Our software commons

and where we can make it take us

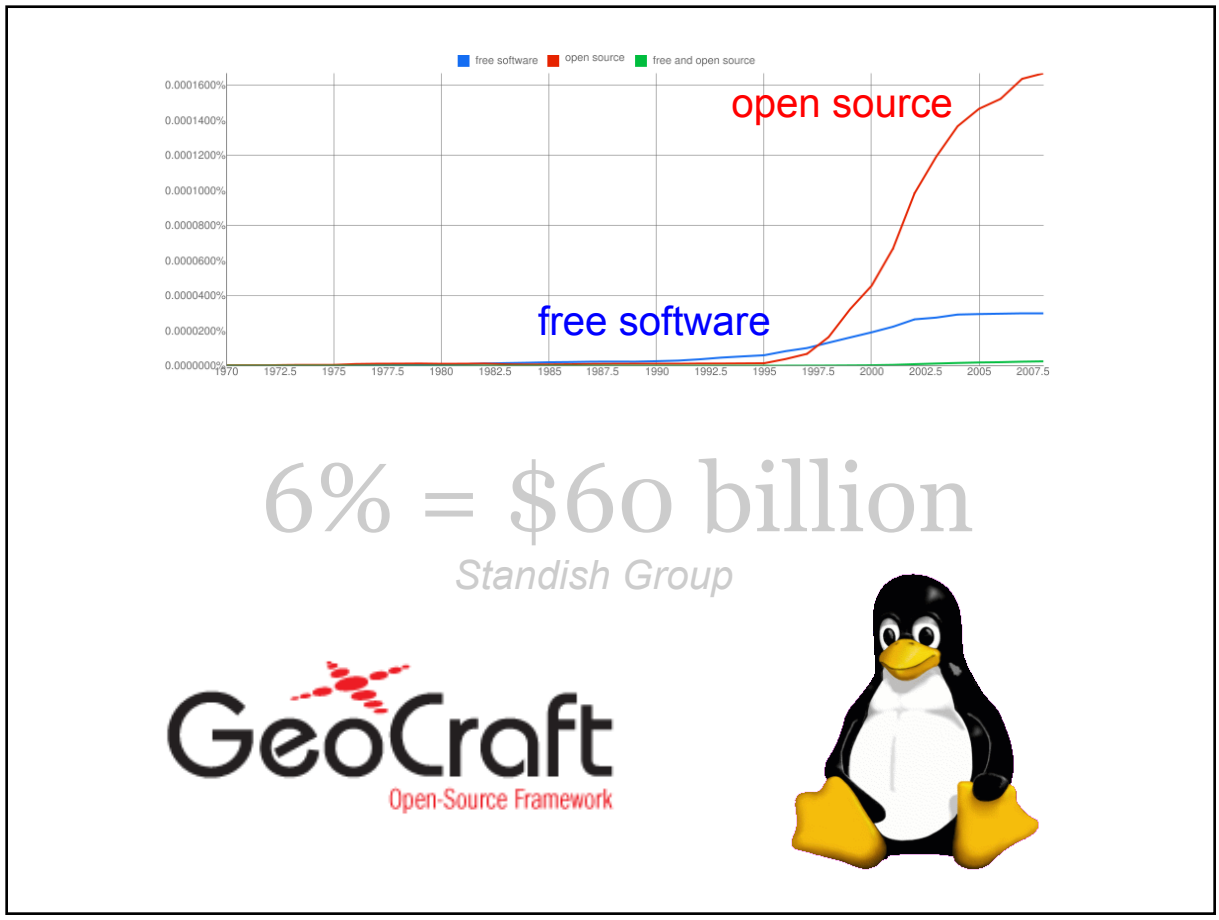
Matt Hall

matt @ agilegeoscience.com

Agile*

Some comments for this online version: This talk was originally given at the Society of Industrial and Applied Mathematics (siam.org) conference Mathematical and Computational Issues in the Geosciences, March 2011, Long Beach, California (SIAM-GS-11). I was invited to give the talk by Berndt Flemisch of the University of Stuttgart. I would not normally be at a SIAM conference, but was very happy to come along and learn.

Format: I have removed most of the builds from this file to simplify it and reduce the number of slides. So some of the slides look harder to explain all at once than they actually are. I have also removed dark backgrounds from several slides, so it's easier to print if you like.



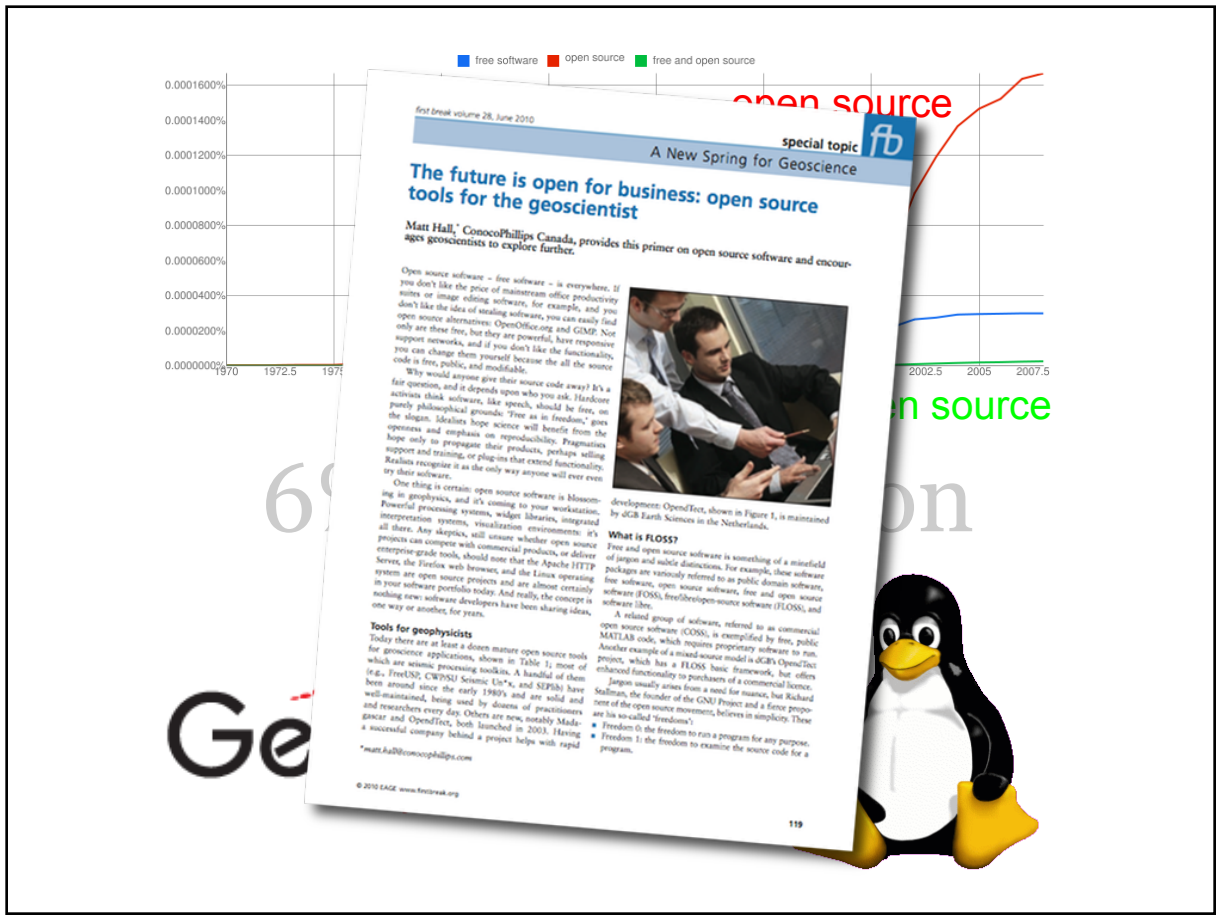
A bit of context.

My own awareness of open source goes back to the early 2000's when I was working at Landmark, a software company. Our software required very expensive hardware, but then the Linux revolution came along. Now the computer that used to cost $1M could cost $25k. Uptake was rapid, for the oil industry.

Then I moved to ConocoPhillips and in 2008 was peripherally involved in the GeoCraft project, an open source geophysical interpretation and prototyping platform. I found this very exciting.

And of course open source is growing in the public imagination, especially as the concept spreads to publishing, ideas, etc. This Google Ngram shows the growth of the terms in published literature.

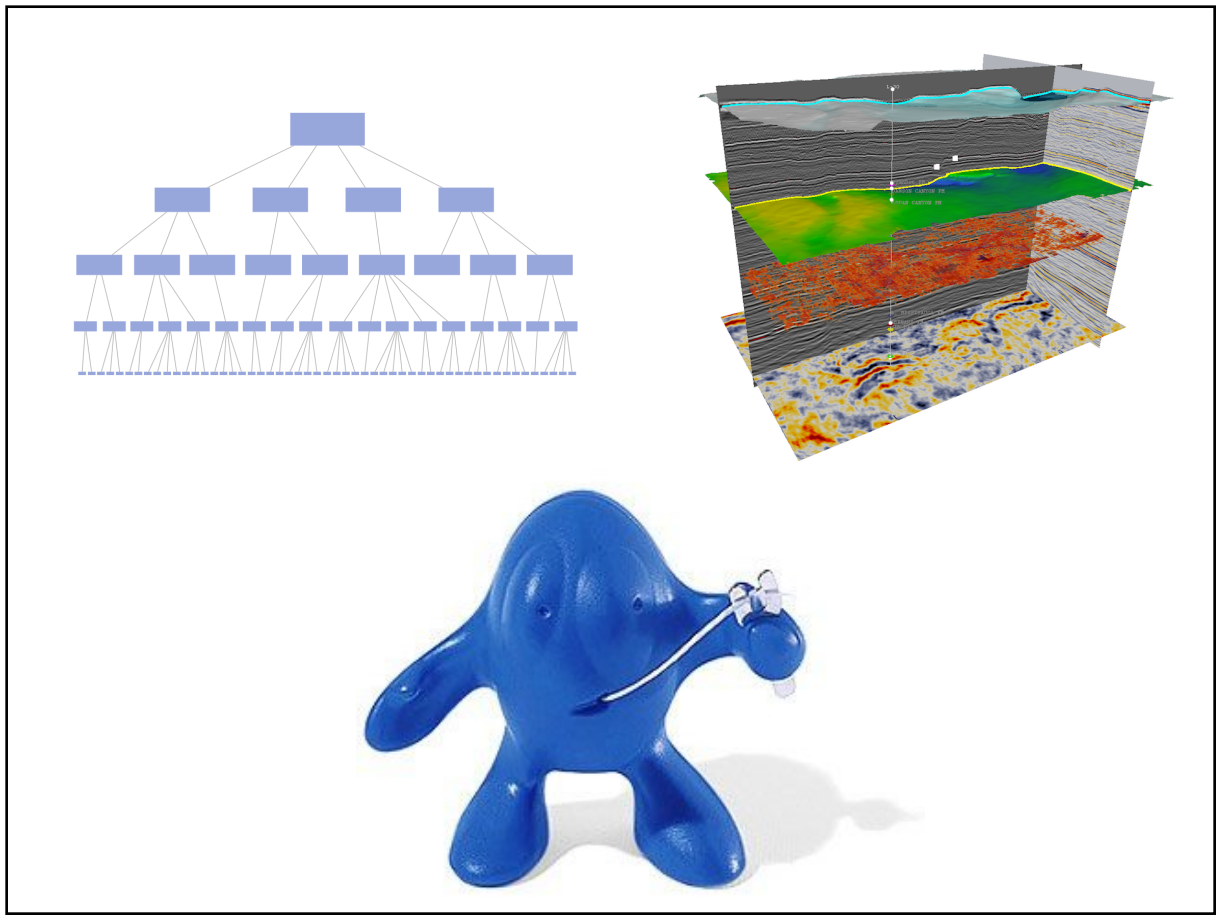
Finally, we have private groups like the Standish Group looking at the open source market as about 6% of the global IT market - but this was in 2008. It is surely growing.



Then last year I wrote this article about free, libre and open source software (FLOSS) in geoscience. It was really just a set of notes from a lot of reading I'd been doing, and a list I'd been compiling of FLOSS projects. (Hall, 2010, EAGE First Break, v28, June).

And no, I didn't choose the cheesy stock photo.

This paper is the reason I'm at this conference.

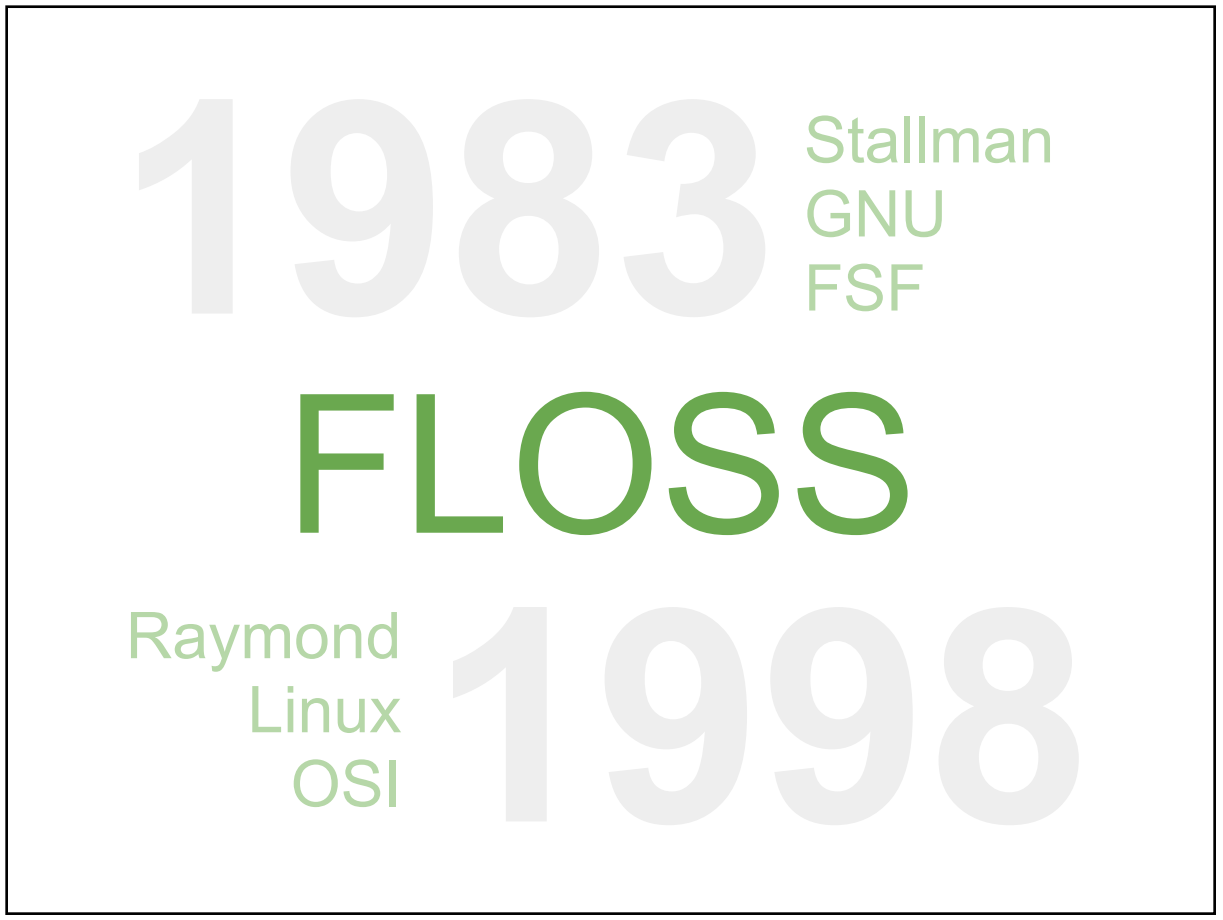


So I want to hit a couple of themes in this primer. The first is simply to introduce you to FLOSS, the history, the jargon, the legalities.

The second is to tell you a bit about FLOSS in geoscience, what it means to me as an interpreter, and where I think it might go over the next few years.

If I have time (I didn't), I have some ideas about how FLOSS might improve our teamwork in organizations and the community at large. We'll see how it goes. (There were too many questions).

Naturally, everything I tell you here is just my opinion, biased, tainted by my experienced, possibly flawed.



Many of you know about the history, perhaps some don't. And since it's important to understand the history to understand the licensing, here goes... in a nutshell...

Early 80's: Richard Stallman, fed up with proprietary software, highly idealistic, political, and driven. Starts re-writing the proprietary and patented UNIX from scratch. Calls it GNU's Not UNIX. Sets up the Free Software Foundation to promote its ideals.

Later, late 90's, along comes the Linux crowd. Linux is the kernel that GNU lacks. GNU-Linux is born. Lots of people react to Stallman's rhetoric and worry about the ambiguity of the word 'free' in 'free software' (which doesn't mean no-cost, but free as in freedom). Eric Raymond et al coin 'open source' instead and set up the Open Source Initiative.

Arguments ensue and go on to this day. I prefer the acronym FLOSS as a compromise position because personally I don't care for the subtleties. It's all the same stuff (mostly).

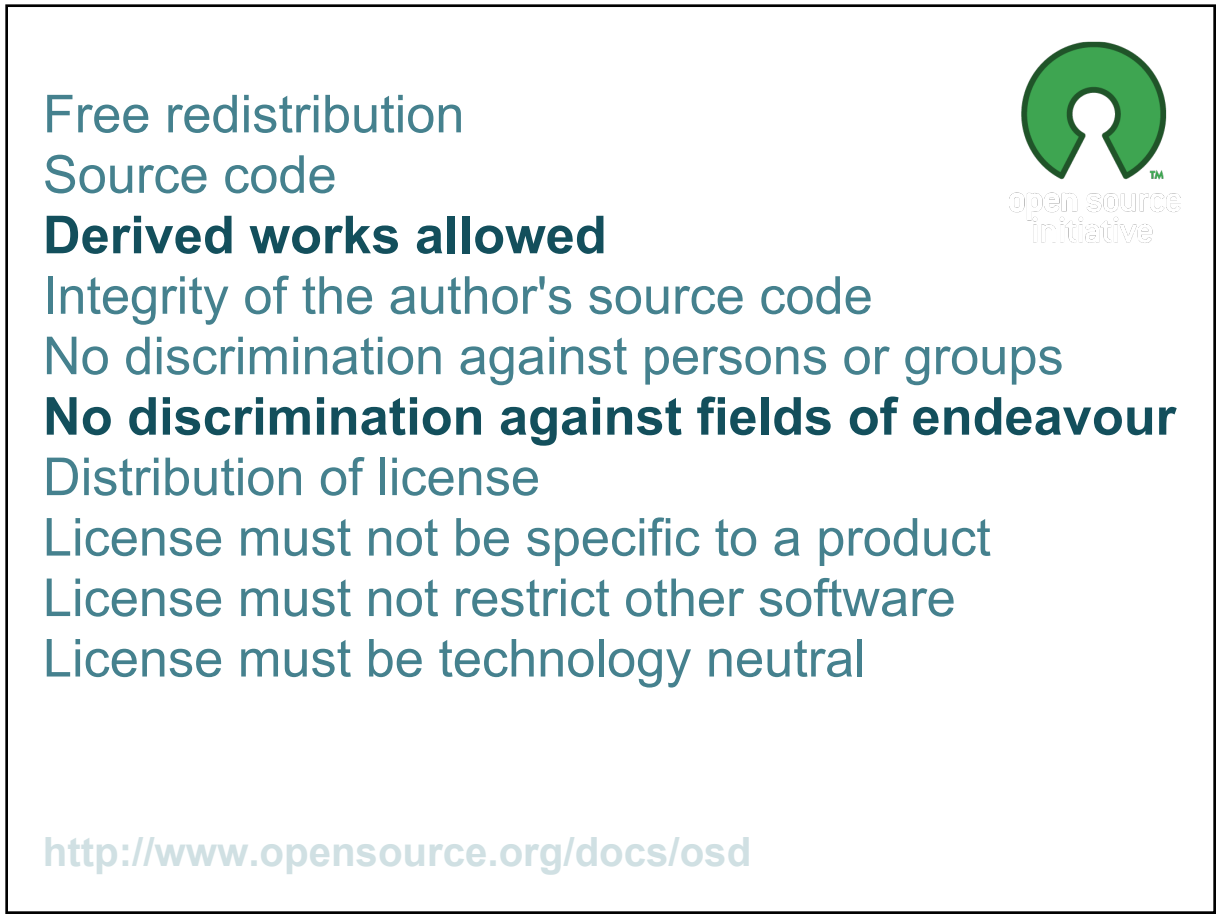
0	use for any purpose	
1	see and change the code	me
2	redistribute the code to others	you
3	distribute the modified code	us

Richard Stallman's four freedoms. This is the heart of the FSF.

You can read them. They set out rights for me to get stuff done, then for me to be able to help you get stuff done, and then for me to help my community.

Sounds good, but the key part comes later, in the license that enshrines these rights. As you'll see, Freedom 3 is not the end of the story: Stallman forces people to pass these rights on to those recipients.

You *will* be free! Or else.



Raymond et al don't like that bit. Their stance: You can be free, but don't force those freedoms on others. You are free not to do that. It's more permissive.

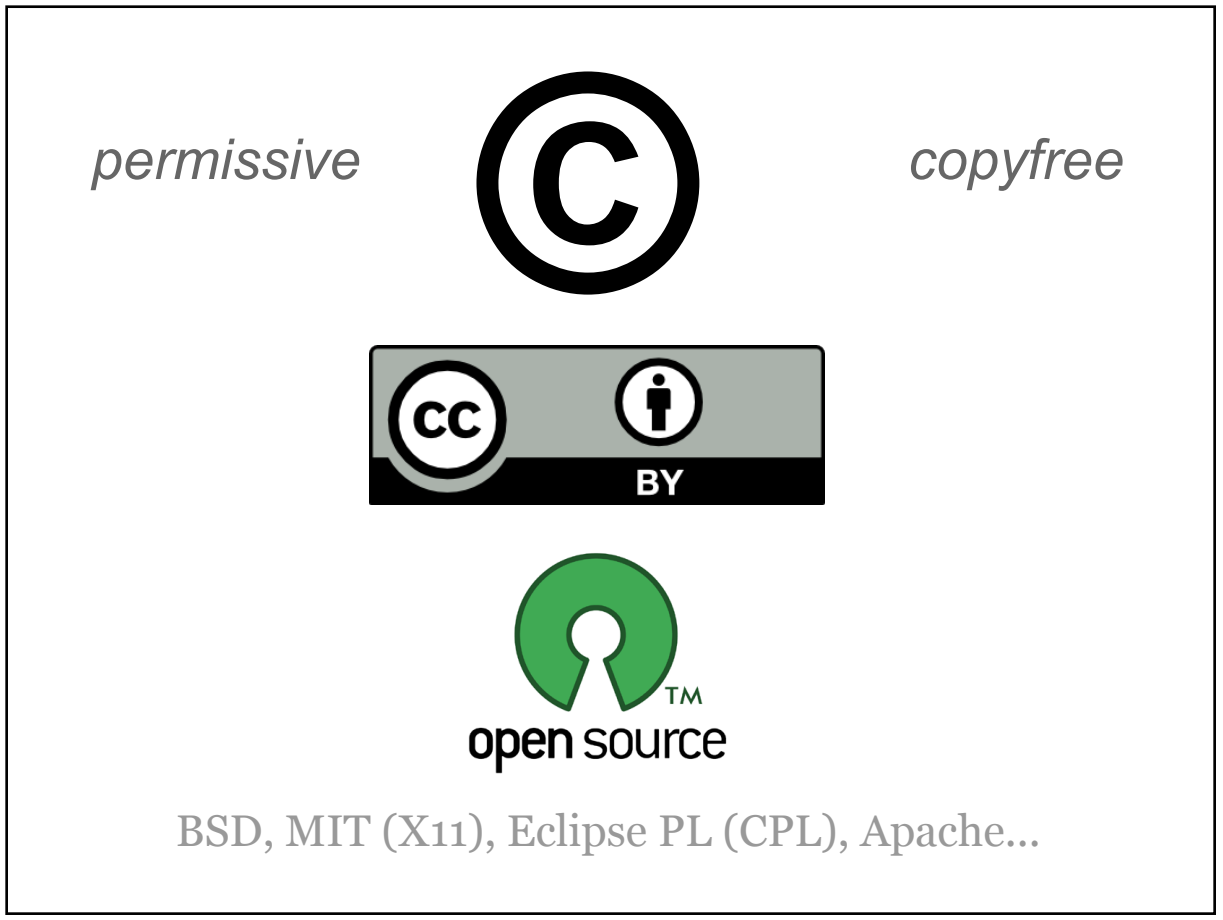
Stallman disagrees that this is *more* free than the FSF's view.

Either way, the OSI have their own definition - the so-called Open Source Definition, or OSD. You can read them, but the key ones are 3 and 6.

3 - Derived works of course are always allowed, but in the OSI's definition, these works do not have to inherit their parent's license.

6 - Free software enshrines this same requirement, but it's noteworthy that many licenses for creative work, such as some flavours of the Creative Commons license, do not (eg CC-*-NC). I think idealists like these licenses, but they are very restrictive, and eliminate many potential readers/users. Just my opinion.

Note that the conditions listed here mean that open MATLAB code, for example, does not count as free or open source.



Before we can talk about licenses, we need to talk about copyright. And copyleft. And copy free. And public domain.

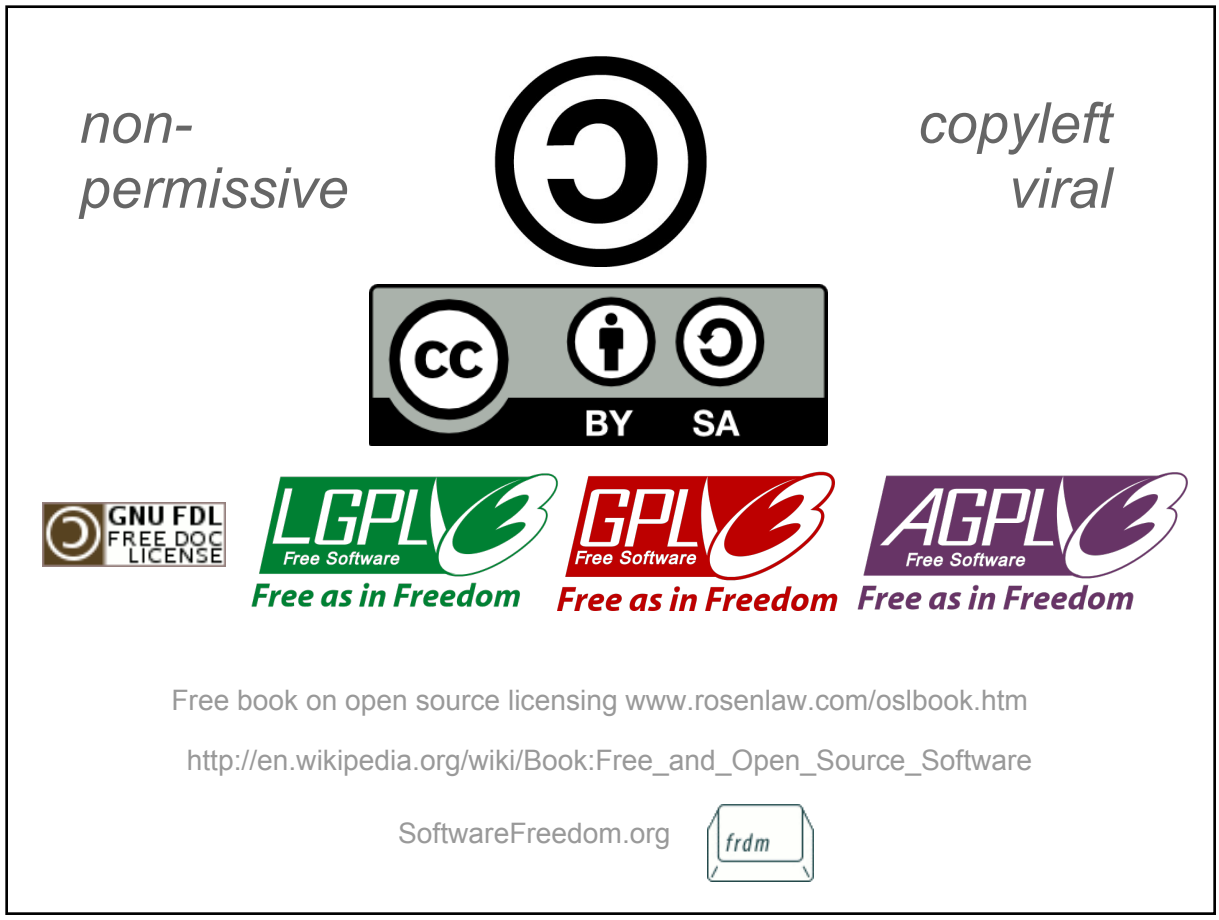
It's a common misconception that free software, open source, Creative Commons, etc, are not copyrighted. This is not the case. Copyright, people perhaps intuitively know, is the set of exclusive rights you have as a creator (or that your employer has!). The right to copy, propagate, perform, whatever.

Licenses simply permit others to do some or all of these things, with or without some restrictions, without your permission.

A simple kind of license is the CC-BY. You can copy my stuff, and use it for anything you like, just give me attribution. Credit where due, basically. I use this license for most of my work.

The OSI certifies licenses that are - more or less - like this. These are some common examples: BSD, MIT, etc.

These licenses are called copyfree. (There is nuance there, but this is not the time or place... if you worry about nuance, talk to a lawyer).



There are various ways to make the license more restrictive. A common way is to insist on the propagating of the right to share This is usually expressed as a moral conviction (though this is disputable), and is the prevalent model, at least in geoscience.

The CC-BY-SA (share alike) license is like this. It's the license Wikipedia uses. Many blogs use it too.

The GNU General Public License is like this, as are all the GNU licenses.

The LGPL is noteworthy because it was created as a kind of compromise: you can cover a library with LGPL so that non-GPL'd software can link to it.

These licenses are sometimes called copyleft, or even 'viral' (usually meant in a derogatory way, IMO).

To find out more about this stuff, here are some resources.

	Geology	Geophysics
GPL v3 Free Software Free as in Freedom	7	21
open source	4	10
Other	3	16

A quick survey of some geology and geophysics software shows that non-permissive licenses are rather more popular, so far anyway, than permissive ones.

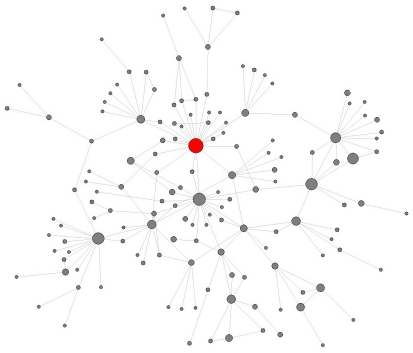
'Other' includes unclear licenses, custom licenses, conditions like 'non-commercial', or no license at all.

No license at all, with a copyright notice, is strange. This is 'open source' but if you don't grant the right to do anything with it, it's doesn't meet the Open Source Definition, and is not free. You should clarify your position for potential users.

No license, without a copyright notice, is de facto 'public domain'. Public domain belongs to everyone, and we can do whatever we like with it, including using it without attribution.

A side note here is trademark. Bayer lost aspirin to generic-dom in the 1940's. Any value they had in the name, the brand, disappeared --- by being copied.

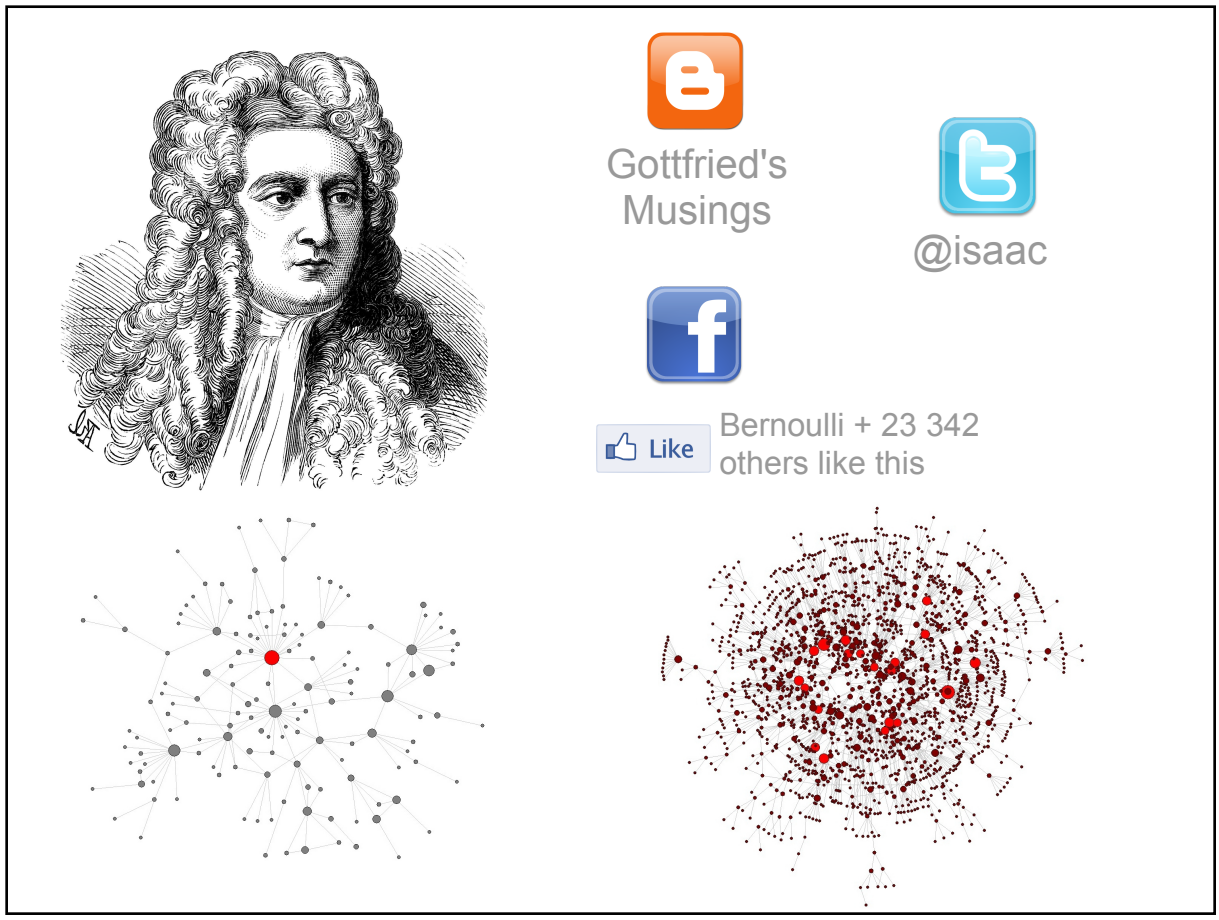
This is kind of relevance if you consider yourself, or at least your work, as a kind of brand. I think it's really all about attribution.



Attribution clear

Partem difficilis,
laboriosus,
inexpeditus,
dubius,
conabilis,

Isaac Newton had no problems with attribution. He lived in a small world network, and everyone knew what everyone else was doing, more or less. The world moved in slow motion by today's standards --- sharing was slow. And difficult, not least because learnd people wrote in Latin... somewhat akin to coding in Fortran. <raucous laughter - I love geeks>



Today of course, sharing is incredibly easy, and incredibly fast. Before long, if we're not careful, attribution --- the originator or creator --- is lost. Our networks are large, and once lost, the ownership may never be re-eastablished.

Attribution is massively important if we want there to be anything other than a free-for-all. Link to original sources, research the originator, name the creator. Treat others the way you'd like to be treated!

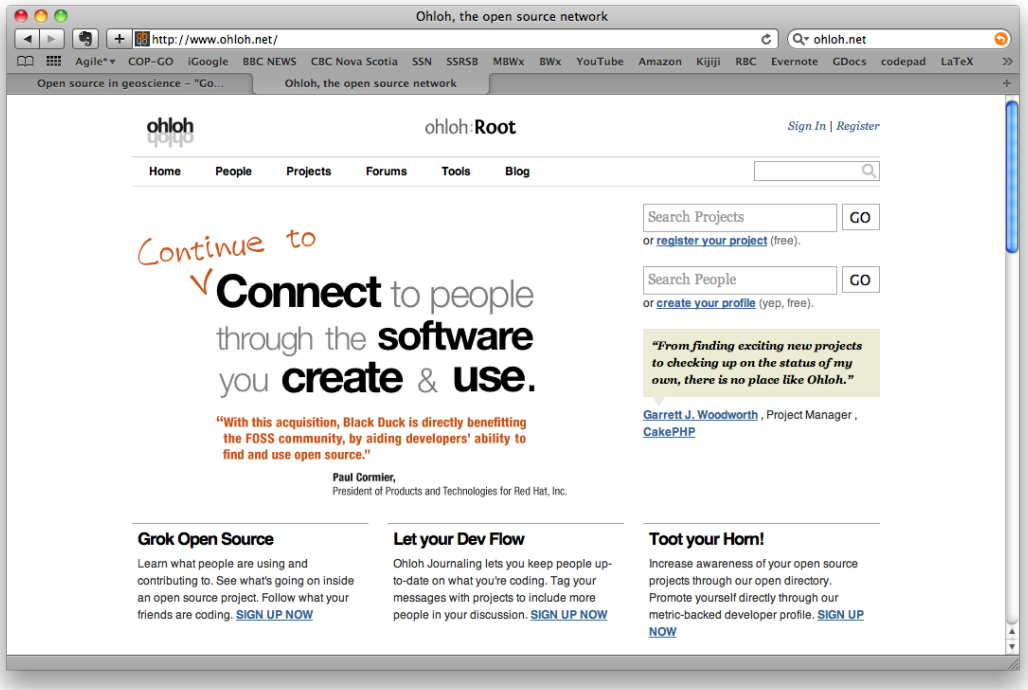
If in doubt over permission or licenses - ask! Most people are happy to say where they got something, or give you permission to use it if it's theirs.

When I was researching the paper I mentioned earlier, I was a little surprised to find that the open source community is just as rigorous and concerned about copyright, trademarks and permissions as the proprietary world. Naively, I thought that since they gave away software (not strictly true - they license it), they wouldn't care. But they don't want to be genericized (is that a word?) or forgotten, so they defend their names and brands.

Be respectful and careful when you use these marks, and even the words they represent. Linux, for example, is a trademark of Linus Torvalds.

It's only by good practice and diligence that we can expect others to have the same concern for our own work.

There - end of rant.

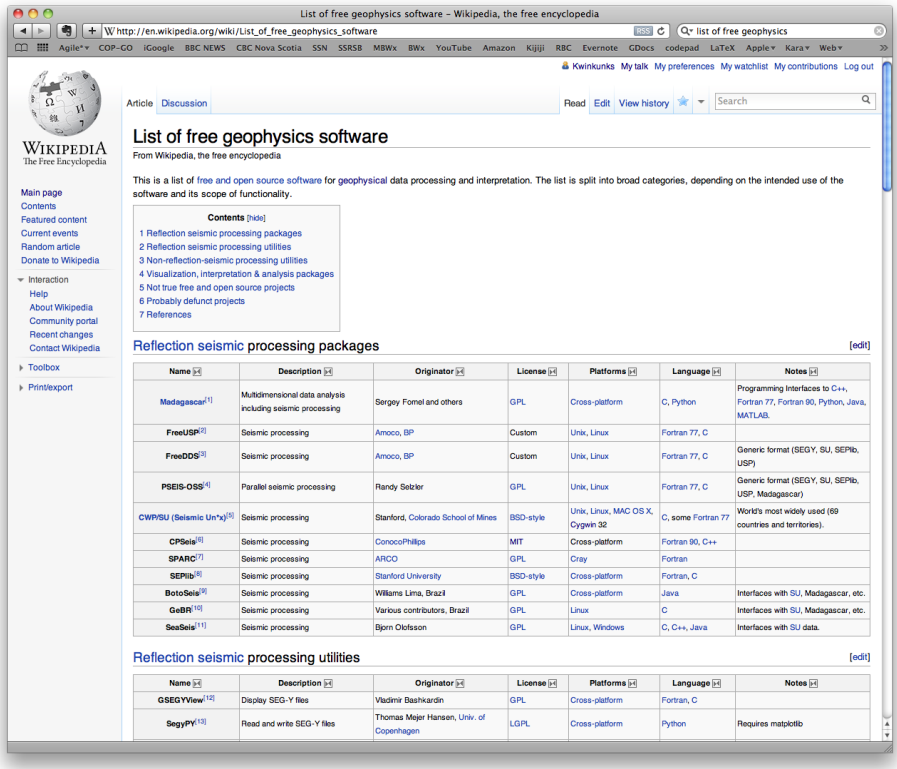


Geophysics 41 Geology 36 Geoscience 12

Ohloh.net is a great place to go to look for and learn about FLOSS. you can search for projects, register your own project, and find out all sorts of data about a project, like what license it uses, what languages it is coded in, how often it is updated, etc.

To me, this is an example of the kind of meta-infor and richness that will drive the community forward, encouraging competition and collaboration. The closed-source world cannot compete with this.

As you see, according to Ohloh, there are about a hundred or so registered geo-FLOSS projects.



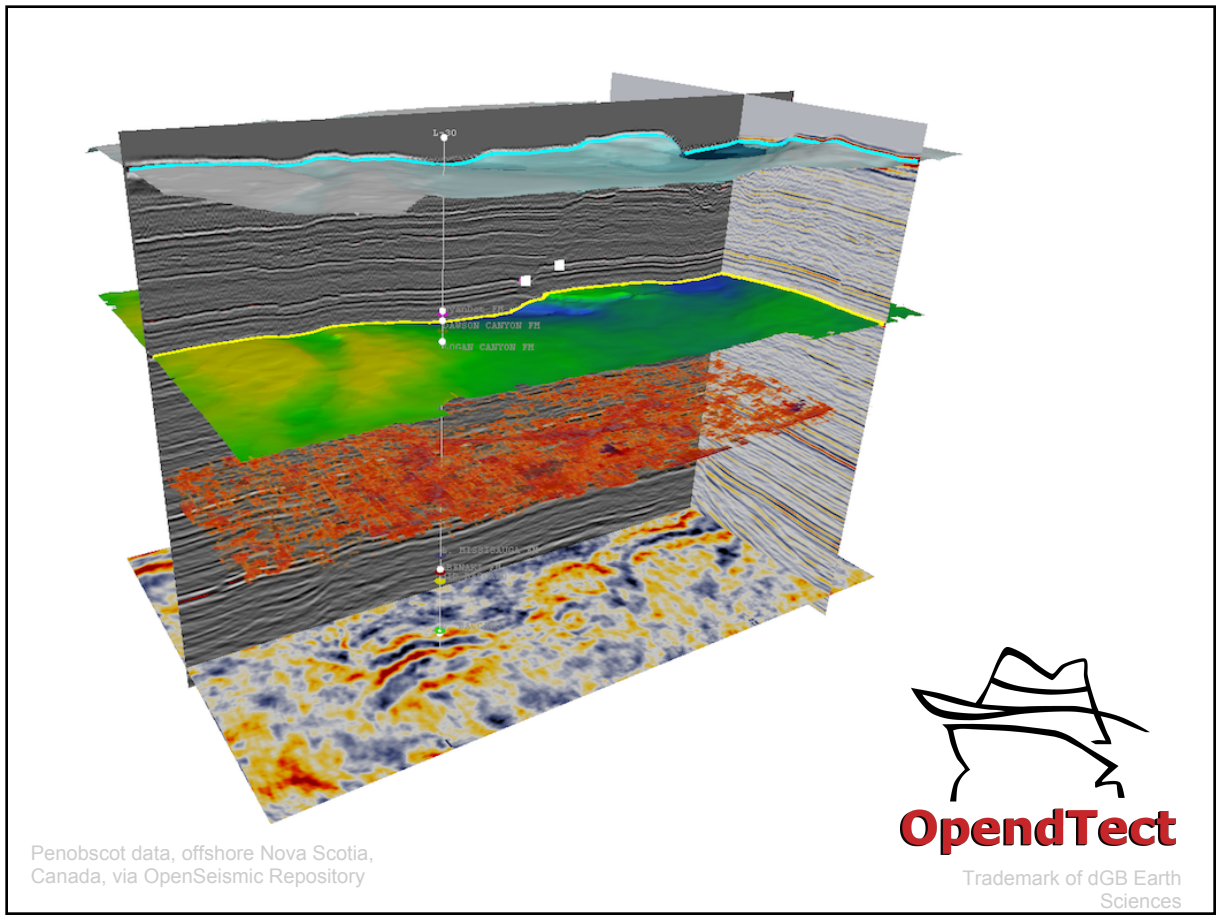
Geophysics 34 Geology 11

There is also a pair of curated lists on Wikipedia, one for each of geology and geophysics.

Same sort of info: license, language, platform, etc. And links to the projects.

If your project is not there, or if you find mistakes there, please edit the pages (it's a wiki!).

So by this count, arguably more accurate than Ohloh's search reckoning, there are at least 45 projects... though a few days at this conference have shown me that there are certainly some projects missing.

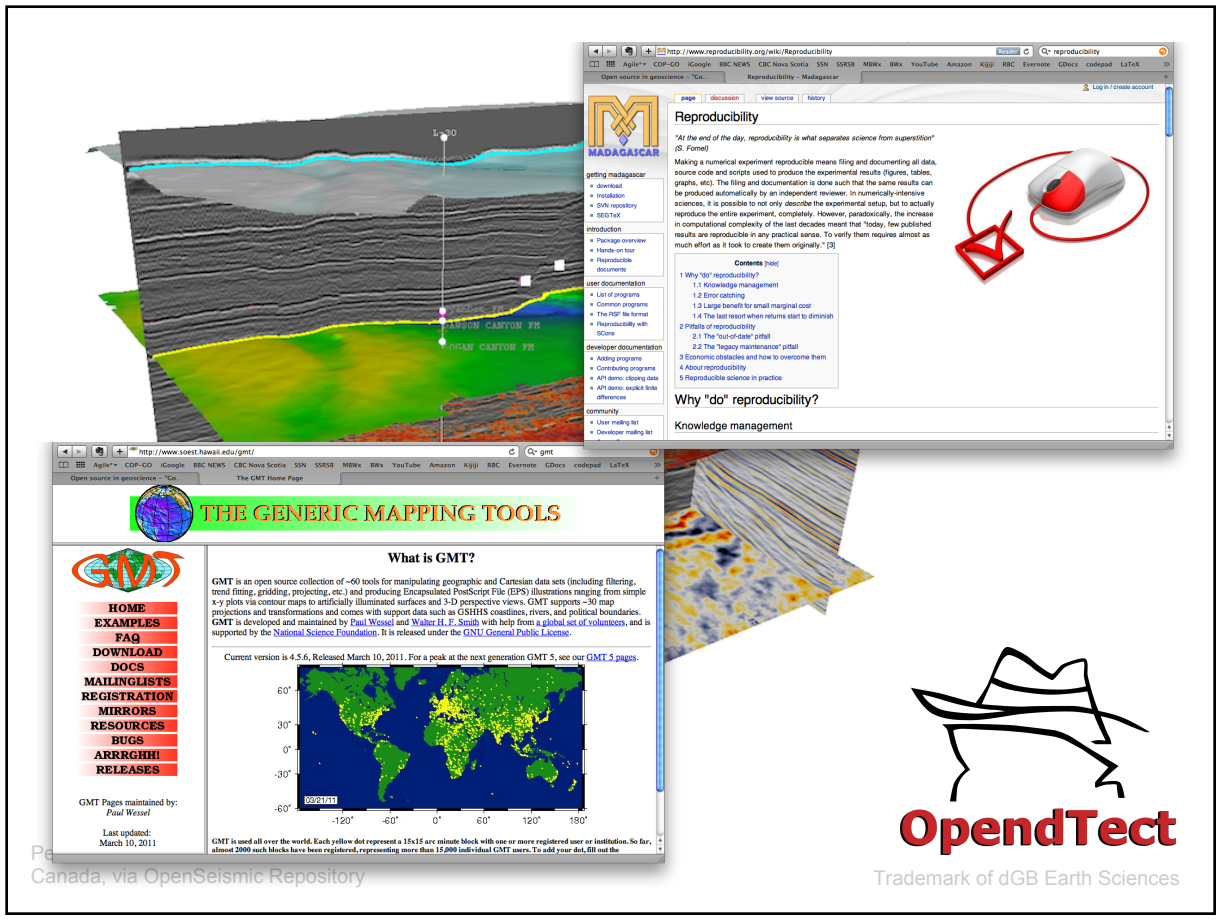


So what's the real story for individuals - for geologists and geophysicists? The story is - there is a lot of great, functional software out there.

This tool is developed by paid professional coders in the Netherlands. It is full-featured, documented, and licensed under the GPL.

Tools like this one are commonly sold for anything between $10k and $100k. As Kevin Kelly likes to say - even if your stuff is not free (of charge), you are competing with free.

I use this tool in my consulting practice.



The really wonderful thing about it is perhaps its place in a new ecosystems of open tools. It integrates seamlessly with Madagascar, Sergey Fomel's wonderful seismic processing suite (and reproducibility flagship). Also with GMT, a full-featured mapping suite from the University of Hawaii. Don't let its 1994-throwback website fool you - GMT competes easily with proprietary tools.

This sort of integration is possible because all of these platforms are open and free.

Integrating proprietary software is very hard, as anyone working in an oil company as a subsurface scientist can attest. It usually involves awk - nuff said.

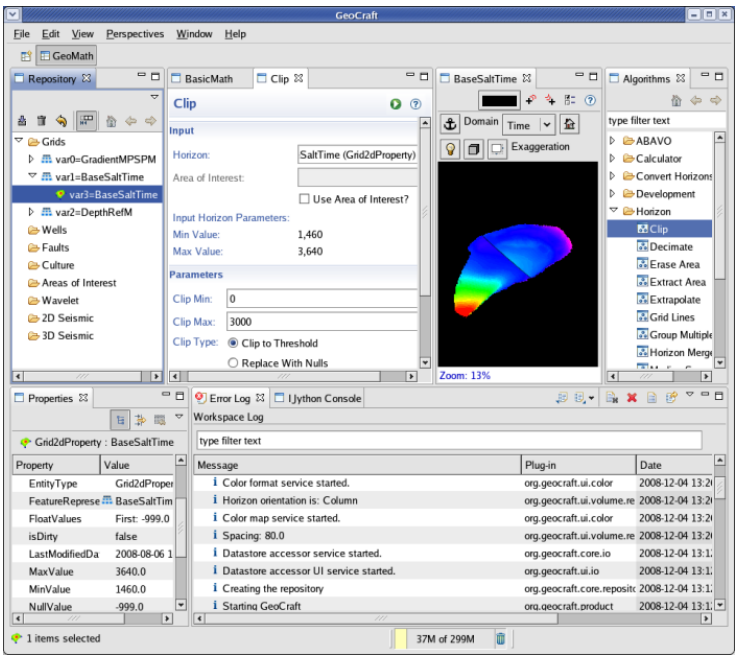
Yes, awk is Free Software.

ConocoPhillips *open source projects*

CPSeis™

JavaSeis

GeoCraft
Open-Source Framework

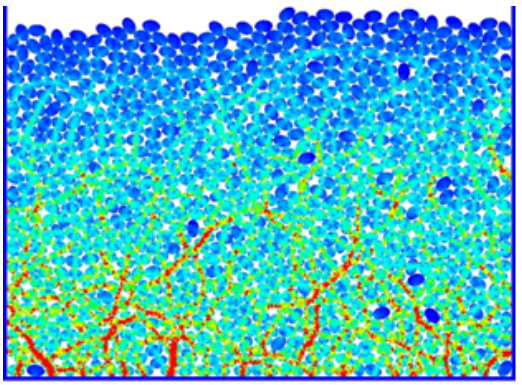


As I mentioned, another corporate supporter of open source --- though sometimes you might not know it --- is ConocoPhillips, with three open source projects: CPSeis.org, JavaSeis.org and GeoCraft.org. These all have permissive licenses.

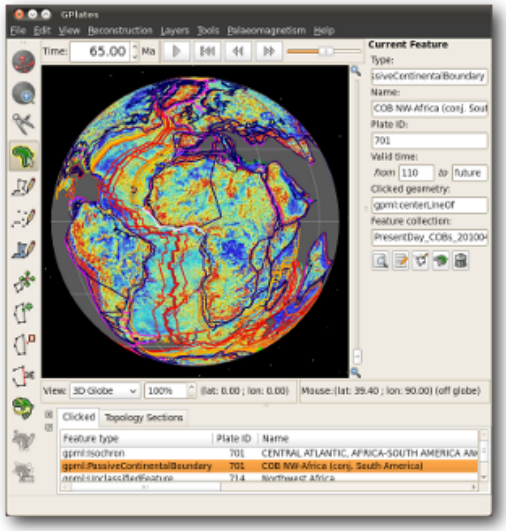
The last of these is the most interesting to me --- it is a framework for quantitative exploration geophysics. Complete with a Python command line (iJython). This is the kind of tool that can transform an ordinary interpreter into an agile innovator.

For some reason, ConocoPhillips rarely if ever mentions these projects in public, perhaps with the exception of JavaSeis.

[Statoil is another oil company with an appreciation for FLOSS, at least as a research tool.]



Virtual Geoscience Workbench

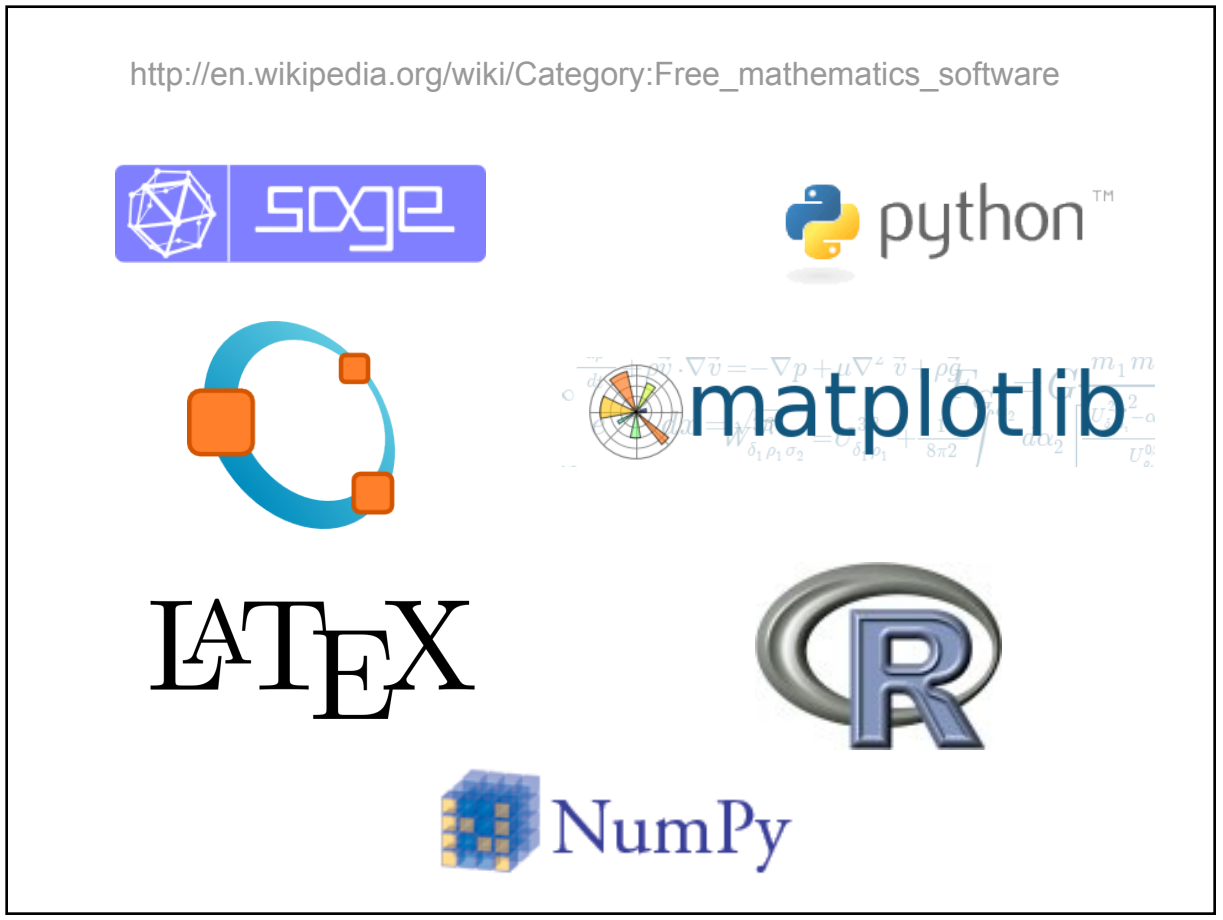


http://www.gplates.org/

There are lots of other tools out there - a couple of examples here, both new additions to those Wikipedia pages I mentioned.

One is a discrete-finite element solver, the other a virtual globe for plate tectonic reconstructions.

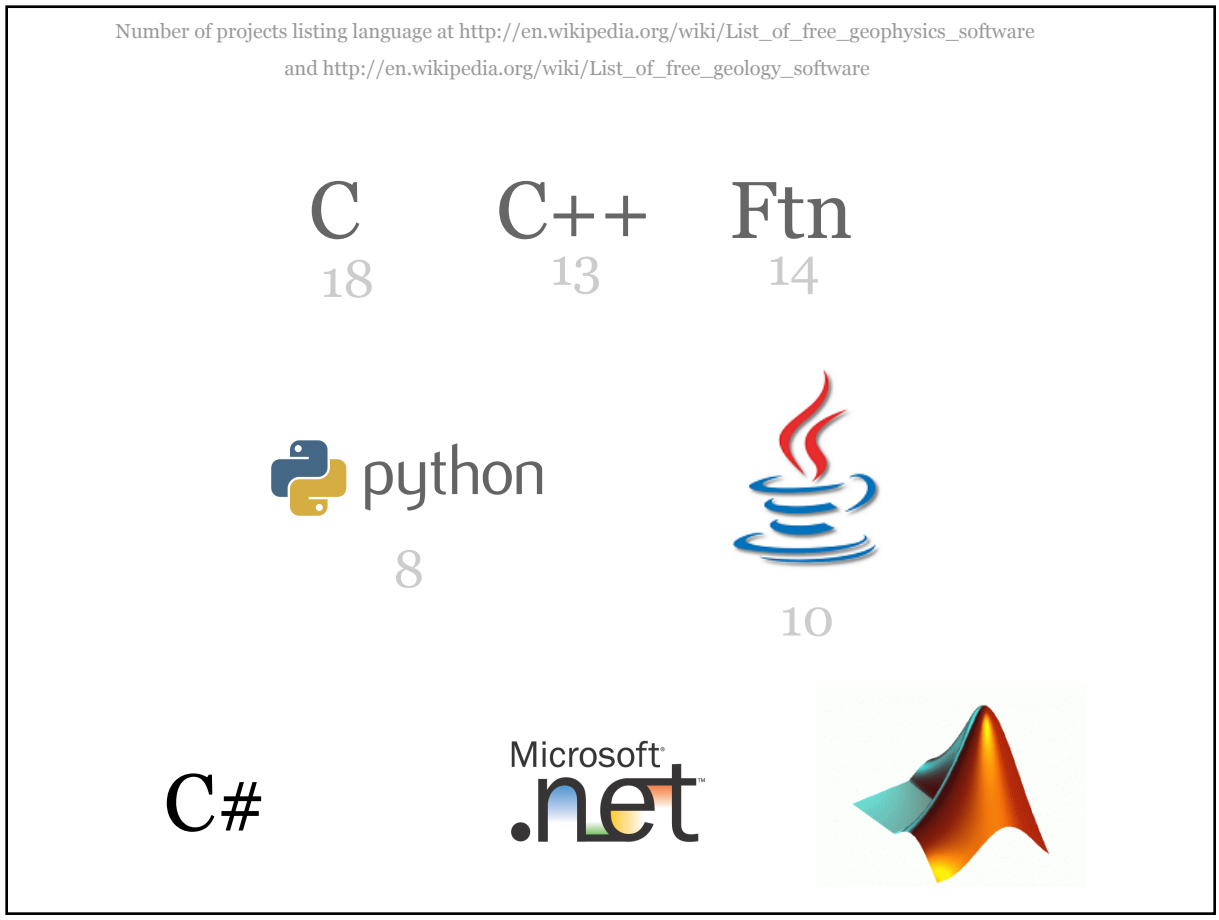
Tremendous variety.



Since I was coming to a math conference, I thought I should at least mention some of the many FLOSS tools for math.

They range from Maple-like solvers (SAGE) to MATLAB-like programming environments (GNU Octave), to programming languages (Python).

Link to list.

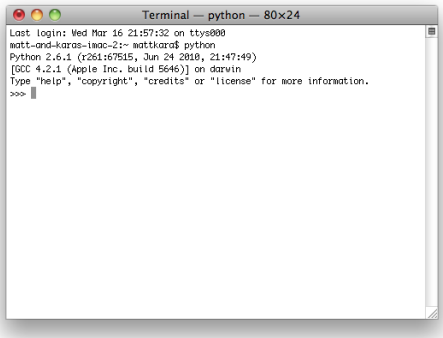


Python is very interesting to me because I have a slim chance of reading and maybe even writing it. It is also 100% Free and Open.

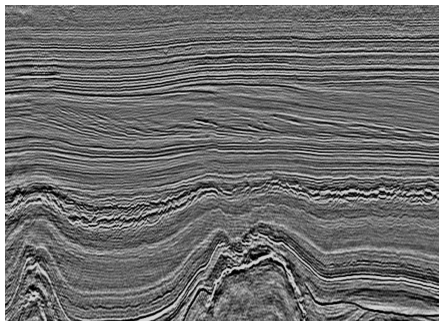
These numbers show how many projects are using each language, again from those Wikipedia pages. As you see, the 'traditional' languages are at the top: C, C++. There's quite a bit of legacy Fortran, but I don't know that a lot of people are still coding software for release with it. I could be wrong. I hope I'm not.

C#, .NET, MATLAB etc are not free and open and therefore not really part of the discussion... porting MATLAB code to GNU Octave is fairly easy, however.

What's so great about Python anyway?

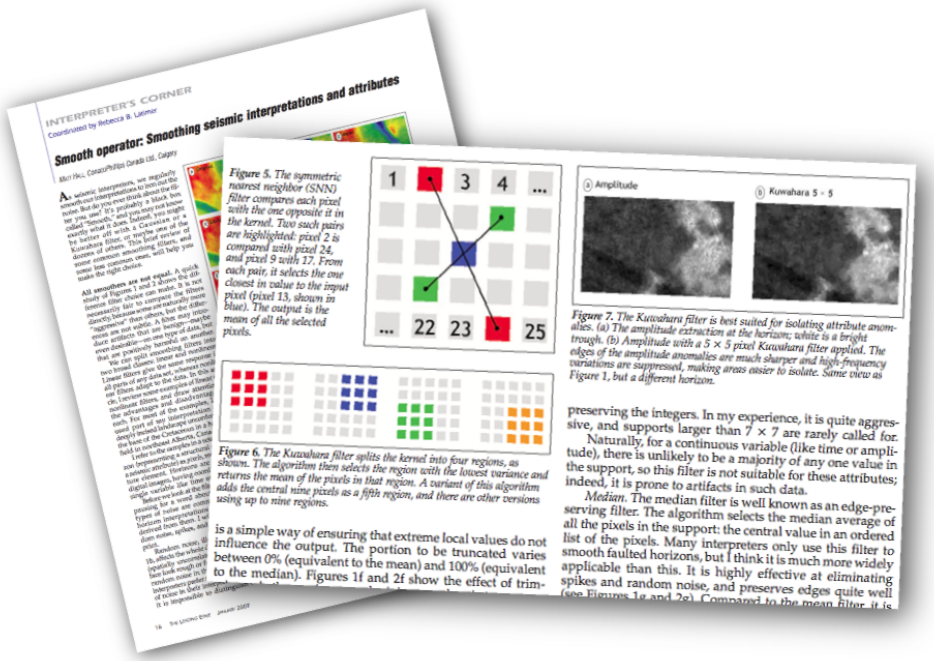



```
import segypy
data, header, traceheader = segypy.readSegy ('mydata.sgy')
segpy.wiggle(data,header)
```



SegyPY - Thomas Mejer Hansen

Well, this is rather superficial of me perhaps, but look at this command to read and draw some seismic data. Now - I am using a library, but finding and installing that library was easy... (cont'd)


INTERPRETER'S CORNER

Coordinated by Rebecca B. Latimer

Smooth operator: Smoothing seismic interpretations and attributes

Figure 5. The symmetric nearest neighbor (SNN) filter compares each pixel with the one opposite it in the kernel. Two such pairs are highlighted: pixel 2 is compared with pixel 24, and pixel 9 with 17. From each pair, it selects the one closest in value to the input pixel (pixel 13, shown in blue). The output is the mean of all the selected pixels.

Figure 6. The Kuwahara filter splits the kernel into four regions, as shown. The algorithm then selects the region with the lowest variance and returns the mean of the pixels in that region. A variant of this algorithm adds the central nine pixels as a fifth region, and there are other versions using up to nine regions.

Figure 7. The Kuwahara filter is best suited for isolating attribute anomalies. (a) The amplitude extraction at the horizon; white is a bright trough. (b) Amplitude with a 5 × 5 pixel Kuwahara filter applied. The edges of the amplitude anomalies are much sharper and high-frequency variations are suppressed, making areas easier to isolate. Same view as Figure 1, but a different horizon.

...so the idea is that now I can read an article in Geophysics, or The Leading Edge, and --- assuming the authors reveal their methods openly (!) --- I can try them out immediately on my own data. I can improvise, tweak and improve.

This is powerful: Now I can test ideas on the fly and be free from GUIs and other people's ideas of what I want to do with my data.

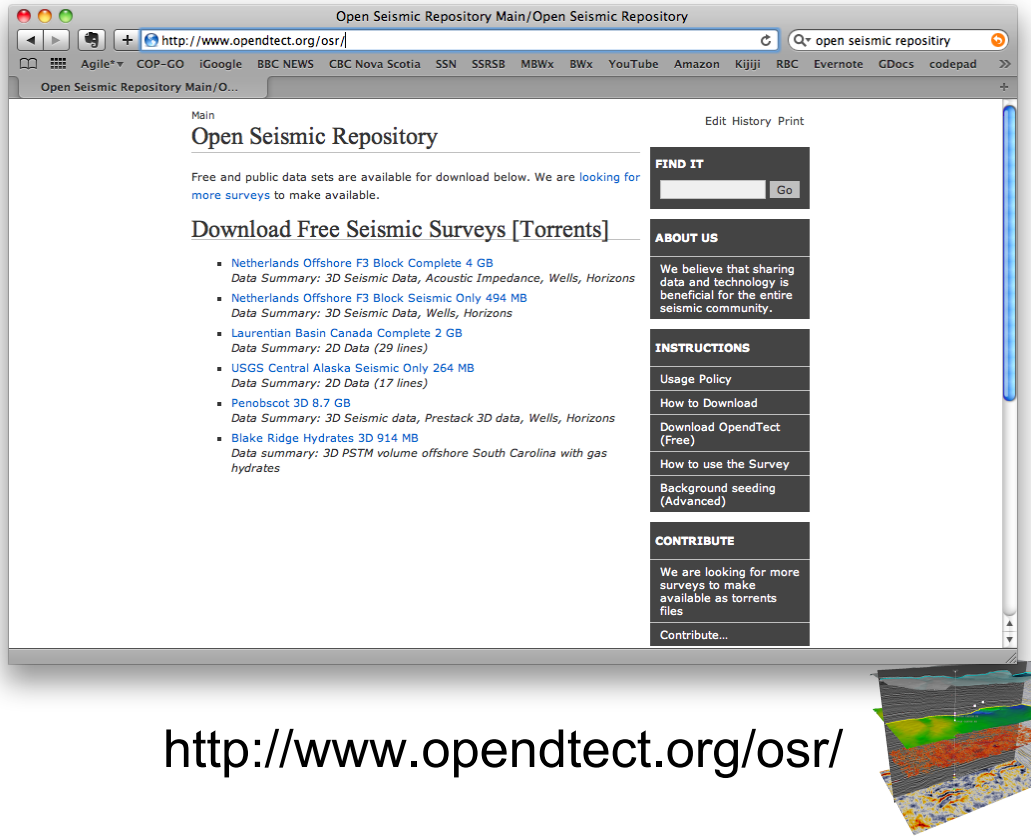
I am free! Thank you Python. One day, maybe...

Just as Wordpress (free software, by the way) has allowed anyone at all to post news, views, and real, rich, meaningful content on the Internet (lots of alternatives, too, of course)...

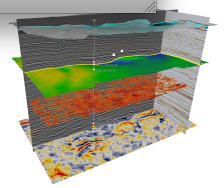
And just as perhaps Android App Inventor (not itself free, but interfacing free software) may allow the easy creation of mobile applications by non-programmers...

Perhaps one day a rich enough ecosystem of geoscience FLOSS can be exposed in an API-like way --- or, better, an App Store like way --- and code, algorithms, ideas, can be re-mixed and tweaked to solve new problems.

These open source hosting platforms - lots of examples - are perhaps the start of this exposure... but it has a way to go before it is as painless as, say, finding music in iTunes.



http://www.opendtect.org/osr/

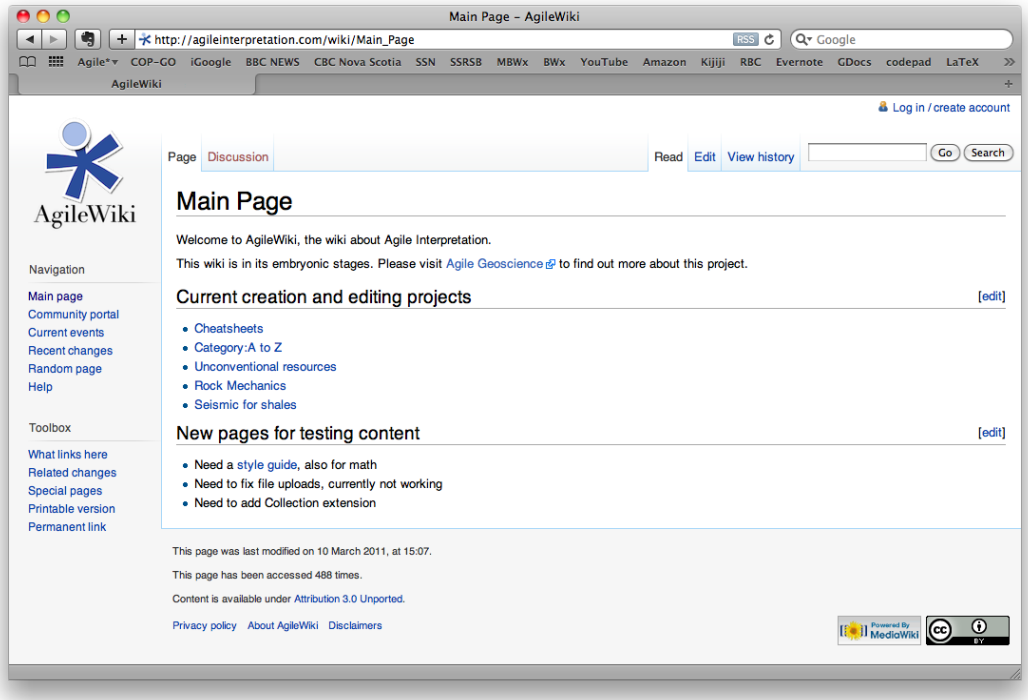


The software side of the equation is just one piece... To really move forward, I think we will require open data. One day, I believe this will be mandated in oil and gas --- industry will need every advantage as our energy equation gets out of balance. I think society will demand this.

Open data is also a fundamental requirement for Reproducibility.

And there are a few outposts of innovation here. The Virtual Seismic Atlas (Leeds) was a start, but it's just images, not data.

dGB (of OpendTect fame) have taken the biggest and boldest step I've seen: the Open Seismic Repository. It's not iTunes-easy, but it's getting there. Plug-and-play into OpendTect. Brilliant.



In my own humble endeavours, we are soon to launch AgileInterpretation, a wiki for subsurface science in oil and gas. Pretty narrow remit, perhaps, but you have to start somewhere. We hope to announce this next month, though it is live today and you are welcome to kick the tyres (prepare to be underwhelmed though; we are still building some skeletal content to launch with). Agileinterpretation.com --- watch this space.

> Don't worry about people stealing an idea. If it's original, you will have to ram it down their throats.
>
> **Howard Aiken**
> US computer scientist, 1900–73

> It may be hard to monetize fame, but it is impossible to monetize obscurity.
>
> **Cory Doctorow**
> Canadian writer & activist, b.1971

> Stop keeping your ideas secret.
>
> **Seth Godin**
> Marketer, b. 1960

Where's all this going? I have no idea. The beauty is its very organic nature, its non-linear, unplanned, grass-roots development.

There will always be naysayers --- people worry about losing out to openness. They see their advantage as their output --- their ideas. This is totally wrong. Your advantage is other people knowing about and using your ideas.

No one will steal anything --- you will have to fight to be heard.

I ove this quote (Doctorow's). This best-selling author does very nicely in bookstores, even though he gives away all his work on his website. He knows that he just needs readers --- recognition (and money) will follow from that.

So: share, publish, and be Free!