

# NUMERICAL/SOFTWARE CHAOS AND REPRODUCIBILITY ISSUES IN HPC

---

Arnaud Legrand



Theory Days on Computational tools in physics:  
*performances vs. reproducibility*

Toulouse/Online, March 2024



# REPRODUCIBILITY CRISIS

## Is there a reproducibility crisis A Nature survey, 2016

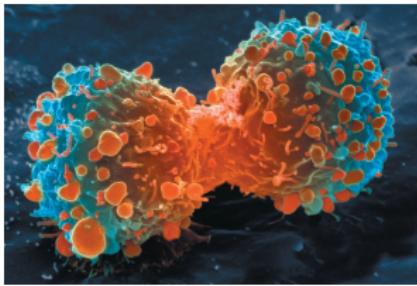
### COMMENT

JOHN MELICK Shifts expertise to track mutations where they emerge ▶ 20

EARTH SYSTEM Past climates give valuable clues to future warming ▶ 37

HISTORY OF SCIENCE Don't let us forget better track using Google ▶ 40

WYLIE VANCE and ANTHONY STODDART Wyllie Vance and Anthony Stoddart harness ▶ 41



Many landmark findings in preclinical oncology research are not reproducible, in part because of inadequate cell lines and animal models.

## Raise standards for preclinical cancer research

C. Glenn Begley and Lee M. Ellis propose how methods, publications and incentives must change if patients are to benefit.

Efforts over the past decade to characterize the genetic alterations in human cancers have led to a better understanding of what drives them. Although this is a major achievement, it has also led to a cancer field that hoped that this would lead to more effective drugs. Historically, our ability to translate cancer research from the lab to the clinic has been miserably low. And, clinical

trials in oncology have the highest failure rate compared with other therapeutic areas. Given the high unmet need in oncology, it is understandable that the hurried to clinical development may have lowered the bar. Certainly, the limitations of preclinical tools with suboptimal predictive validation will contribute to the failure of many trials. The cost rate is not sustainable or acceptable, and

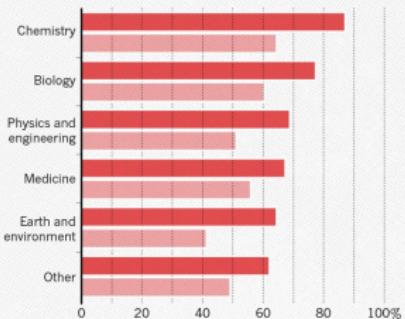
investigation must focus on those that approach to translating discovery research into greater clinical success and impact.

Many factors are responsible for the high failure rate in oncology, including the inherently difficult nature of this disease. Certainly, the limitations of preclinical tools with suboptimal predictive validation will contribute to the failure of many trials. The cost rate is not sustainable or acceptable, and

## HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.

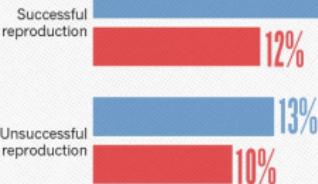
- Someone else's
- My own



## HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

Although only a small proportion of respondents tried to 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

## Must try harder

Too many sloppy mistakes are creeping into scientific papers at the data — and at themselves.

## Six red flags for suspect work

C. Glenn Begley explains how to recognize the preclinical papers in which the data won't stand up

## Error prone

Biologists must realize the pitfalls of massive amount of data

## Know when your numbers are significant

- Nekrutenko & Taylor, *Nature Genetics* (2012)
- Alsheikh-Ali et al. *PLoS ONE* (2011)
- Begley & Ellis *Nature* (2012)

# REPRODUCIBLE RESEARCH: AN UMBRELLA TERM

Connexions with:

- Open Science/Access/Data/Source/Reviews/Laboratory notebooks/...

Many different terms and issues depending on the domain

- Reproduce, Replicate, Repeat, Rerun, Redo, Reuse, Register, Report

# REPRODUCIBLE RESEARCH: AN UMBRELLA TERM

Connexions with:

- Open Science/Access/Data/Source/Reviews/Laboratory notebooks/...

Many different **terms** and **issues** depending on the **domain**

- Reproduce, Replicate, Repeat, Rerun, Redo, Reuse, Register, Report

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

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

# REPRODUCIBLE RESEARCH: AN UMBRELLA TERM

Connexions with:

- Open Science/Access/Data/Source/Reviews/Laboratory notebooks/...

Many different **terms** and **issues** depending on the **domain**

- Reproduce, Replicate, Repeat, Rerun, Redo, Reuse, Register, Report

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

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

**Genomics** software engineering, comput. reproducibility, provenance

**Computational fluid dynamics** numerical chaos, parallel architectures

**Artificial Intelligence** most of the above 😊

# REPRODUCIBLE RESEARCH: AN UMBRELLA TERM

Connexions with:

- Open Science/Access/Data/Source/Reviews/Laboratory notebooks/...

Many different **terms** and **issues** depending on the **domain**

- Reproduce, Replicate, Repeat, Rerun, Redo, Reuse, Register, Report

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

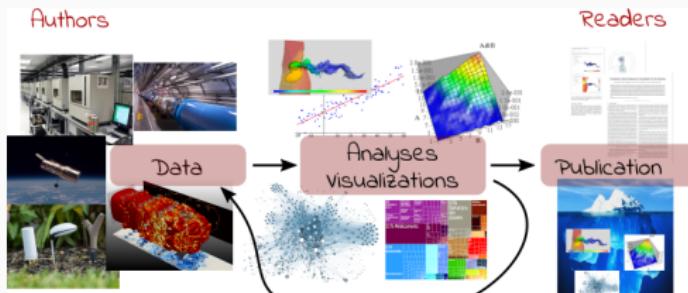
**Psychology, Nutrition** HARKING, p-hacking  $\rightsquigarrow$  pre-registration

**Genomics** software engineering, comput. reproducibility, provenance

**Computational fluid dynamics** numerical chaos, parallel architectures

**Artificial Intelligence** most of the above 😊

AFAIC, I care about **transparency**



# NO TRANSPARENCY NO CONSENSUS



## (HP)COMPUTATIONAL REPRODUCIBILITY ISSUES

---

# CLASSICAL COMPUTATIONAL SCIENCE CHALLENGES

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

## Tracking parameters and data

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

# CLASSICAL COMPUTATIONAL SCIENCE CHALLENGES

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

## Tracking parameters and data

- **\*.dat**? Ooh, so did you run 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? When did I do this?

# CLASSICAL COMPUTATIONAL SCIENCE CHALLENGES

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

## Tracking parameters and data

- **\*.dat**? Ooh, so did you run 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? When did I do this?

## Tracking code version

- **my\_code** is revision **21b95ecfa0911d6ca87668482b11ab9498edd8f3**

# CLASSICAL COMPUTATIONAL SCIENCE CHALLENGES

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

## Tracking parameters and data

- \*.dat? Ooh, so did you run 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? When did I do this?

## 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 ...`

# CLASSICAL COMPUTATIONAL SCIENCE CHALLENGES

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

## Tracking parameters and data

- \*.dat? Ooh, so did you run 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? When did I do this?

## 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 ...`

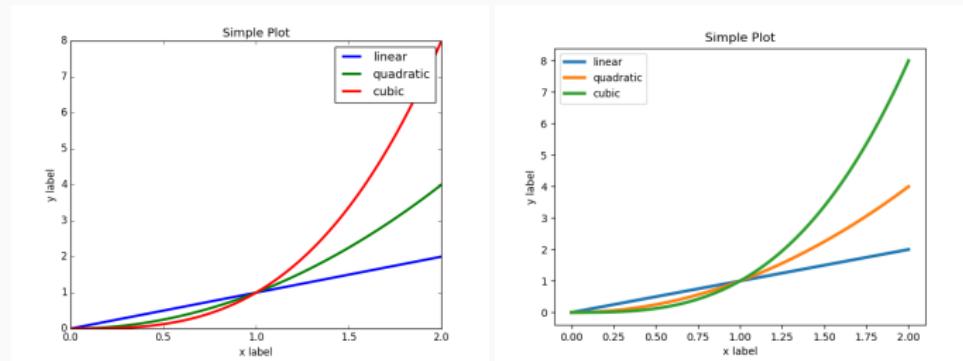
## Handle complex sequences and reuse results

leverage cloud/supercomputers and their ecosystem (workflow, RJMS, data warehouse)

## SOFTWARE DEPENDENCIES: HORROR STORIES

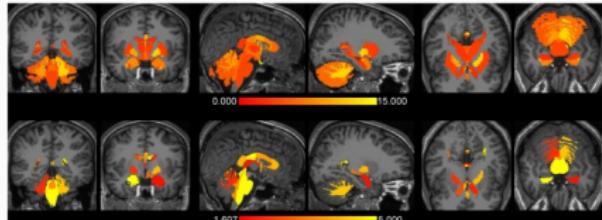
# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution



# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity



The Effects of FreeSurfer Version, Workstation Type, and Macintosh Operating System Version on Anatomical Volume and Cortical Thickness Measurements (PLOS ONE, 2012)

*Significant differences in volume and cortical thickness were revealed across FreeSurfer versions:*

- volume:  $8.8 \pm 6.6\%$  (range 1.3-**64.0%**)
- cortical thickness:  $2.8 \pm 1.3\%$  (range 1.1-7.7%)

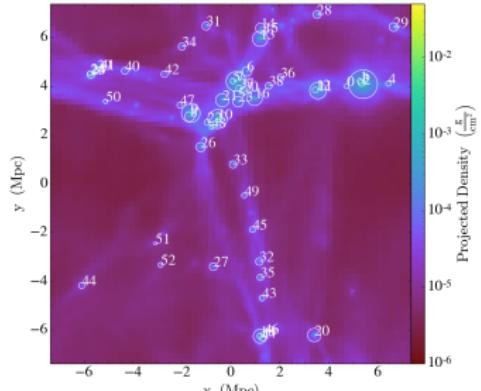
*About a factor two smaller differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

*In the context of an ongoing study, users are discouraged to update to a new major release of either FreeSurfer or operating system.*

*Formal assessment of the accuracy of FreeSurfer is desirable.*

# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity
- Impact of the compiler

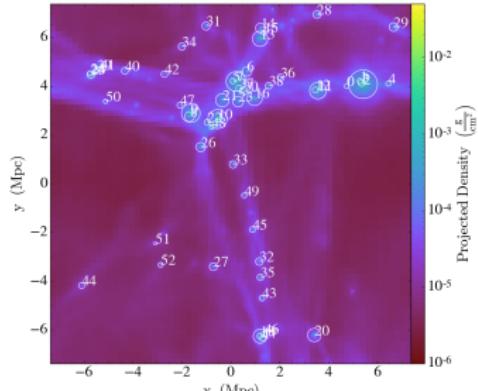


Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler  | Optim. | Largest Halo<br>Avg Mass. | Std. Err  | Walltime |
|-----------|--------|---------------------------|-----------|----------|
| gcc@6.2.0 | None   | 2.273E 46                 | 1.069E 44 | 22h      |

# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity
- Impact of the compiler
  - Not even mentioning the lack of determinism stemming from threads, MPI collective communication and non-associative operations

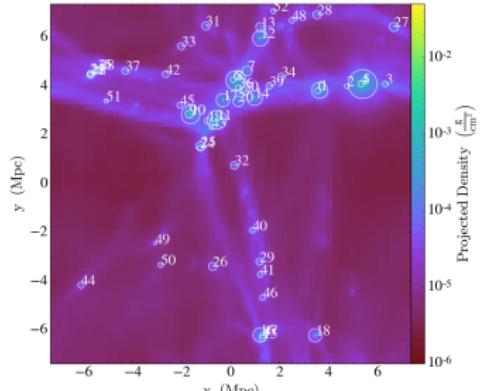


Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler  | Optim. | Largest Halo<br>Avg Mass. | Std. Err  | Walltime |
|-----------|--------|---------------------------|-----------|----------|
| gcc@6.2.0 | None   | 2.273E 46                 | 1.069E 44 | 22h      |

# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity
- Impact of the compiler
  - Not even mentioning the lack of determinism stemming from threads, MPI collective communication and non-associative operations

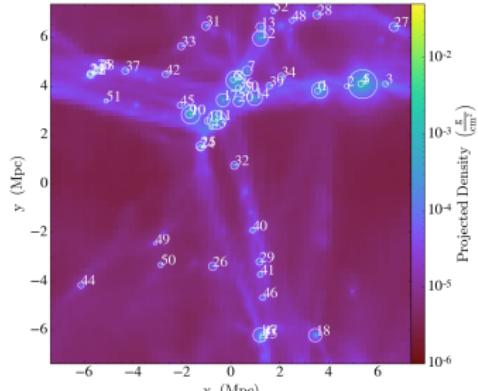


Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler  | Optim. | Largest Halo<br>Avg Mass. | Std. Err  | Walltime |
|-----------|--------|---------------------------|-----------|----------|
| gcc@6.2.0 | None   | 2.273E 46                 | 1.069E 44 | 22h      |

# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity
- Impact of the compiler
  - Not even mentioning the lack of determinism stemming from threads, MPI collective communication and non-associative operations

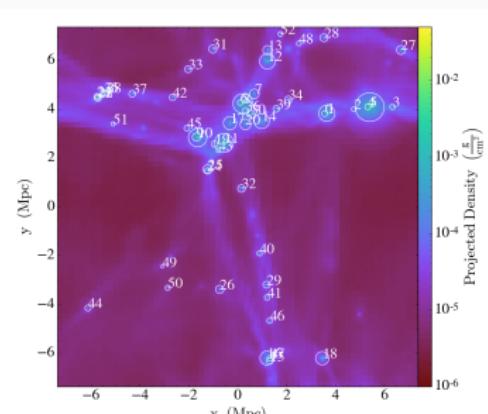


Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler  | Optim. | Largest Halo<br>Avg Mass. | Std. Err  | Walltime |
|-----------|--------|---------------------------|-----------|----------|
| gcc@6.2.0 | None   | 2.273E 46                 | 1.069E 44 | 22h      |
| gcc@6.2.0 | Normal | 2.266E 46                 | 1.218E 44 | 10h      |
| gcc@6.2.0 | High   | 2.275E 46                 | 1.199E 44 | 9h       |

# SOFTWARE DEPENDENCIES: HORROR STORIES

- Software environment evolution
- OS heterogeneity
- Impact of the compiler
  - Not even mentioning the lack of determinism stemming from threads, MPI collective communication and non-associative operations



Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler     | Optim. | Largest Halo |           | Walltime |
|--------------|--------|--------------|-----------|----------|
|              |        | Avg Mass.    | Std. Err  |          |
| gcc@6.2.0    | None   | 2.273E 46    | 1.069E 44 | 22h      |
| gcc@6.2.0    | Normal | 2.266E 46    | 1.218E 44 | 10h      |
| gcc@6.2.0    | High   | 2.275E 46    | 1.199E 44 | 9h       |
| intel@16.0.3 | None   | 22.71 E 46   | 1.587E 44 | 39h      |
| intel@16.0.3 | Normal | 43.30 E 46   | 1.248E 44 | 7h       |
| intel@16.0.3 | High   | 2.268E 46    | 1.414E 44 | 6h       |
| cce@8.5.5    | Low    | 43.11 E 46   | 1.353E 44 | 16h      |
| cce@8.5.5    | Normal | 2.271E 46    | 1.261E 44 | 6h       |
| cce@8.5.5    | High   | 2.272E 46    | 1.341E 44 | 5h       |

# 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) ≠ 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) ≠ 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                                     | [-0.827396059946821368141165...] |
| (multiple precision interval arithmetic) | -0.827396059946821368141165...]  |

Courtesy of Christophe Denis

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

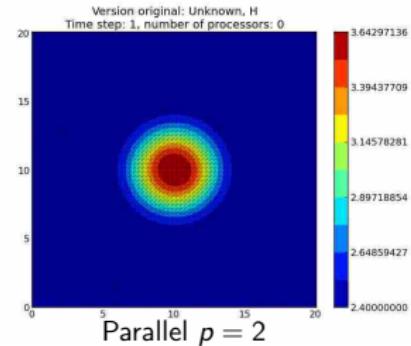
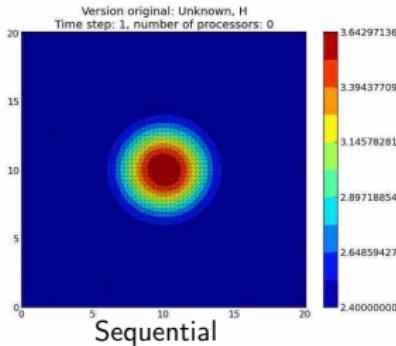
# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

## 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, ...)



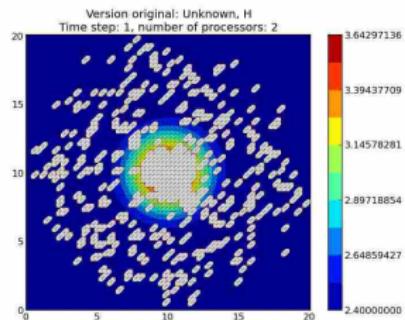
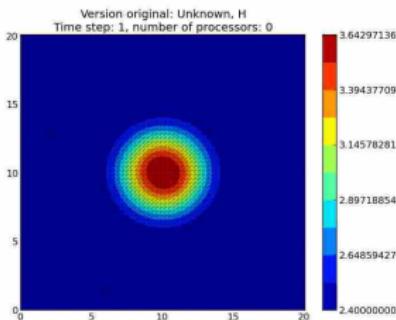
Courtesy of P. Langlois and R. Nheili  
13 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 1



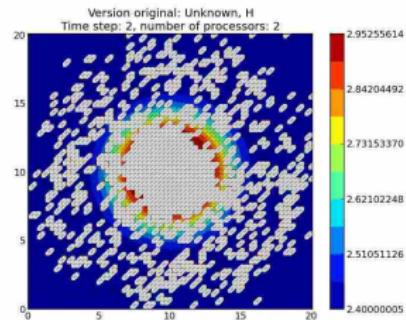
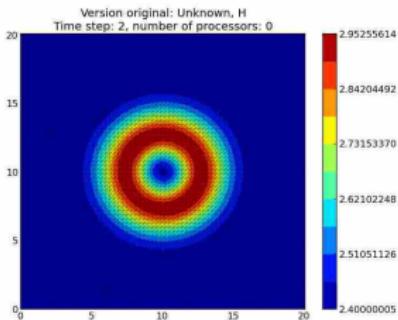
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 2



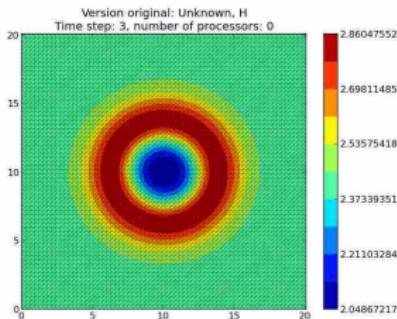
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

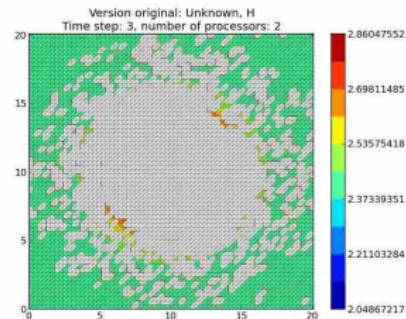
A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 3



Sequential



Parallel  $p = 2$

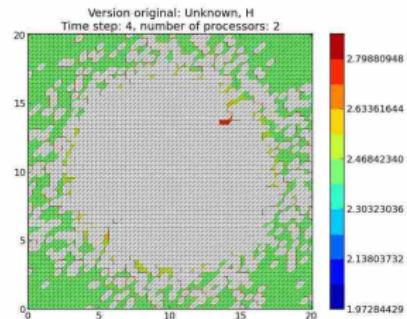
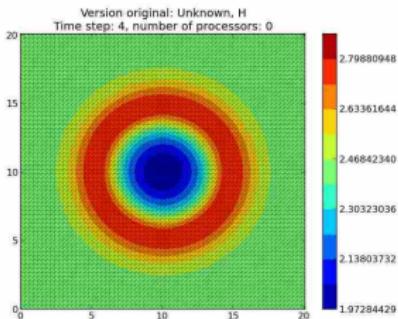
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 4



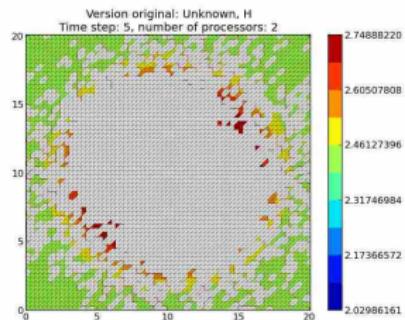
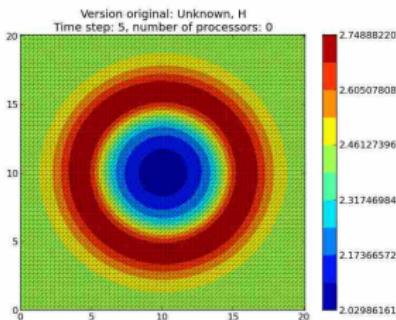
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 5



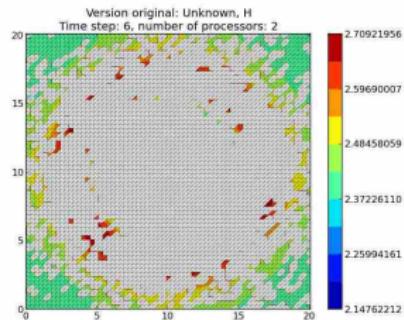
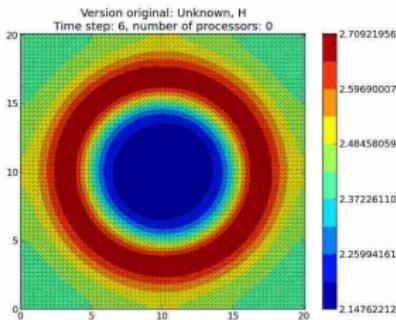
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 6



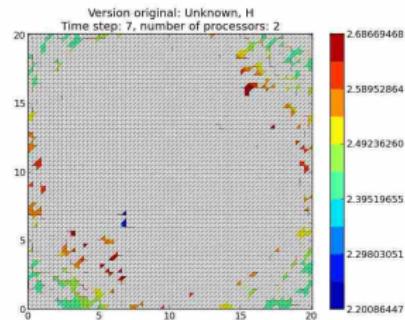
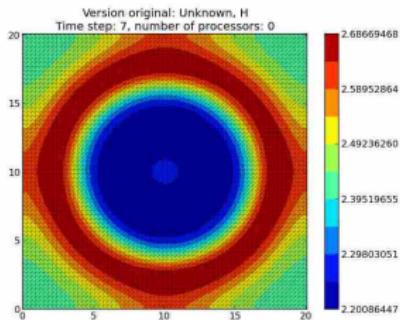
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 7



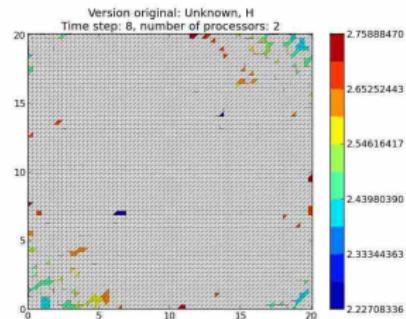
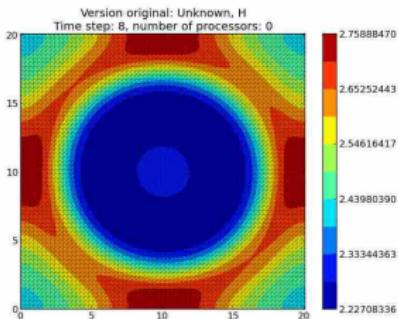
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 8



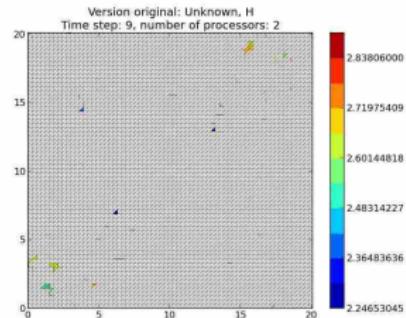
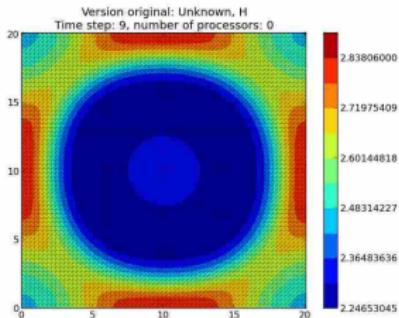
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 9



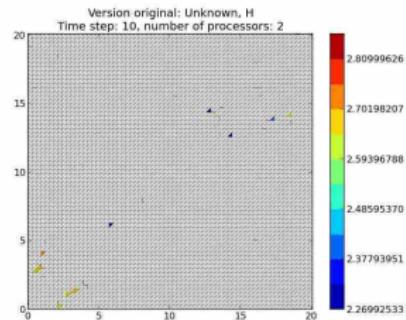
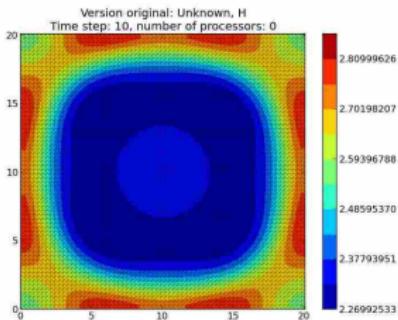
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 10



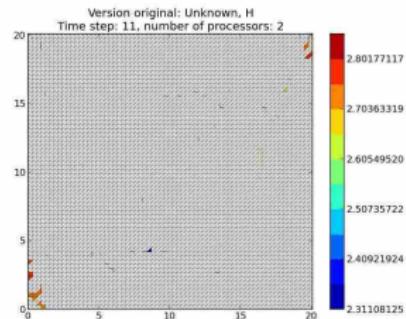
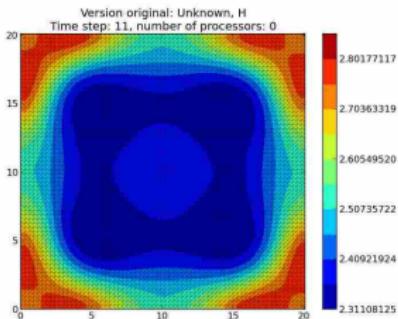
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 11



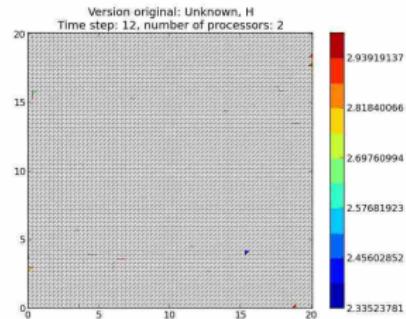
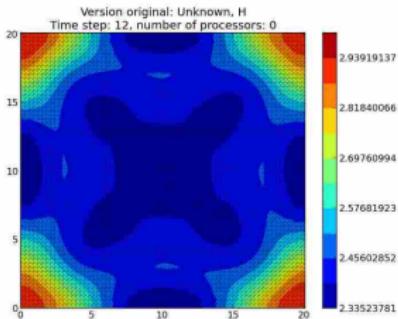
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 12



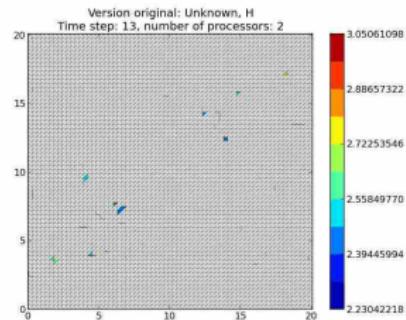
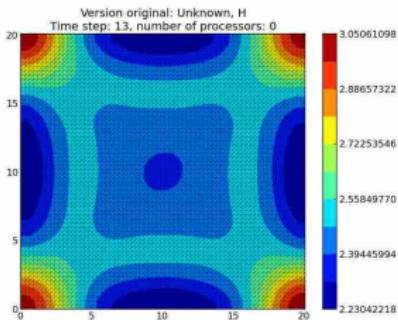
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 13



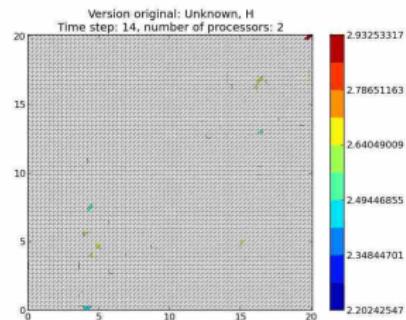
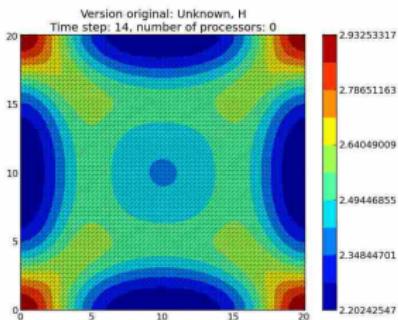
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

Numerical reproducibility?

time step = 14



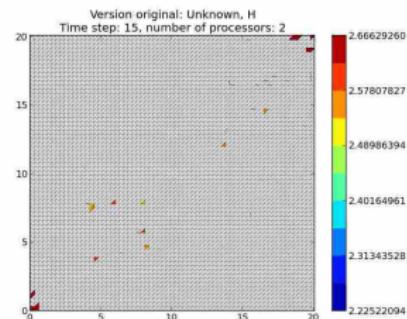
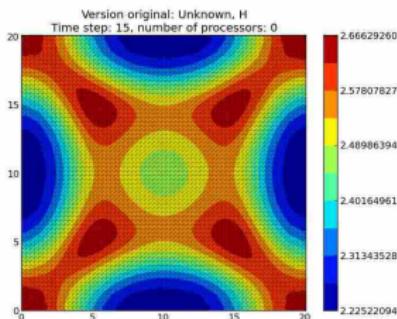
Courtesy of P. Langlois and R. Nheili  
14 / 54

# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

A white plot displays a non-reproducible value

NO numerical reproducibility!

time step = 15



Courtesy of P. Langlois and R. Nheili  
14 / 54

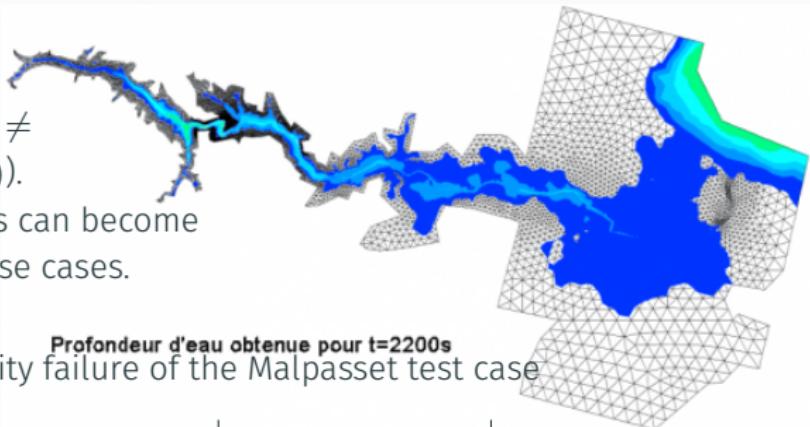
# DID I MENTION WE HAVE PARALLEL MACHINES NOWADAYS?

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

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

Profondeur d'eau obtenue pour t=2200s

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.

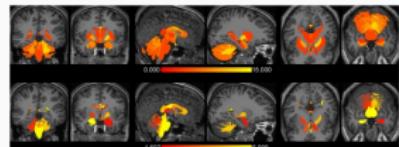
Courtesy of P. Langlois and R. Nheili

# COMPLEXITY $\rightsquigarrow$ REPRODUCIBILITY ISSUES

The slightest difference **may** have significant consequences

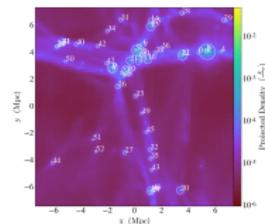
FreeSurfer

(PLOS ONE, 2012)



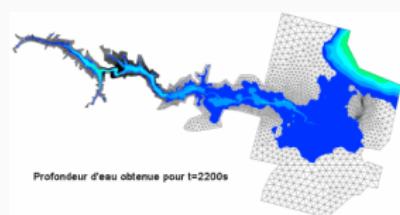
Astrophysics

(ResCuE-HPC, 2018)



Fluid Dynamics

(TELEMAC-2D)

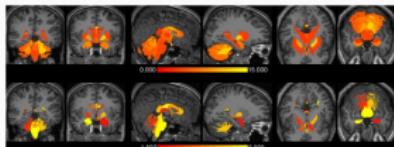


# COMPLEXITY $\rightsquigarrow$ REPRODUCIBILITY ISSUES

The slightest difference **may** have significant consequences

FreeSurfer

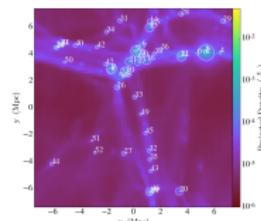
(PLOS ONE, 2012)



Software environment problem  
(permeability + versions differ)

Astrophysics

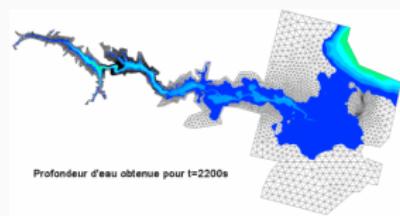
(ResCuE-HPC, 2018)



Numerical chaos problem  
Compiler optimization influence

Fluid Dynamics

(TELEMAC-2D)



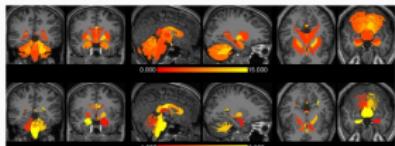
Numerical chaos problem  
Unstable and Parallel Dot-Product

# COMPLEXITY ~ REPRODUCIBILITY ISSUES

The slightest difference **may** have significant consequences

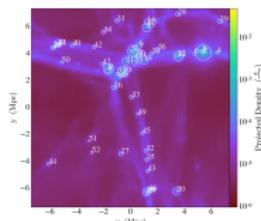
FreeSurfer

(PLOS ONE, 2012)



Astrophysics

(ResCuE-HPC, 2018)

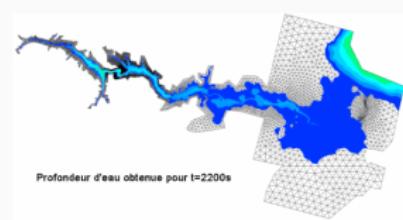


Software environment problem  
(permeability + versions differ)

*Numerical chaos problem*  
Compiler optimization influence

Fluid Dynamics

(TELEMAC-2D)



*Numerical chaos problem*  
Unstable and Parallel Dot-Product

1. Controlling your **environment** (isolation with containers)
2. Controlling your **dependencies** (versions, reproducible recipe)

It is not a good sign if your code is sensitive. You'll need:

1. Perfect control (debug)
2. Variation generation (test)
3. Locate unstable routines (sensitivity analysis)

# MAIN ISSUES

---

1. Tracking the process, parameters, and data and making them available
  - Version control, laboratory/computational notebooks, archives
2. Tracking/controling the software environment
  - Containers, reproducible environments
3. Handling numerical issues
  - Identifying sources of numerical instabilities
  - Dealing with hardware heterogeneity

## ADDITIONAL HPC CONSTRAINTS

- Software is complex and difficult to maintain
- Users (scientists) want
  - rapid development
  - rapidly evolving environment
  - old/legacy code
  - efficient code
  - process more data in a shorter time (multi-scale, multi-physics)
  - use AI/neural networks 😊
  - and reproducibility!!!
- Hardware keeps evolving and complexify



I know it's High Performance Computing, but maybe we should slow down

*There is more to life than increasing its speed*

– Gandhi.

# HANDLING SOFTWARE ENVIRONMENT CHAOS

---

# COMPLEX ECOSYSTEMS

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

---

3.5.1

# COMPLEX ECOSYSTEMS

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

3.5.1

```
1 apt show python3-matplotlib
```

Package: python3-matplotlib

Version: 3.5.1-2+b1

Source: matplotlib (3.5.1-2)

Maintainer: Sandro Tosi <morph@debian.org>

Installed-Size: 27.6 MB

Depends: libjs-jquery, libjs-jquery-ui, python-matplotlib-data (>= 3.5.1),  
python3-dateutil, python3-pil.imagetk, python3-pyparsing (>= 1.5.6),  
python3-six (>= 1.4), python3-numpy (>= 1:1.20.0), python3-numpy-abi9,  
python3 (<< 3.11), python3 (>= 3.9~), python3-cycler (>= 0.10.0),  
python3-fonttools, python3-kiwisolver, python3-packaging, python3-pil,  
python3:any, libc6 (>= 2.29), libfreetype6 (>= 2.2.1),  
libgcc-s1 (>= 3.3.1), libqhull-r8.0 (>= 2020.1), libstdc++6 (>= 11)

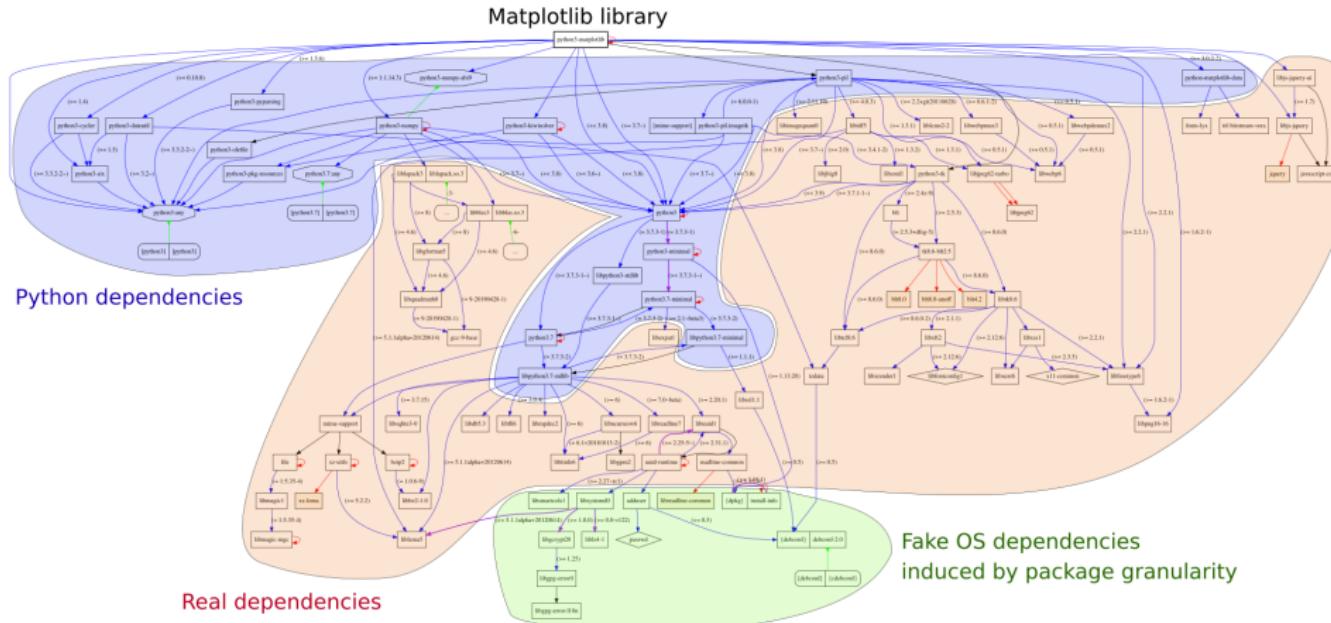
Recommends: python3-tk

Suggests: dvipng, ffmpeg, fonts-staypuft, ghostscript, gir1.2-gtk-3.0, inkscape,  
ipython3, librsvg2-common, python-matplotlib-doc, python3-cairoffi,  
python3-gi, python3-gi-cairo, python3-gobject, python3-pyqt5,  
python3-scipy, python3-sip, python3-tornado, texlive-extra-utils,  
texlive-latex-extra

Enhances: ipython3

# COMPLEX ECOSYSTEMS

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



python3-scipy, python3-sip, python3-tornado, texlive-extra-utils,  
texlive-latex-extra

Enhances: ipython3

## POTENTIAL SOLUTIONS: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



Automatic tracking

# POTENTIAL SOLUTIONS: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



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>`

# POTENTIAL SOLUTIONS: CONTAINERS AND PACKAGE MANAGERS

The good



The bad



The ugly



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)

# POTENTIAL SOLUTIONS: CONTAINERS AND PACKAGE MANAGERS

The good



Automatic tracking

Containers

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

Package managers (the ugly 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)

The bad



## DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2`

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2` (commit `78c293967830979f988b39cea2587a7c26b`),

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2` (commit `78c293967830979f988b39cea2587a7c26b`),  
(built/run in this environment with such and such options)

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2` (commit `78c293967830979f988b39cea2587a7c26b`),  
(built/run in this environment with such and such options)
- `v_1.5` or `v_1.4.3` are often *better* than `v_1.4.2`

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2` (commit `78c293967830979f988b39cea2587a7c26b`),  
(built/run in this environment with such and such options)
- `v_1.5` or `v_1.4.3` are often *better* than `v_1.4.2`
- Convenient but very problematic for reproducibility

# DESCRIBING SOFTWARE/HARDWARE DEPENDENCIES

**Runtime dependencies** interpreter, libraries, other programs

**Build dependencies** compilers, headers, `autotools/cmake`, etc.

**Version** a loose description

- `v_1.4.2` (commit `78c293967830979f988b39cea2587a7c26b`),  
(built/run in this environment with such and such options)
- `v_1.5` or `v_1.4.3` are often *better* than `v_1.4.2`
- Convenient but very problematic for reproducibility

**Package set** order actually matters (side effects)

- Installing **A**, then **B** may be different from installing **B**, then **A**

# DEBIAN DEPENDENCIES

```
1 dpkg --status python3-matplotlib
```

```
Package: python3-matplotlib
Version: 3.6.3-1+b1
Source: matplotlib (3.6.3-1)
Depends: libjs-jquery, libjs-jquery-ui, python-matplotlib-data (>= 3.6.3),
          python3-dateutil, python3-pil.imagetk, python3-pyparsing (>= 1.5.6),
          python3-six (>= 1.4), python3-numpy (>= 1:1.22.0), python3-contourpy,
          python3 (<< 3.12), python3 (>= 3.11~), python3-numpy-abi9,
          python3-cycler (>= 0.10.0), python3-fonttools, python3-kiwisolver,
          python3-packaging, python3-pil, python3:any, libc6 (>= 2.34),
          libfreetype6 (>= 2.2.1), libgcc-s1 (>= 3.3.1),
          libqhull-r8.0 (>= 2020.1), libstdc++6 (>= 11)
```

# DEBIAN DEPENDENCIES

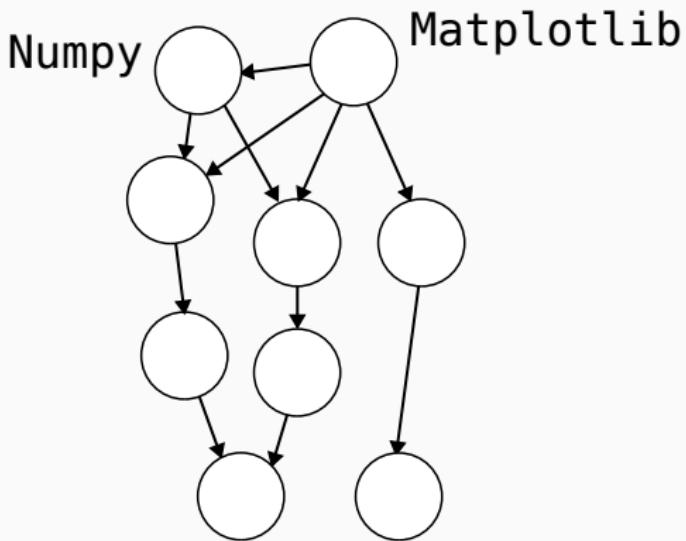
```
1 dpkg --status python3-matplotlib
```

```
Package: python3-matplotlib
Version: 3.6.3-1+b1
Source: matplotlib (3.6.3-1)
Depends: libjs-jquery, libjs-jquery-ui, python-matplotlib-data (>= 3.6.3),
          python3-dateutil, python3-pil.imagetk, python3-pyparsing (>= 1.5.6),
          python3-six (>= 1.4), python3-numpy (>= 1:1.22.0), python3-contourpy,
          python3 (<< 3.12), python3 (>= 3.11~), python3-numpy-abi9,
          python3-cycler (>= 0.10.0), python3-fonttools, python3-kiwisolver,
          python3-packaging, python3-pil, python3:any, libc6 (>= 2.34),
          libfreetype6 (>= 2.2.1), libgcc-s1 (>= 3.3.1),
          libqhull-r8.0 (>= 2020.1), libstdc++6 (>= 11)
```

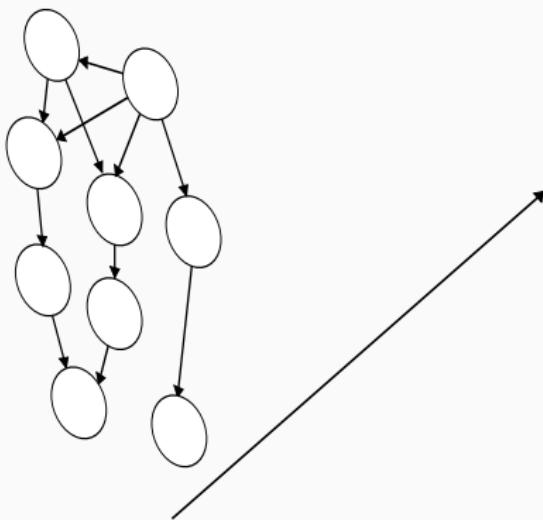
On a given day:

- Several versions of each package are available on the server
- Installing the latest version of a package may require upgrading some other packages

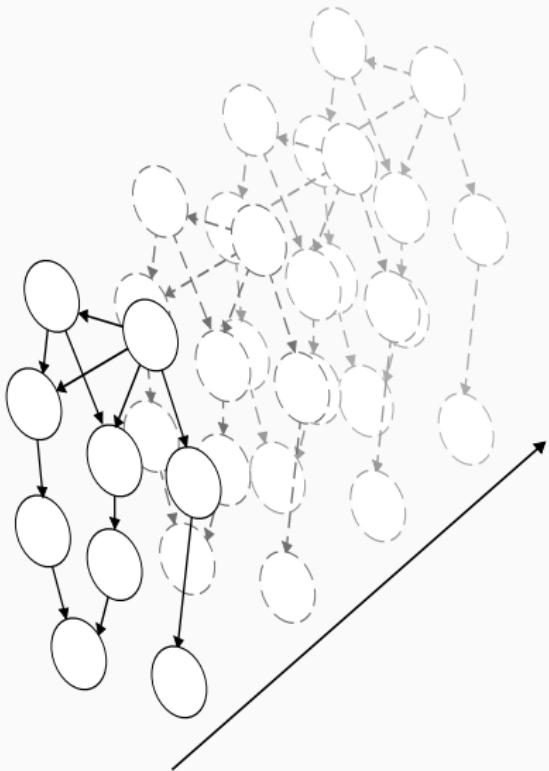
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



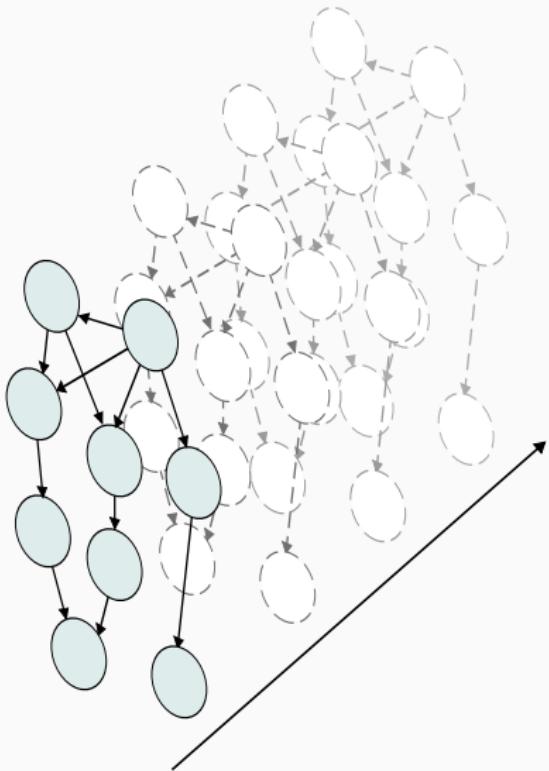
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



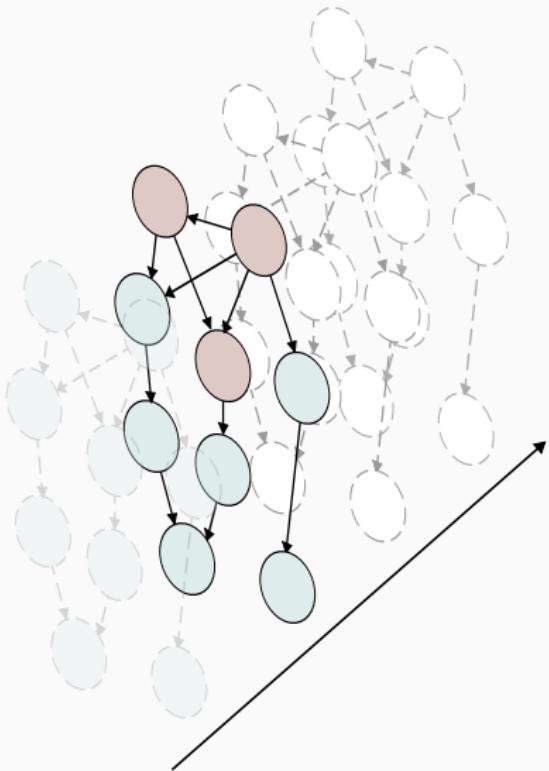
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



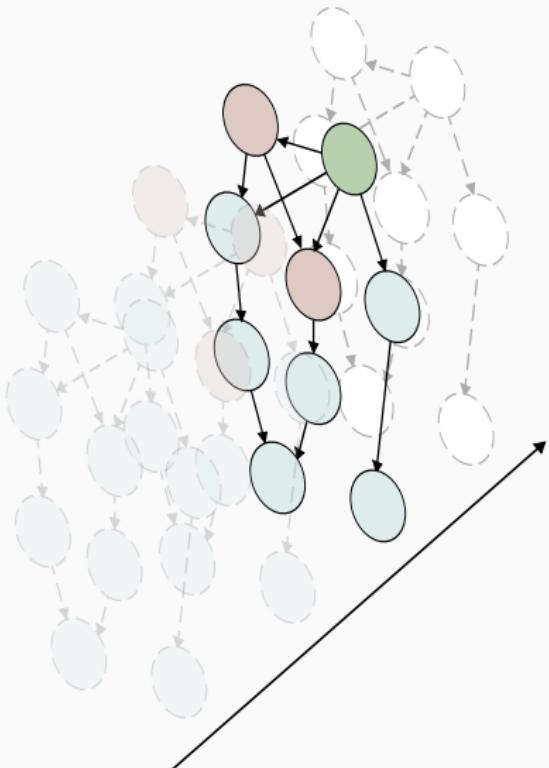
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



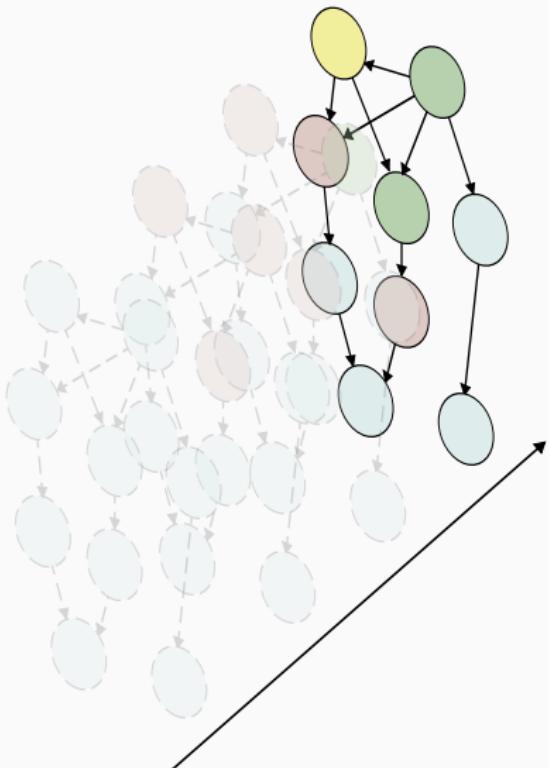
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



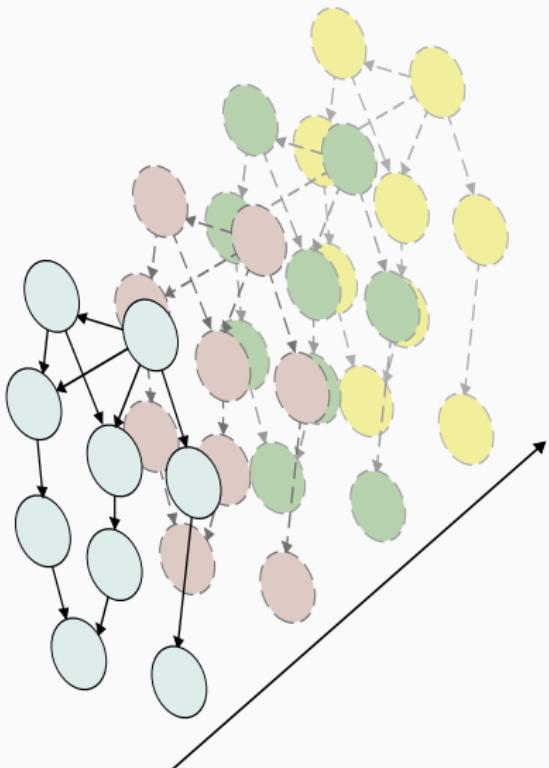
## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



## LOOSE VS. STRICT DEPENDENCIES IN PICTURE



## GUIX IN A NUTSHELL (1/2)

GUIX is not a ~~container technology~~, it is meant for **package management!**

## GUIX IN A NUTSHELL (1/2)

GUIX is not a ~~container technology~~, it is meant for **package management!**

- A GUIX **recipe** (aka `manifest.scm`)

```
1 (specifications->manifest
2   (list "hello" "coreutils"))
1 guix shell -C -m manifest.scm -- ls /    # -C = --container
```

## GUIX IN A NUTSHELL (1/2)

GUIX is not a ~~container technology~~, it is meant for **package management!**

- A GUIX **recipe** (aka `manifest.scm`)

```
1 (specifications->manifest
2   (list "hello" "coreutils"))

1 guix shell -C -m manifest.scm -- ls /  # -C = --container
```

- The description of the **versions** is in the `manifest.scm`

```
1 guix describe -f channels > channels.scm
```

```
1 (list (channel
2       (name 'guix)
3       (url "https://git.savannah.gnu.org/git/guix.git")
4       (branch "master")
5       (commit
6         "d09a4cc7c739f4201821623f62c69bcef9c20f52")
7       (introduction
8         (make-channel-introduction
9           "9edb3f66fd807b096b48283debdcccfea34bad"
10          (openpgp-fingerprint
11            "BBB0 2DDF 2CEA F6A8 0D1D E643 A2A0 6DF2 A33A 54FA")))))
```

# GUIX IN A NUTSHELL (1/2)

GUIX is not a ~~container technology~~, it is meant for **package management!**

- A GUIX **recipe** (aka `manifest.scm`)

```
1 (specifications->manifest
2   (list "hello" "coreutils"))

1 guix shell -C -m manifest.scm -- ls / # -C = --container
```

- The description of the **versions** is in the `manifest.scm`

```
1 guix describe -f channels > channels.scm
```

```
1 (list (channel
2       (name 'guix)
3       (url "https://git.savannah.gnu.org/git/guix.git")
4       (branch "master")
5       (commit
6         "d09a4cc7c739f4201821623f62c69bcef9c20f52")
7       (introduction
8         (make-channel-introduction
9           "9edb3f66fd807b096b48283debdcccfea34bad"
10          (openpgp-fingerprint
11            "BBB0 2DDF 2CEA F6A8 0D1D E643 A2A0 6DF2 A33A 54FA")))))
```

- A **time-machine**

```
1 guix time-machine -C channels.scm -- shell -m manifest.scm -- ls /
```

## GUIX IN A NUTSHELL (2/2)

- Under the hood:
  - A **deamon** compiles everything in a collection of directories  
`/gnu/store/8fpk2cja3f07xls48jfnpgrzrljpqivr-coreutils-8.32/`
  - All the directories are assembled (with symlinks) in a  
`/gnu/store/j5964hh821p2h5mcadpvj16l1m9330gv-profile/` dir
  - Environment variables (**PATH**, **LD\_LIBRARY\_PATH**, ...) are updated accordingly

## GUIX IN A NUTSHELL (2/2)

- Under the hood:
  - A **deamon** compiles everything in a collection of directories  
`/gnu/store/8fpk2cja3f07xls48jfnpgrzrljpqivr-coreutils-8.32/`
  - All the directories are assembled (with symlinks) in a  
`/gnu/store/j5964hh821p2h5mcadpvj16l1m9330gv-profile/` dir
  - Environment variables (**PATH**, **LD\_LIBRARY\_PATH**, ...) are updated accordingly
- Several **containerization** options

```
1 guix shell --container coreutils -- ls
2 guix shell coreutils -- ls  # Fully permeable: expends the env, ...
```

## GUIX IN A NUTSHELL (2/2)

- Under the hood:
  - A **deamon** compiles everything in a collection of directories  
`/gnu/store/8fpk2cja3f07xls48jfnpgrzrljpqivr-coreutils-8.32/`
  - All the directories are assembled (with symlinks) in a  
`/gnu/store/j5964hh821p2h5mcadpvj16l1m9330gv-profile/` dir
  - Environment variables (**PATH**, **LD\_LIBRARY\_PATH**, ...) are updated accordingly
- Several **containerization** options

```
1 guix shell --container coreutils -- ls
2 guix shell coreutils -- ls # Fully permeable: expends the env, ...
```

- Various **export** formats (**docker**, **squashfs**, **debian**, **tarball**, **module**, **relocatable**...)

```
1 guix pack --format=docker --save-provenance -m manifest.scm
```

Allows to carefully control/nest environments and adapt to your context

- For more information: <https://hpc.guix.info/>
  - Lastest Guix Workshop on HPC was held in Bordeaux on Nov. 7, 2024, after the JCAD
  - Check out the **Café Guix**

# HANDLING NUMERICAL CHAOS

---

## IS THERE A BASELINE TO COMPARE WITH ?

**Baseline** From previously established results (e.g. a sequential test-case with **-00**), checked for indirect confirmations (e.g. physical properties such as energy are conserved)

- Until an error is found...

**Bitwise reproducibility** may be out of reach

CESM  $\approx$  two million lines across 13,000 subroutines and 3,000 functions

### Community Earth System Model - Ensemble Consistency Test (CESM-ECT)

A suite of statistical tests to determine whether a new simulation set up (new machine, compiler, etc.) is statistically distinguishable from an accepted ensemble.

- Then try to perform root-cause-analysis of numerical discrepancy by comparing snapshots of specific variables.

Still, bitwise reproducibility makes debugging easier. Is there anything we can do in this direction ?

# REPRODUCIBLE BLAS

Optimized BLAS involve FMA, tiling, cache optimization, loop unrolling, vectorization (SSE, AVX, AVX-512), dynamic resource allocations...

Use accumulators, binning, and distillation: *TwoProd*, *FastAccSum*, *HybridSum*, *iFastSum*...

**ReproBLAS** <https://bebop.cs.berkeley.edu/reproblas/> (2013-2018)

- Matrix-matrix multiply, matrix-vector multiply, and reductions (namely summation, absolute value summation, dot product, and 2-norm) for the 4 basic types of data: double, float, double complex, and float complex.
- Future versions (under development) will include: trsv, trsm, OpenMP, MPI

**ExBLAS** Exact (fast, accurate, and reproducible) BLAS (2014-2016)

- Intel Xeon Phi many-core accelerator, and both NVIDIA and AMD GPUs
- ExSUM, ExDOT, ExGEMV, ExTRSV, ExGEMM

**RARE-BLAS** Reproducible, Accurately Rounded and Efficient BLAS (2016-2017)

- OpenMP and MPI

CADNA and VERROU (2014...) stochastic rounding through `valgrind`  
(see talk by C. Denis)

- Each floating point operation is computed three times using a **random rounding** modes. The number of exact significant digits is then estimated from these three samples
- **Delta-Debugging** to locate which parts of the analyzed source code are likely to be responsible numerical instabilities.
- Provides **location/type** (e.g. **UNSTABLE CANCELLATION**) of the detected numerical instability
- **Common sources of instability**
  - Parallel dot products + `MPI_Reduce`
  - Aggregation of contributions from the ranks
- After the round-off error **analysis**
  - Replace or modify algorithms leading to poor precision
  - Use compensated algorithms
  - Use multi-precision arithmetic (for example MPFR)

# CODE BISECTION AT COMPILE TIME

FLiT <https://pruners.github.io/flit/>  
(compare compilation options) at  
file/symbol granularity

In practice, this modality of search has helped us successfully attribute root causes down to a small set of functions. For example, in [...], FLiT-based testing revealed that a compiler optimization level that involved the use of AVX2, FMA, and higher precision intermediate floating-point values produced a result that had a relative difference of 193% from the baseline of `g++ -O2`. The  $L_2$  norm over the mesh went from approximately 5 to 15 after the optimizations. Using Symbol Bisect, the problem was located to be within one simple function that calculates  $M = M + aAA^T$

– Keeping Science on Keel When Software Moves, CACM 2021

Figure 3. Performance profile of compilations of Example 9 from MFEM. The compilations with the fastest bitwise equal and fastest overall speeds are labeled.

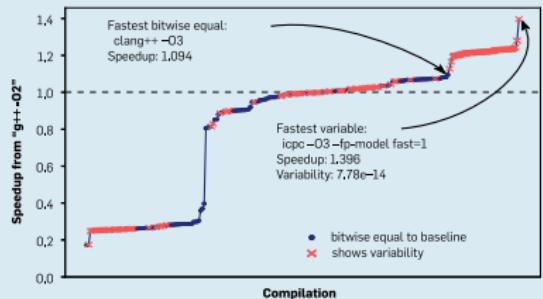


Figure 4. File Bisect and Symbol Bisect.

File Bisect (above) links object files from two compilations to make a mixed executable.

Symbol Bisect (below) mixes function symbols from two different compiled versions of the same source file.

Some function symbols are changed into weak symbols (empty small circles) and are discarded by the linker,

This allows mixing of functions from two compilations of a single source file.

## File Bisect

- baseline (e.g., g++ -O0)
- under test (e.g., g++ -O3)
- final executable (mixed)

## Symbol Bisect

- baseline (e.g., g++ -O0)
- under test (e.g., g++ -O3)
- final executable (mixed)

## MY INTUITION / OPINION

Using a task based model (*à la* StarPU) may help

- More **modular** code
- Better **exploitation of (heterogeneous)** computing **resources**
- Easier control of **asynchronism** in algorithms
- Improve the *ability* to **change precision**, to hook **inspectors** on numerical stability, etc.

## A BIT OF ADVERTISING

---

# GLOBAL REPRODUCIBILITY NETWORKS

<https://www.ukrn.org/global-networks/>



- About
- Community
- Resources
- Activities
- Join UKRN
- Contact
- Search icon

## Global Networks

Outside the UK? Find a [Reproducibility Network](#) in your area

[See full Global Networks Statement](#)

### Global Reproducibility Networks

A Reproducibility Network (RN) is a national, peer-led consortium of researchers that aims to promote and ensure rigorous research practices by establishing appropriate training activities, designing and evaluating research improvement efforts, disseminating best practice and working with stakeholders to coordinate efforts across the sector. RNs aim for broad disciplinary representation and an intensive interdisciplinary dialogue (e.g., with funding agencies, publishers, learned societies and other sectoral organisations, as well as researchers from all disciplines and across all career stages).

To reach as many researchers as possible, and to operate as efficiently as possible, we are keen to support other countries interested in creating similar networks. If you are interested in setting up a national RN, or finding out who in your country is working towards this, please [contact us](#).



<http://www.recherche-reproductible.fr/>

RECHERCHE REPRODUCTIBLE

COMMUNAUTÉ ACTIVITÉS RESSOURCES 

## Bienvenue sur le réseau de la recherche reproductive

*Pour un chercheur, il n'y a rien de plus frustrant que l'impossibilité de reproduire des résultats majeurs obtenus quelques mois auparavant. Les causes de ce type de déconvenues sont multiples et parfois pernicieuses. Ce phénomène participe à ce que certains identifient comme une "crise de la reproductibilité de la recherche". — Vers une recherche reproductive, Desquibet et al., 2019.*

Le réseau français de la recherche reproductive est un réseau national composé d'universitaires intéressés par l'étude des facteurs qui contribuent à la robustesse de la recherche, la promotion des activités de formation et la diffusion des bonnes pratiques ainsi que des recommandations. Les questions de reproductibilité touchent toutes les disciplines et le réseau vise une large représentation disciplinaire. Si vous êtes intéressé, envoyez-nous un [courriel](#) et rejoignez le réseau !

## Prochains événements

Mar 26, 2024 [Journées du réseau français de recherche reproductive 2024](#)

Nov 9, 2023 [The role of replication in scientific controversies](#)

Nov 8, 2023 [Workshop on Reproducible Software Environments for Research and High-Performance Computing](#)

# FRENCH REPRODUCIBILITY NETWORK DAYS: 1ST EDITION

March 2023, Institut Pasteur, Paris



Toward an *inter-disciplinary* look on our respective approaches and definitions

- Observational, experimental, statistical, computational, etc.

Identify volunteers, possible collaboration topics, diversity of concerns/practices, educational resources,...

DECINE • NEUROSCIENCES • ECONOMIE • PHILOSOPHIE  
ISTOIRE • DROIT • ECONOMIE • PHILOSOPHIE  
GENETIQUE • ASTRONOMIE • ECOLOGIE  
ANTHROPOLOGIE • PSYCHOLOGIE • PHYSIQUE  
CHIMIE • BIOLOGIE • SOCIOLOGIE • PHYSIQUE  
MEDECINE • NEUROSCIENCES • INFORMATIQUE  
HISTOIRE • DROIT • ECONOMIE • CHIMIE  
RECHERCHE • REPRODUCTIBLE • FRANCE  
THEMATIQUES • MECANIQUE • GENETIQUE  
ECONOMIE • PHILOSOPHIE • ANTHROPOLOGIE  
PSYCHOLOGIE • ECOLOGIE • DROIT • CHIMIE  
SOCIOLOGIE • ASTROPHYSIQUE  
NEUROSCIENCES • INFORMATIQUE  
PIDEMOLOGIE • HISTOIRE • ECONOMIE  
THEMATIQUES • MECANIQUE • GENETIQUE  
**INSTITUT PASTEUR**  
**8 & 9 MARS 2023**  
RECHERCHE • ROBUSTE • REPLICABLE  
IENCE • EXPERIENCE • CODE • OBSERVATION  
RALISABLE • EXPERIENCE • CODE • ANALYSE  
STATISTIQUES • RESULTAT • METHODE • ANALYSE  
ES • NUMERIQUE • RESULTAT • METHODE • ANALYSE  
SCIENCE • ETAT • ECHANTILLON • ROBUSTE  
ICABLE • GENERALISABLE • DES EXPERIMENTATION  
O • OBSERVATION • STATISTIQUES • ANALYSE  
NEES • EXPERIENCE • SCIENCE • REPLICABLE • RO  
RECHERCHE • SCIENCE • REPLICABLE • EXPERIMENTATION  
CENTRALISABLE • EXPERIENCE • ANALYSE

# FRENCH REPRODUCIBILITY NETWORK DAYS: 2ND EDITION

GENETIQUE•ASTRONOMIE•PHILOSOPHIE•ANTHROPO  
CHOLOGIE•ECOLOGIE•CHIMIE•BIOLOGIE•SOCIOLOGIE  
•MEDECINE•NEUROSCIENCES•INFORMATIQUE•HISTOI  
ECONOMIE•RECHERCHE-PRODUCTIBLE.FR•HISTO  
MATIQUES•MECANIQUE•GENETIQUE•ASTRONOMIE•E  
CHIMIE•BIOLOGIE•SOCIOLOGIE•PHYSIQUE•MEDECINE•N  
ANCES•INFORMATIQUE•HISTOIRE•DROIT•ECONOMIE•M

## Journées du Réseau Français de Recherche Reproductible

### Grenoble

**26-28 Mars 2024**

SCIENCE•ROBUSTE•REPLICABLE•GENERALISABLE•EX  
E•CODE•OBSERVATION•STATISTIQUES•ANALYSE•DON  
NERIQUE•RESULTAT•METHODE•ECHANTILLON•ROB  
CABLE•GENERALISABLE•EXPERIENCE•CODE•OBSERV  
ATISTIQUES•ANALYSE•DONNEES•EXPERIENCE•RESUL  
T•RECHERCHE•SCIENCE•REPLICABLE•ROBUSTE•RI  
•GENERALISABLE•EXPERIENCE•CODE•OBSERVATION

3rd Edition: March 3-4, 2025 @ Lyon

## MOOCs on REPRODUCIBLE RESEARCH:

MOOC Reproducible Research: Methodological principles for a transparent science, Inria Learning Lab

- Konrad Hinsen, Christophe Pouzat
- 3rd Edition: March 2020 – ... (16,800+)
- Notebooks, version control, simple data formats



# MOOCs on REPRODUCIBLE RESEARCH:

MOOC Reproducible Research: Methodological principles for a transparent science, Inria Learning Lab

- Konrad Hinsen, Christophe Pouzat
- 3rd Edition: March 2020 – ... (16,800+)
- Notebooks, version control, simple data formats



MOOC Reproducible Research II: Practices and tools for managing computations and data (May-Sep 2024, ≈ 2,000)

- Managing data (`FITS/HDF5, git annex`)
- Software environment control (`docker, singularity, guix`)
- Scientific workflow (`make, snakemake`)
- Statistics, Numerical Chaos