython/Jupyter Notebook

Web app: create and share documents that contain live code, equations, visualizations, and explanatory text

The screenshot shows two Jupyter Notebook windows side-by-side.

Left Window (Welcome View):

- Header: "jupyter Welcome to P"
- Toolbar: File, Edit, View, Insert, Cell
- Content area:
 - "jupyter Welcome to the..."
 - A yellow warning box: "WARNING Don't rely on this server..."
 - "Your server is hosted there..."
 - "Run some Python code..."
 - Instructions: "To run the code below:"
 - Click on the cell to select it
 - Press SHIFT+ENTER
 - A full tutorial link: "A full tutorial for using the Jupyter Notebook..."

Right Window (Exploring the Lorenz System):

- Header: "jupyter Lorenz Differential Equations (autosaved)"
- Toolbar: File, Edit, View, Insert, Cell, Kernel, Help
- Cell Toolbar: Python 3
- Section: "Exploring the Lorenz System"
- Text: "In this Notebook we explore the [Lorenz system](#) of differential equations:"
- Equation block:
$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$
- Text: "This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ , β , ρ) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963."
- Code cell (In [7]):

```
interact(Lorenz, N=fixed(10), angle=(0.,360.),
          sigma=(0.0,50.0), beta=(0.,5.), rho=(0.0,50.0));
```
- Interactive sliders for parameters:
 - angle: 308.2
 - max_time: 12
 - σ : 10
 - β : 2.6
 - ρ : 28
- Figure: A 2D plot showing the Lorenz attractor, a complex, fractal-like pattern formed by three interlocking trajectories.

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

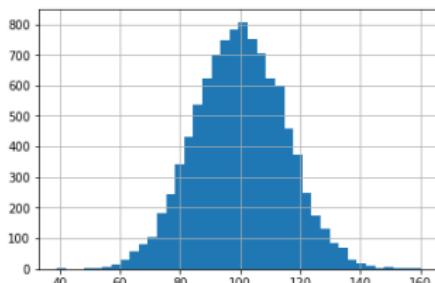
3.141592653589793

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... ☺).



Computational Document (Jupyter Notebook)

Document initial dans son environnement

jupyter example_pi (modified)

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

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

In [1]:

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

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

In [2]:

```
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=np.pi/2)
2/(sum((x*np.sin(theta))>1))/N
```

Out[2]: 3.1437198694098765

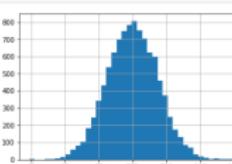
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

In [3]:

```
%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*

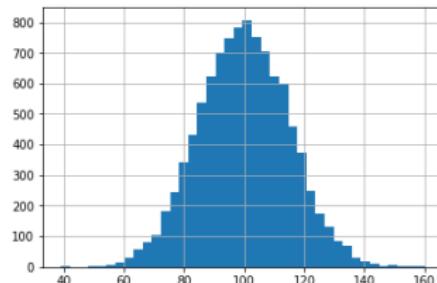
3.141592653589793

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=np.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... ☺).



Computational Document (Jupyter Notebook)

Document initial dans son environnement

jupyter example_pi (modified)

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

Un document computationnel

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

In [1]:

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

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

In [2]:

```
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=np.pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

Out[2]: 3.1437198694098765

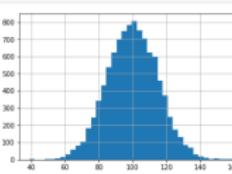
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

In [3]:

```
%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

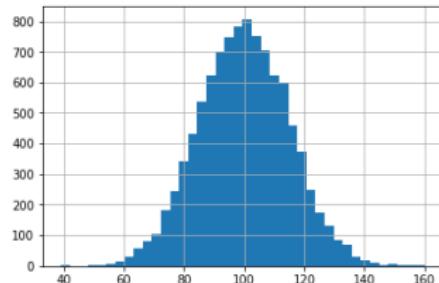
3.141592653589793

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=np.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... ☺).



Computational Document (Jupyter Notebook)

Document initial dans son environnement

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

```
In [1]:  
from math import *  
print(pi)  
3.141592653589793
```

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

```
In [2]:  
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
```

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... ☺).

```
In [3]:  
%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*

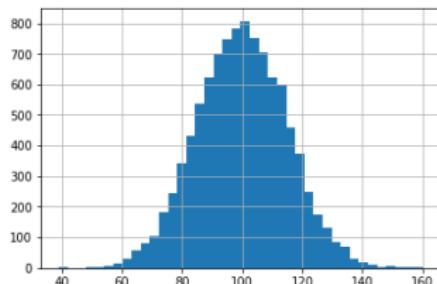
3.141592653589793

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... ☺).



Computational Document (Jupyter Notebook)

Document initial dans son environnement

jupyter example_pi.ipynb

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

In [1]:

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

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

In [2]:

```
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

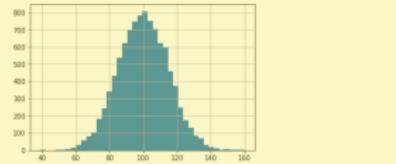
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

In [3]:

```
%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

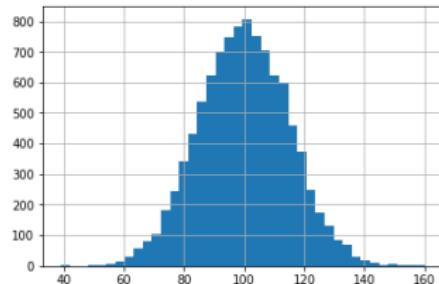
3.141592653589793

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... ☺).



Computational Document (Jupyter Notebook)

Document initial dans son environnement

jupyter example_pi.ipynb

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

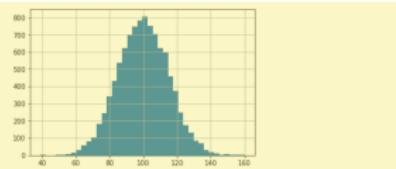
```
In [1]:  
from math import *  
print(pi)  
3.141592653589793
```

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

```
In [2]:  
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
```

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... ☺).

```
In [3]:  
%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

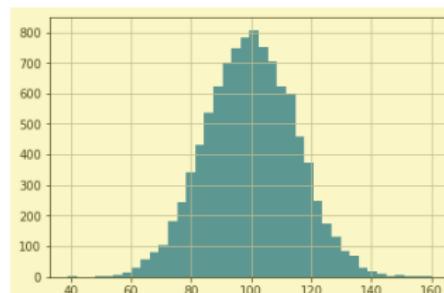
Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
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)
```

Export

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... ☺).



An *Provenance-Rich* Paper: ALPS2.0

The ALPS project release 2.0:
Open source software for strongly correlated
systems

B. Bauer¹ L. D. Carr² H.G. Evertz¹ A. Feiguin⁴ J. Freire⁵
S. Fuchs⁶ L. Gamper¹ J. Gukelberger¹ E. Gull⁷ S. Guertler⁸
A. Hehn¹ R. Igarashi^{9,10} S.V. Isakov¹ D. Koop¹ P.N. Ma¹
P. Mateo¹¹ H. Matsuo¹¹ O. Parcollet¹² G. Pawłowski¹³
J.D. Picon¹⁴ L. Pollet^{1,15} E. Santos² V.W. Scarola¹⁶
U. Schollwöck¹⁷ C. Silva¹ S. Surer¹ S. Todo^{10,11} S. Trebst¹⁸
M. Troyer¹ M. L. Wall¹⁹ P. Werner¹ S. Wessel^{19,20}

¹Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland

²Department of Physics, Colorado School of Mines, Golden, CO 80401, USA

³Institut für Theoretische Physik, Technische Universität Graz, A-8010 Graz, Austria

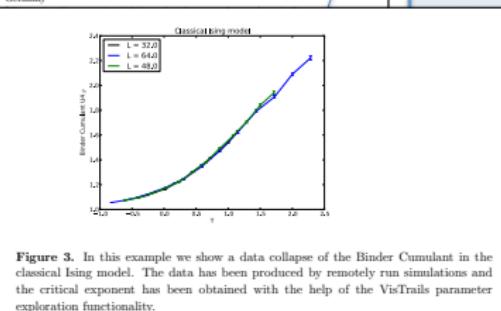
⁴Department of Physics and Astronomy, University of Wyoming, Laramie, Wyoming 82071, USA

⁵Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, Utah 84112, USA

⁶Institut für Theoretische Physik, Georg-August-Universität Göttingen, Göttingen, Germany

⁷Columbia University, New York, NY 10027, USA

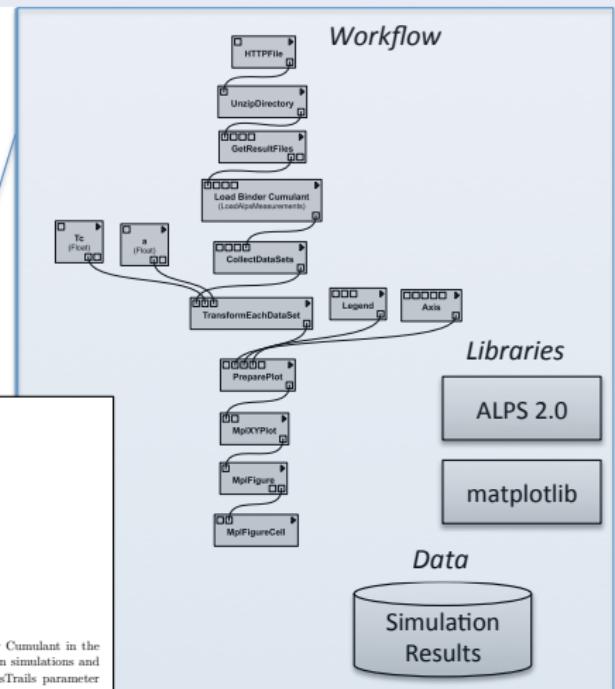
⁸Bethe Center for Theoretical Physics, Universität Bonn, Nussallee 12, 53115 Bonn, Germany



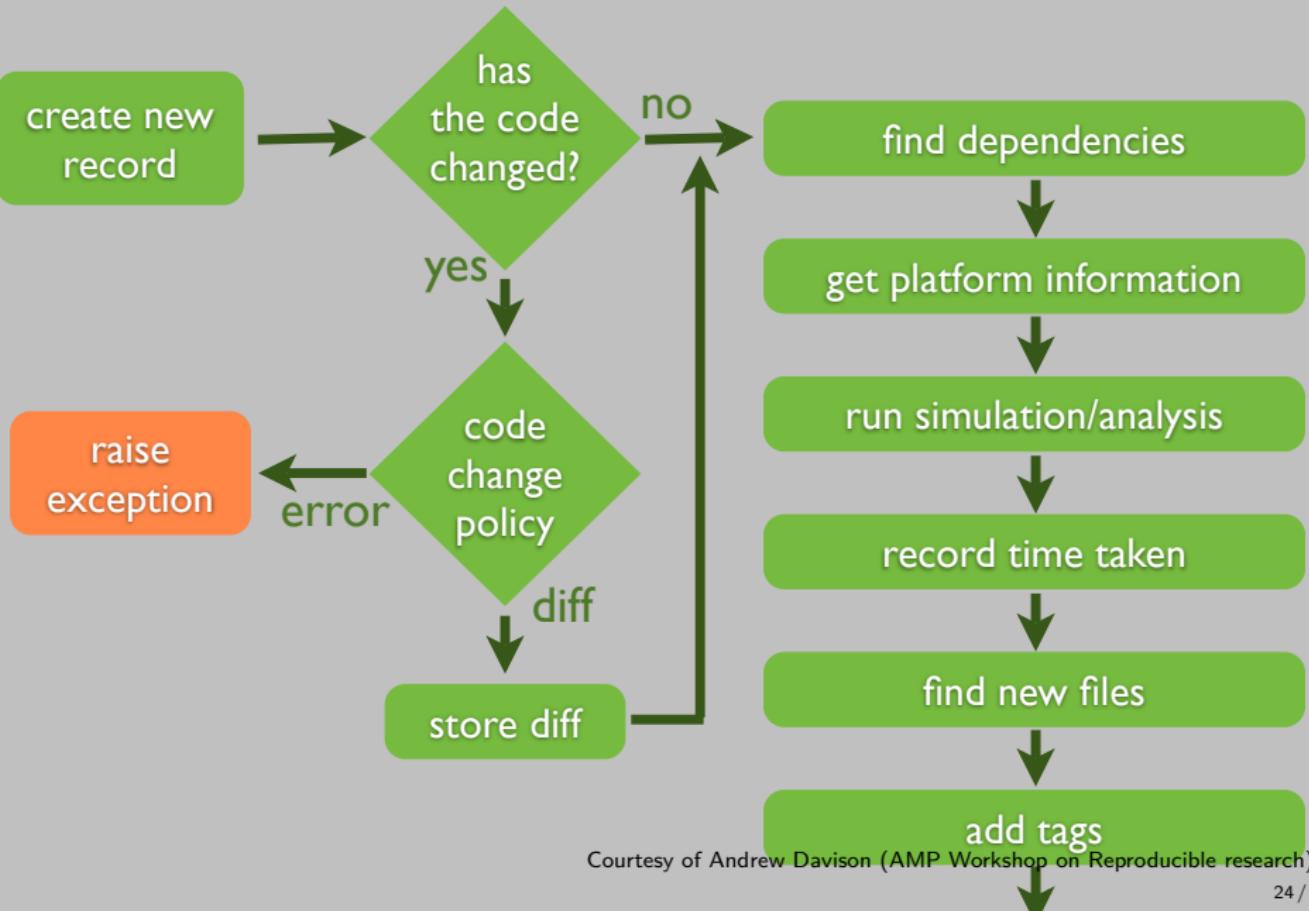
† Corresponding author

Figure 3. In this example we show a data collapse of the Binder Cumulant in the classical Ising model. The data has been produced by remotely run simulations and the critical exponent has been obtained with the help of the VisTrails parameter exploration functionality.

Courtesy of Juliana Freire (AMP Workshop on Reproducible research)



Sumatra: an "experiment engine" that helps taking notes



Courtesy of Andrew Davison (AMP Workshop on Reproducible research)

Sumatra: an "experiment engine" that helps taking notes

```
$ smt comment 20110713-174949 "Eureka! Nobel prize  
here we come."
```

Sumatra: an "experiment engine" that helps taking notes

```
$ smt tag "Figure 6"
```

Sumatra: an "experiment engine" that helps taking notes

The screenshot shows a web-based application window titled "Sumatra: TestProject: List of records". The browser address bar displays "http://127.0.0.1:8002/". The main content area is a table with the following columns:

Delete Include data	Label	Reason	Outcome	Duration	Processes	Simulator		Script			Date	Time	Tags
						Name	Version	Repository	Main file	Version			
<input type="checkbox"/>	20100709-154255		'Eureka! Nobel prize here we come.'	0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:42:55	
<input type="checkbox"/>	20100709-154309			0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:09	
<input type="checkbox"/>	haggling	'determine whether the gourd is worth 3 or 4 shekels'	'apparently, it is worth NaN shekels.'	0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:20	foobar
<input type="checkbox"/>	20100709-154338	'test effect of a smaller time constant'		0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:38	
<input type="checkbox"/>	haggling_repeat	Repeat experiment haggling	The new record exactly matches the original.	0.58 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:47	

My daily solution: Emacs/Org-mode

My journal a single file

- Chronological organization with TAGS
- Heterogenous information
- Code execution is convenient

My students' laboratory notebooks

- Conventions
- ≈ Chronological organization
- Track info about experiments (who, what, when, how, **why**) and data analysis

Replicable articles

- Easily built on the previous documents
- Many hidden sections (data download, perl/R scripts)
- Sources connect figures with data and meta-data

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Complex Environments

```
1 import matplotlib  
2 print(matplotlib.__version__)
```

3.10.1+dfsg1

Complex Environments

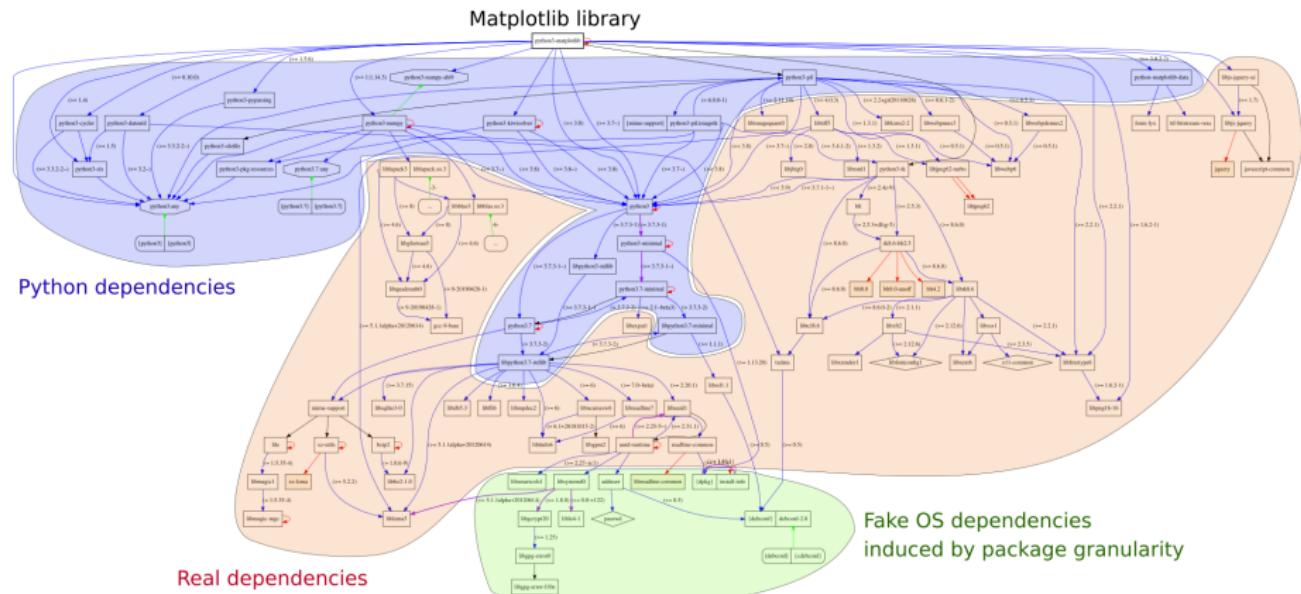
```
1 import matplotlib  
2 print(matplotlib.__version__)
```

3.10.1+dfsg1

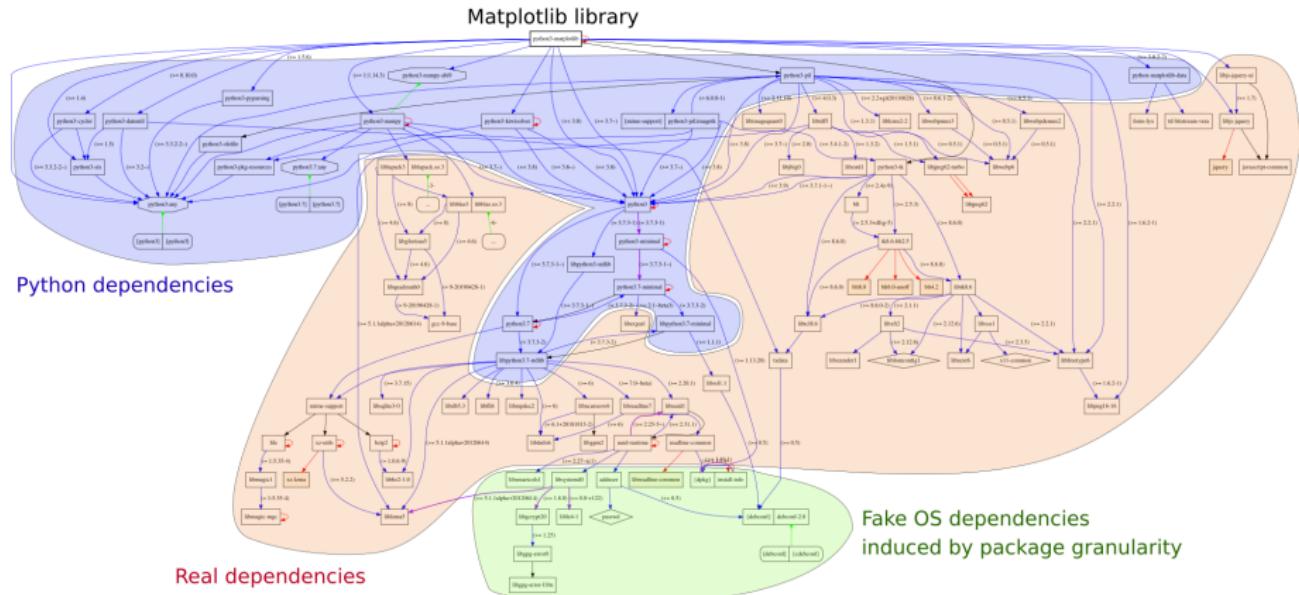
Version: 2.1.1-2

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)

Complex Environments



Complex Environments



Virtual images or Docker containers allow to freeze environments

Two approaches: *preserve the mess* vs. *encourage cleanliness*

Reprozip

Automagically pack your experiment to fight dependency hell

ON THE ORIGINAL MACHINE

```
$ pip install reprozip
$ reprozip trace ./myexperiment -my --options inputs/somefile.csv other_file_here.bin
experiment: 0%... 25%... 50%... 75%... 100%
result: 42.137
Configuration file written in .reprozip/config.yml
Edit that file then run the packer -- use 'reprozip pack -h' for help
$ reprozip pack my_experiment.rpz
[REPROZIP] 17:26:42.588 INFO: Creating pack my_experiment.rpz...
[REPROZIP] 17:26:42.589 INFO: Adding files from package coreutils...
[REPROZIP] 17:26:42.601 INFO: Adding files from package libc6...
[REPROZIP] 17:26:42.906 INFO: Adding other files...
[REPROZIP] 17:26:43.450 INFO: Adding metadata...
```

ON ANOTHER MACHINE

```
$ pip install reprounzip[all]
$ reprounzip vagrant setup my_experiment.rpz mydirectory
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'remram/debian-7-amd64'...
==> default: Booting VM...
==> default: Machine booted and ready!
==> default: Running provisioner: shell...
$ reprounzip vagrant run mydirectory
experiment: 0%... 25%... 50%... 75%... 100%
result: 42.137
$ reprounzip vagrant upload /tmp/new_config:global-config
$ reprounzip vagrant run mydirectory --cmdline ./myexperiment --other --options
inputs/somefile.csv
experiment: 0%... 25%... 50%... 75%... 100%
result: -17.814
```

Other options:

CARE, CDE, Reprozip, Pipenv

Docker, Singularity

Spack, GUIX, NIX

Controlling experiments

- Naive way: sh + ssh + ...
- Better way: use a workflow management system (taverna, galaxy, kepler, vistrails, ...)
- Parallel/distributed experiments require specific experiment engines
 - ▶ **Expo** (2007-, G5K)
 - ▶ **XPflow** (2012-, G5K)
 - ▶ **Execo** (2013-, G5K)

} although nothing specific to G5K

 - ▶ Plush (2006-, Planetlab)
 - ▶ OMF (2009-, Wireless)
 - ▶ Splay (2008), ...

They differ in the underlying paradigms and the platforms for which they have been designed

A survey of general-purpose experiment management tools for distributed systems, T. Buchert, C. Ruiz, L. Nussbaum, O. Richard, FGCS, 2014

- Control your **numerical results** (random generators, libraries, rounding and non-determinism, ...)

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controloing your environment/experiments

Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Hosting platforms

- Your webpage 😞
- Articles: Arxiv, HAL
- Code: Github or Institutional Gitlab
- Figshare, Zenodo 😊
- Companion websites ([elsevier executable paper](#) 😞, [runmycode exec&share](#), [code ocean](#) ...)

This may seem easy but is quite tricky:

- Arbitrary limits can make your life painful
- Perennity ([Roberto Di Cosmo's talks](#))
 - CodeSpaces murdered on Amazon, Google Code termination, Gitorious shutdown, ...
 - Disruption of the web of reference: URLs decay (half-life of 4 years), DOIs have little guarantee, ...



Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controloing your environment/experiments

Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

Publishing initiatives

Artifact evaluation and ACM badges Article appendix (SC)



Platforms Elsevier executable paper, ACM pushes for code ocean

Double blind vs. Open reviews plus "legal aspects"

RR-oriented & Open Journals IPOL



Funding Agencies

US *NSF has released a new announcement encouraging submission of research projects relating to **reproducibility and replicability**.*

– **Jim Kurose, Oct. 2016**

*The NIH is committed to promoting **rigorous and transparent research** in all areas of science supported by a variety of grant programs.*

– **NIH: Rigor and Reproducibility guidelines**

Europe *In 2012, the European Commission encouraged all EU Member States to **put publicly-funded research results in the public sphere** in order to make science better and strengthen their knowledge-based economy.*

– **Open Science (Open Access)**

*A key element will be capacity building to link literature and data in order to enable a more transparent evaluation of research and **reproducibility** of results.*

H2020-EINFRA-2014-2015

–

Encouraging the re-use of research data generated by publicly funded research projects

– **Science with and for Society, 27 October 2017**

Outline

① Science Crisis ?

② Is Computer Science Really Concerned With This?

③ Reproducible Research/Open Science in a Nutshell

④ Illustrating Nice Ideas Through Different Tools

Taking notes and explaining

Controlling your environment/experiments

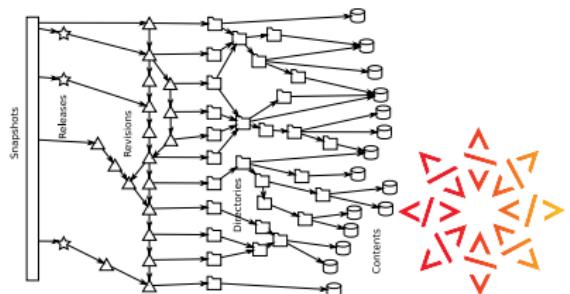
Making data/code available

Changing evaluation practices

⑤ What can Computer Scientists do ?

On the "technical" side (1/2)

- Better documenting what we do: **Laboratory notebooks**
 - Literate programming is great for analysis, and reproducible articles but does not go well yet with conducting experiments and workflows
 - A real adoption of such practice requires more storage and the ability to navigate in such information
- Better software engineering practice: Public releases, **devops approach**, reproducible builds, numerical aspects
 - Moving/evolving technology. Preservation ? Adoption ?
 - Should not slow down research
- Fighting against software/data degradation: **Software Heritage**, zenodo
 - Challenges: multiple! curation, access/privacy, exploitation, navigation, storage, ...



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

On the "technical" side (2/2)

- Better experimental practice and platforms: FIT IoT-lab, G5K are world leading experimental infrastructures; rely on standard simulators (**SimGrid, NS3**)
 - Maintenance cost, keeping in pace with technology, practices for prototype platforms, control, sharing of experimental conditions with others, experimental engines



- Need for convergence (SILEX) in term of software infrastructure and practice (e.g., security, account management, access, isolation, experiment management, etc.) ?

On the "social" side

Slight **cultural changes** in our **relation to publication** and **daily practice**

- Changing our social model to favor adoption of better practice
 - Artifact evaluation, open reviews, ... (e.g., IPOL, ReScience)
 - Alternative evaluation models (publishing, funding, ...)
- Learning is the essence of our work. ~> Train our researchers and **students**
 - MOOC on Reproducible Research in Oct. 2018 on FUN
 - Train in statistics, experimental practice, design of experiments

It's up to us. We should care and take the lead

Main benefit:

- Higher confidence in our work ~> definite competitive advantage
- Our research becomes **sound, deeper, auditable, more visible, reusable**