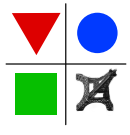
Open/Reproducible Science and Free Software
Sharing codes and results

Stéphane Popinet

Institut Jean le Rond ∂'Alembert, CNRS & Sorbonne Université



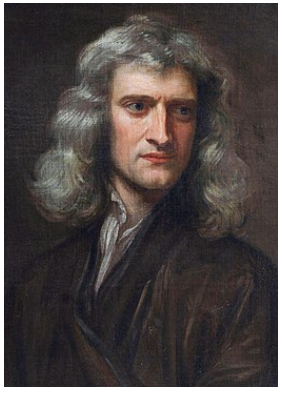
∂'Alembert
Institut Jean le Rond d'Alembert

Outline

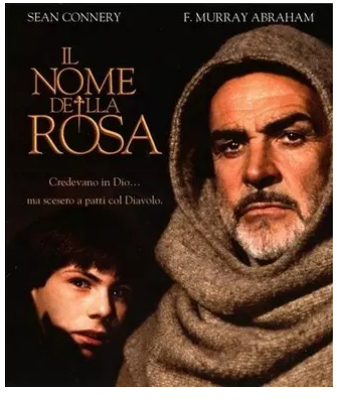
1. A brief history of "Free Software" and links with "Open Science"
2. Reproducible and collaborative science: why?
3. How to do it: an example with Basilisk
4. Documentation and "Literate programming"
5. Conclusions

"Open/Reproducible" is intrinsic to the scientific method (Roger Bacon, ca 1250)

"The Republic of Letters" a central process of the Enlightenment in the 17th and 18th centuries




The concept of *Res Publica Literaria* dates back to the 15th century (at least)



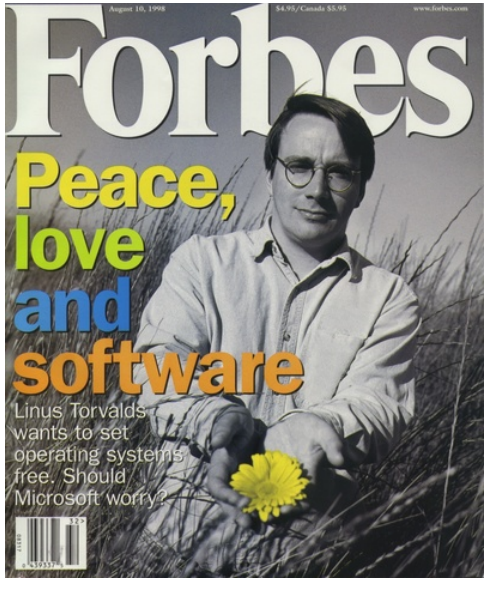
"The Name of the Rose", Umberto Ecco, 1980

Richard Stallman, libertarianism, hacker culture and the Free Software Foundation (1985)

The GNU is Not Unix (GNU) project (1983): en entirely *free* (as in freedom) operating system

Linux versus Microsoft: a *paradigm shift* in software development

Cover picture for the 2002 book:
Free as in Freedom: Richard Stallman's Crusade for Free Software



Linus Torvalds, 1998
Creator of Linux (and git)

1971 ARPANET an american network of military and academic sites

1972 Thompson, Kernighan and Ritchie create C, UNIX (and emails)

1980s the Internet emerges from international extensions of ARPANET

1990 (Sir) Tim Berners-Lee creates the WWW while at CERN by linking hypertext (i.e. HTTP/HTML) with the internet

The primary goal was easier sharing of information (documentation) between CERN engineers and scientists

Standardisation is the key: Berners-Lee is now head of the World Wide Web Consortium (W3C)

1991 Linus Torvalds releases Linux version 0.0.2

1990s – now Free/Open Source Software runs most of the web (linux servers, apache etc.) 85% of smartphones and 100% of supercomputers are linux-powered

2000s the GAFA(M) would not be what they are without Free Software

Karl Popper, Friedrich Hayek, Milton Friedman, Margaret Thatcher, Ronald Reagan

1944 *The Road to Serfdom*, Hayek (foundations of neo-liberalism)

1945 *The Open Society and Its Enemies*, Popper

1948 *Individualism and Economic Order*, Hayek

1963 *Conjectures and Refutations: The Growth of Scientific Knowledge*, Popper

1973 11 September, Pinochet + CIA overthrow & kill Salvador Allende in Chile

1974 "Nobel prize in economics" awarded to Hayek and Myrdal (a Keynesian)

1976 Milton Friedman wins controversial Nobel prize in economics

A geek's CV

1983 My first computer is a Sinclair ZX-81 (4 kilobytes of RAM)

1988 Upgrade to an Atari 1040 ST (1 MHz clock, 1 megabyte of RAM!)

1993 First PC with Linux (version 1.2), 66 MHz clock, 4 MB RAM

1999 The GNU Triangulated Surface (GTS) library is one of the first Free Software project released on SourceForge (once a famous "forge")

2001 First release of the Gerris Flow Solver

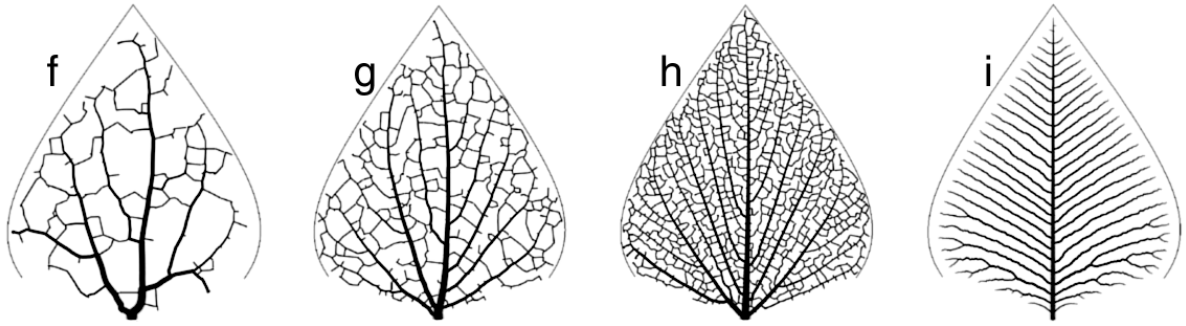
2007 Gerris wins second place in the first "New Zealand Open Source Awards"

2013 First release of the Basilisk library

2022 Member of the jury for the first Open Science Awards for Open Source Research Software of the Ministry of Research and Higher Education

2023 100 international participants for the 6th Basilisk/Gerris Users' Meeting in Paris (5–7th July)

GTS: Computational Geometry (triangulations, Voronoi, differential geometry, optimisation etc.)



Modeling and visualization of leaf venation patterns, Runions et al, SIGGRAPH, 2005

Gerris/Basilisk: Computational (Fluid) Mechanics, partial differential equations, numerical analysis etc.

> 700 users subscribed to the Basilisk forum

> 100 scientific papers (JFM, JCP, PRF) published every year depend on Basilisk

- Science has always been open because it is a collective and collaborative endeavour
- Reproduction of scientific results a pre-requisite of the scientific method
  - Can you reproduce your own results?
  - Can somebody else reproduce your results?

    Journal articles often standardise how *experiments* should be reported
  - Can other scientists **trust** your results?
- Possible but rarely done for scientific software and numerical methods
  - Trade secrets?
  - Dirty secrets?
  - or laziness...

. . . because "sharing is good" (altruism) and "sharing is good *for me*" (selfishness)

- Potential drawbacks
  - Sharing requires additional work
  - Sharing code is not considered as "serious" as publishing papers
  - Someone might *steal my ideas*!
  - Someone could say my code/data etc. are not good enough
- Potential benefits
  - The additional work (documentation, reproducibility) will benefit me first
  - My work (including publications) will be more visible
  - My ideas/code will be used and this will create new opportunities
  - Someone can help me improve my code, and can contribute new code, ideas, documentation etc.

Anthropologists Bronisław Malinowski (1884–1942) and Marcel Mauss (1872–1950) studied the culture of Trobriand Islanders (off Papua New Guinea).

The *social status* of the Trobriand Islanders seems to be directly linked with the value of the *gifts* they make to the community.

The gain in *social status* is (meant to be) the only reward.

At varying degrees this idea is common to many cultures and religions:
charity, philanthropy, donations etc.

The most important aspect of a successful (Open Source / Free Software) project is the **community** (i.e. people) not the source code $\Rightarrow$ **communication!**

The basilisk.fr website

- a freely editable **wiki**: anybody can contribute **documented code**
- the code can be executed directly on the server (via the web browser)
- the website **is the code**
- the website/code is version-controlled and can be retrieved/mirrored/modified locally using standard tools (as a hierarchy of files editable with a text editor)
- *Continuous integration* is encouraged $\Rightarrow$ more than 300 test cases

Other communication tools

- an active user forum / mailing list
- monthly interactive video presentations (2$\times$30 minutes) and an archive
- Basilisk Users' Meetings (2011, 2012, 2014, 2017, 2019, 2023)

Literate Programming: Donald Knuth (1984) "programmers are illiterate"

A solver for the Saint-Venant equations

Note that the multilayer solver provides the same functionality and should be prefered for most applications.

The Saint-Venant equations can be written in integral form as the hyperbolic system of conservation laws

$$\partial_t \int_\Omega \mathbf{q} d\Omega = \int_{\partial\Omega} \mathbf{f}(\mathbf{q}) \cdot \mathbf{n} d\partial\Omega - \int_\Omega hg\nabla z_b$$

where Ω is a given subset of space, $\partial\Omega$ its boundary and $\mathbf{n}$ the unit normal vector on this boundary. For conservation of mass and momentum in the shallow-water context, Ω is a subset of bidimensional space and $\mathbf{q}$ and $\mathbf{f}$ are written

$$\mathbf{q} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix}, \quad \mathbf{f}(\mathbf{q}) = \begin{pmatrix} hu_x & hu_y \\ hu_x^2 + \frac{1}{2}gh^2 & hu_xu_y \\ hu_xu_y & hu_y^2 + \frac{1}{2}gh^2 \end{pmatrix}$$

where $\mathbf{u}$ is the velocity vector, h the water depth and z_b the height of the topography. See also Popinet, 2011 for a more detailed introduction.

User variables and parameters

The primary fields are the water depth h, the bathymetry z_b and the flow speed $\mathbf{u}$. η is the water level i.e. $z_b + h$. Note that the order of the declarations is important as z_b needs to be refined before h and h before η.

```
scalar zb[], h[], eta[];
vector u[];
```

The only physical parameter is the acceleration of gravity *G*. Cells are considered "dry" when the water depth is less than the *dry* parameter (this should not require tweaking).

```
double G = 1.;
double dry = 1e-10;
```

The corresponding "unformatted" source

File Edit Options Buffers Tools C Help

Save Undo

```
/**
# A solver for the Saint-Venant equations

<div class="message">
Note that the [multilayer solver](layered/hydro.h) provides the same functionality and
should be prefered for most applications.</div>

The
[Saint-Venant equations](http://en.wikipedia.org/wiki/Shallow_water_equations) can be
written in integral form as the hyperbolic system of conservation laws
$$
  \partial_t \int_{\Omega} \mathbf{q} d \Omega =
  \int_{\partial \Omega} \mathbf{f} (
  \mathbf{q}) \cdot \mathbf{n}d \partial
  \Omega - \int_{\Omega} hg \nabla z_b
$$
...

## User variables and parameters

The primary fields are the water depth $h$, the bathymetry $z_b$ and the flow speed
$\mathbf{u}$. $\eta$ is the water level i.e. $z_b + h$. Note that the order of the
declarations is important as $z_b$ needs to be refined before $h$ and $h$ before $\eta$. */

scalar zb[], h[], eta[];
vector u[];

/**
The only physical parameter is the acceleration of gravity *G*. Cells are considered "dry"
when the water depth is less than the *dry* parameter (this should not require tweaking). */

double G = 1.;
double dry = 1e-10;
```

-:--- **saint-venant.h** All L18 (C/*l Abbrev)

Stéphane Popinet. A vertically-Lagrangian, non-hydrostatic, multilayer model for multiscale free-surface flows. Journal of Computational Physics, *418:109609, 2020.*

All results were obtained using a timestep given by the stability condition (35), with the exception of the Green–Naghdi model which proved unstable, as expected, and had to use the more restrictive hydrostatic condition (17). Note that for the maximum value of $\hat{k}H \approx 60$ displayed on the figure and for 128 grid points per wavelength, the ratio of the non-hydrostatic timestep to hydrostatic timestep is approximately $\sqrt{H/\Delta} = \sqrt{128\,\hat{k}\,H/2\,\pi} \approx 35$.

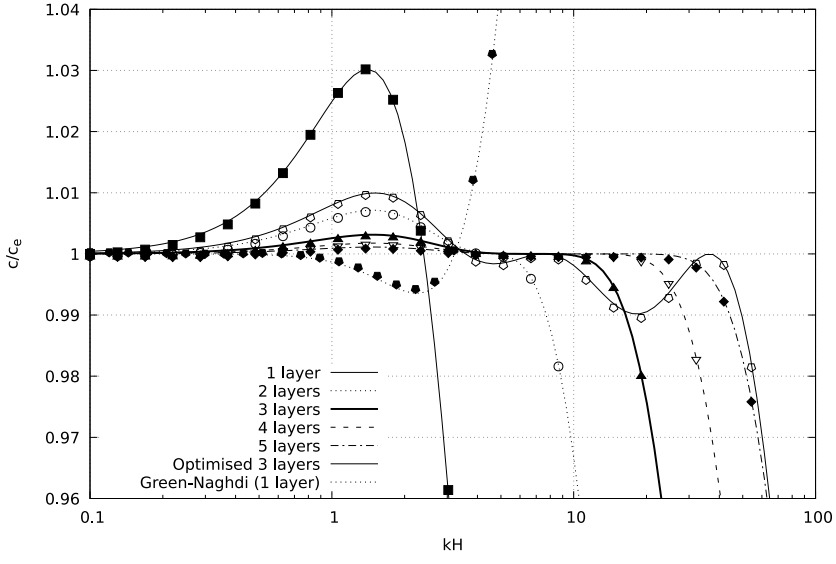


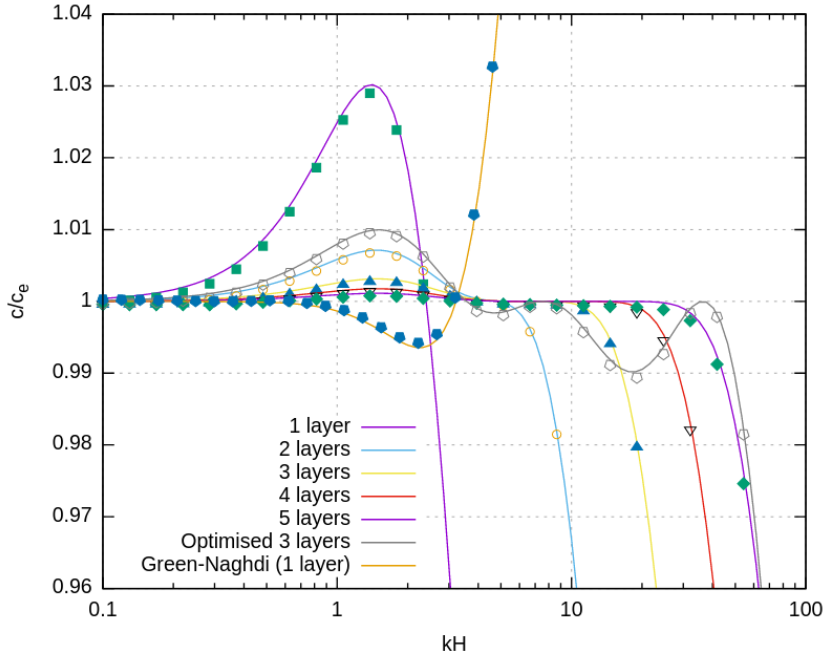
Figure 4. Dispersion relation. The lines are the analytical solutions and the symbols the numerical results (http://basilisk.fr/src/test/dispersion.c).

The same test case is used to estimate the numerical total energy variation per period,

The code runs and generates the (up to date) figures on the website

Note that the standard Green-Naghdi model has a [2,2] Padé approximant dispersion relation (see e.g. Clamond et al. (2017)). The same as Nwogu (1993). Here we used the optimized version of Chazel et al. (2011).

The discrete dispersion relations can be computed using this Maxima script.



Dispersion relation (script)

```
g = 1.
k = 1.

omega1_keller(h_0)=sqrt((4*h_0*g*k**2)/(h_0**2*k**2+4))

omega2_keller(h_0,h_1)=sqrt(((4*h_0*h_1**2+4*h_0**2*h_1)*g*k**4+(16*h_1+16*h_0)*g*k**2)/(h_0**2*h_1**2*k**4+(4*h_1**2+16*h_0*h_1+4*h_0*

omega3_keller(h_0,h_1,h_2)=sqrt((((4*h_0*h_1**2+4*h_0**2*h_1)*h_2**2+4*h_0**2*h_1**2*h_2)*g*k**6+((16*h_1+16*h_0)*h_2**2+(16*h_1**2+64*
```

- Open Science is just Good Science
- Open Science is good **for you** (and your career)
- Free (Scientific) Software is an important part of Open Science in particular because it encourages Reproducible Science (which is just Good Science)
- People are more important than Software $\Rightarrow$ **communication!**
- Many tools are available: wikis, forums, markdown, version control, continuous integration etc. basilisk.fr is an example
- Open Science and Free Software promote **Academic Freedom**
- Institutions ("The State") can help (if they are not too "heavy-handed") even though this clashes with the libertarian ideals of the *hacker culture*

Liberté • Égalité • Fraternité

RÉPUBLIQUE FRANÇAISE