Reproducible Science and Modern Scientific Software Development

13th eVITA Winter School in eScience sponsored by Norges forskningsråd
Dr. Holms Hotel, Geilo, Norway
January 20-25, 2013

Overview of Reproducible Science

Dr. André R. Brodtkorb,
Research Scientist
SINTEF ICT, Dept. of Appl. Math.

Outline

- What is reproducible science and why should I care?
- What does software development have to do with it?

Reproducible Research

Who am I, and why am I speaking about this?

- Ph.D. from the University of Oslo (2010)
  - Have been working with hyperbolic conservation laws on GPUs, shallow water in particular
    - Getting it 90% correct takes "two weeks"
    - Getting it 99% correct takes "two years"
    - Getting it 100% correct is "impossible"
  - Three month stay at the National Center for Computational Hydroscience and Engineering

- Research Scientist at SINTEF ICT since 2010
  - Getting a prototype code to a commercial code is a major challenge!

- Attended ICERM (Brown University) workshop on reproducible research December 2012
  - Learned a lot from the real experts

Acknowledgements

- A large portion of the slides I am presenting are inspired by the ICERM workshop Organizers: David H. Bailey, Jon Borwein, Randall J. Leveque, Bill Rider, William Stein, Victoria Stodden

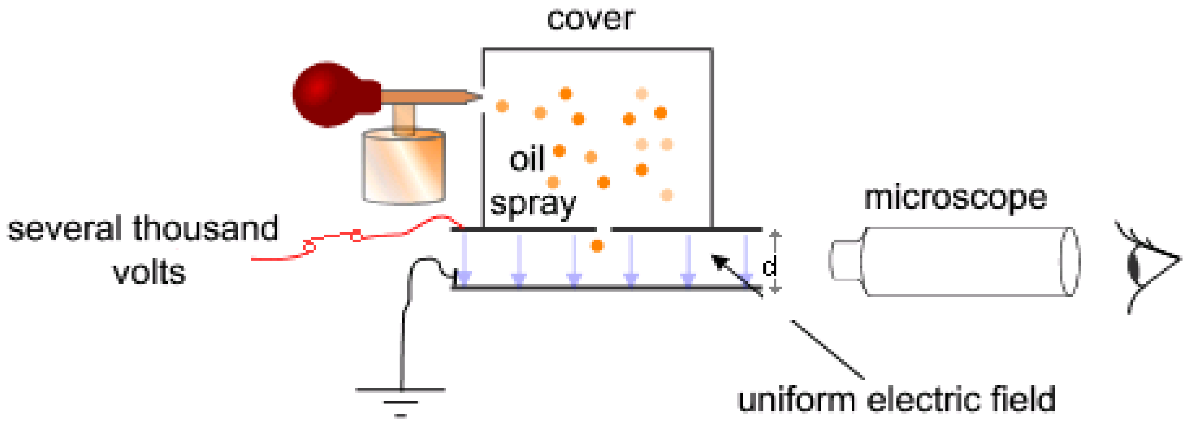
What is reproducible science?

- Science is a very wide spectrum of disciplines:
  - Chemistry
  - Physics
  - Biology
  - Mathematics
  - Computer science
  - Medicine
  - …

- Reproducible research for is most certainly not the same in any two given disciplines!

Measuring the charge of an electron

- Robert Millikan held a famous experiment published in 1910.
  - Part of the reason for his 1923 Nobel prize in physics



- By varying the charge of the electric field, the (charged) oil particles would rise or fall.
  - Millikan discovered that that charge was discrete, and had a value of $1.5924(17)\times10^{-19}$ C with a very small margin of error
  - The value is today believed to be $1.602176487(40)\times10^{-19}$ C

Measuring the charge of an electron

- In 1978 Gerald Holton criticized Millikan, claiming he had manipulated the data
  - Millikan had 175 measurements taken over five months
  - 75 measurements taken over two months published

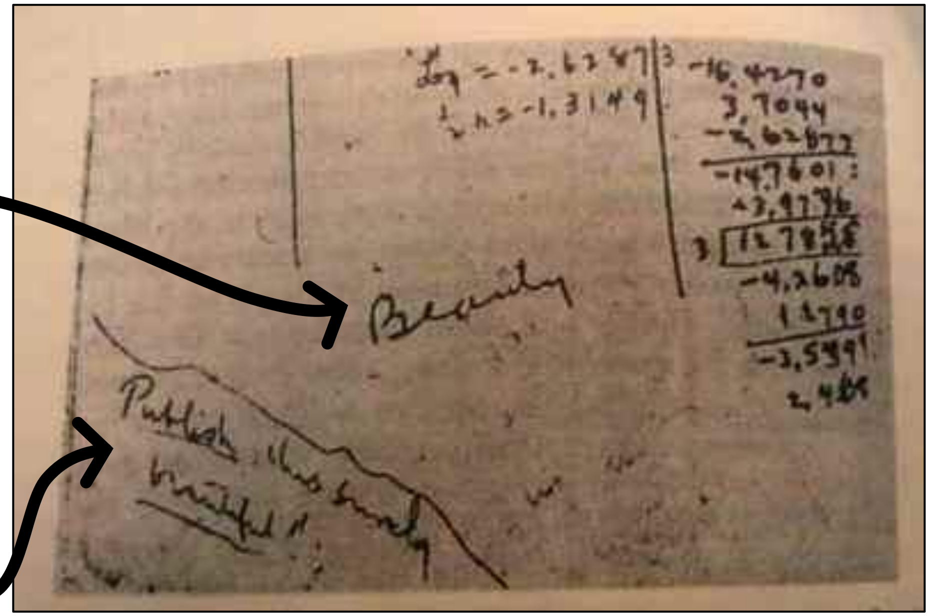
> "It is to be remarked, too, that this is not a selected group of drops, but represents all the drops experimented upon during 60 consecutive days"
> --Millikan, 1913

- Millikan kept a journal of his experiments, which includes notes on the measurements:
  - "Very low, something wrong", "Publish this beautiful one", "Error high will not use", "Too high by 1 ½ %", . . .

Measuring the charge of an electron

Beauty

Publish this surely Beautiful!!



Measuring the charge of an electron

- Millikan had almost exactly the correct value, but extremely small error margins in published result.

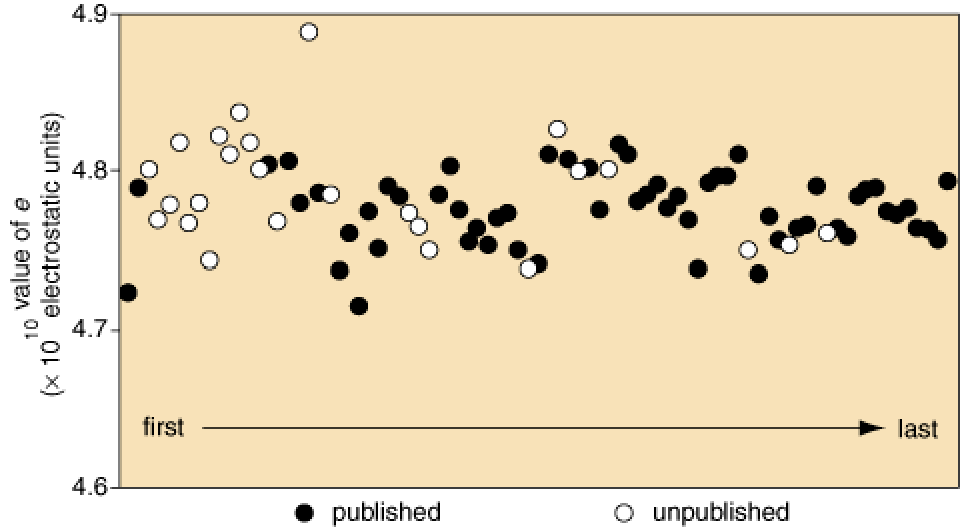


Figure by David Schneider

- A lot of data excluded from publication

Measuring the charge of an electron

We have learned a lot from experience about how to handle some of the ways we fool ourselves. One example: **Millikan measured the charge on an electron** by an experiment with falling oil drops**, and got an answer which we now know not to be quite right**. It's a little bit off because he had the incorrect value for the viscosity of air. **It's interesting to look at the history of measurements of the charge of an electron, after Millikan. If you plot them as a function of time, you find that one is a little bit bigger than Millikan's, and the next one's a little bit bigger than that, and the next one's a little bit bigger than that, until finally they settle down to a number which is higher.**

...

Measuring the charge of an electron

…
Why didn't they discover the new number was higher right away? It's a thing that scientists are ashamed of - this history - because it's apparent that people did things like this: **When they got a number that was too high above Millikan's, they thought something must be wrong - and they would look for and find a reason why something might be wrong. When they got a number close to Millikan's value they didn't look so hard. And so they eliminated the numbers that were too far off, and did other things like that…**

--Richard Feynman

Making up data

- http://dilbert.com/strips/comic/2006-11-11/

Making up data

- The Sudbø case (2006)
  - Made up 900 persons with metical history for cancer research (the use of anti-inflamatory drugs were claimed to reduce risk of mouth cancer).
  - The data was supposedly from a named patient database (which had not yet opened…)
  - Articles retracted (including in The Lancet), his wife and brother were co-authors on several retracted papers, lost his Ph.D., …

- Lancet editor: the biggest scientific fraud conducted by a single researcher

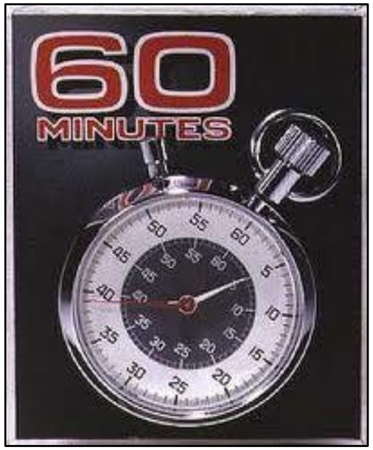
Making up data

- http://dilbert.com/strips/comic/2008-05-08/

Duke University: Breakthrough in cancer research gone bad

"Anil Potti is accused of **falsifying data** regarding the use of microarray genetic analysis for personalized cancer treatment, which was published in various prestigious scientific journals."
[Wikipedia on Anil Potti]

- "As of February 2012, of more than 120 peer-reviewed publication "Published Papers", ten scientific papers authored by Potti and others retracted"
- Patients treated with experimental treatments (personalized treatment)
  - Based on their genes, one would find the most suitable drug for cancer treatment
- 60 minutes documentary: http://www.cbsnews.com/video/watch/?id=7398476n



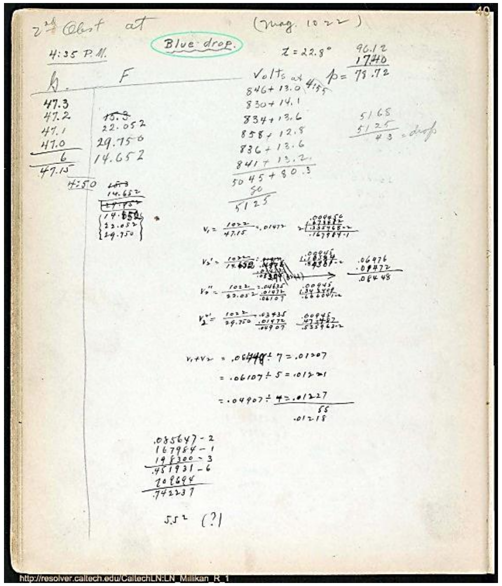
Irreproducible science

What is the difference between Potti, Sudbø and Millikan?

- All manipulated data used in publications in some way
- Millikan was right, the others were wrong

What does this have to do with reproducible science?

- Millikans notebook was important to document his published and unpublished results
- His results have been reproduced again and again
- Other researchers discovered the errors in Potti and Sudbø by studying their data



theguardian

Tenfold increase in scientific research papers retracted for fraud since 1975: Two thirds retracted for scientific misconduct, not error

nature

Findings in six of 53 landmark studies in cancer research can be confirmed

Source:

- Tenfold increase in scientific research papers retracted for fraud, Alok Jha, The Guardian, Monday 1 October 2012
- Drug development: Raise standards for preclinical cancer research, C. Glenn Begley and Lee M. Ellis, Nature 483, 2012

The scientific method and reproducible science

Roger Bacon
1214-1294

1267:
Alchemist who proposes ideas of observation, hypothesis, experimentation, and external verification

Francis Bacon
1561-1626

1620:
Important for the idea of the "scientific method"

Robert Boyle
1626-1691

1665:
"enough information must be included to allow others to independently reproduce the finding"

Inspired by slides presented by Victoria Stodden at ICERM, 2012

Reproducible Research Movement

- 1991: Professor Jon Claerbout (Stanford) requires theses of his students to be reproducible (geophysics)
- A lot of researchers at other institutions see similar problems, and use similar ideas
  - Randy Leveque, Sergey Fomel, David Donoho, Kai Diethelm, ...
- Scandals in some disciplines (cancer research, genomics,...) cause policy changes and huge (local) awareness
- Special issue on in Computing in Science and Engineering (2009)

Reproducible Research Movement

Increasing awareness around 2010 with a large number of workshops:

- **2009:**
  Yale 2009: Roundtable on Data and Code Sharing in the Computational Sciences
- **2011:**
  SIAM CSE 2011: Verifiable, Reproducible Computational Science
  AAAS 2011: The Digitization of Science: Reproducibility and Interdisciplinary ...
  ENAR International Biometric Society 2011: Panel on Reproducible Research
  SIAM Geosciences 2011 Reproducible and Open Source Software in the Geosciences
  AMP / ICIAM 2011 Community Forum on Reproducible Research Policies
  AMP 2011 Reproducible Research: Tools and Strategies for Scientific Computing
- **2012:**
  ICERM 2012 Reproducibility in Computational and Experimental Mathematics
  Supercomputing 2012, Reproducibility of Results
- **2013:**
  eVITA Winter School on Reproducible Science and Modern Scientific Software Development

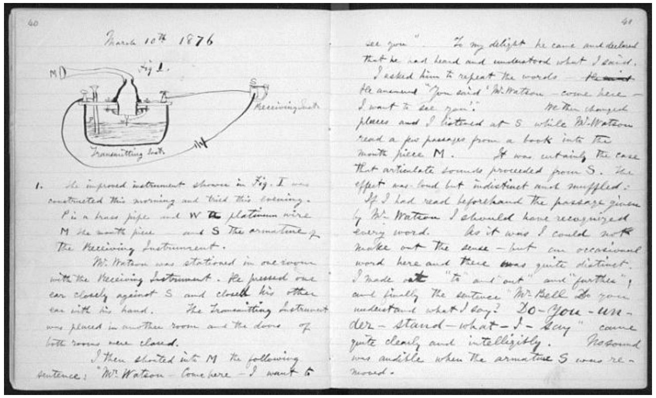
Reproducible Research Movement

- Important journals start changing their policies
  - Allow for supplementary material
  - Require statements on reproducibility
  - …

- Image Processing On Line
  - All published algorithms are available as an interactive web-service
  - http://www.ipol.im/
  - Nicolas Limare of the IPOL editorial board is attending the winter school

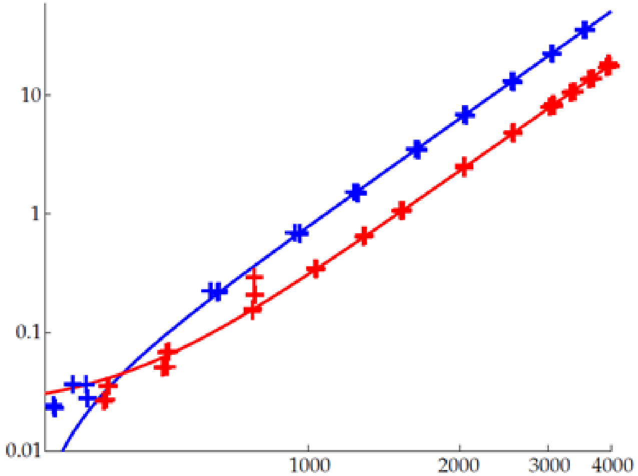
Reproducible Research Movement

- Reproducible science is thoroughly embedded in the scientific work-flow in many disciplines
  - An experimental scientist will keep a lab notebook over all experiments.
  - The notebook will enable him to reproduce the experiment

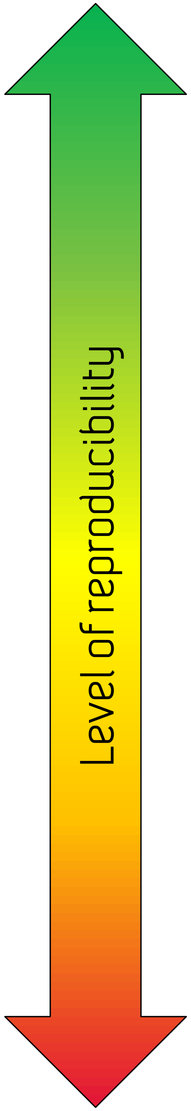


Lab notebook of Graham Bell, 1876

- In computational science, however, this lab notebook has lost its place in the natural work-flow:
  - Possible workflow: write a program, create a graph, change the program, create a graph, etc.
    - Changes can be small: a parameter change, domain size change, …
  - The graphs are kept for publication
  - The program changes are forgotten…



50 Shades of Reproducible Research



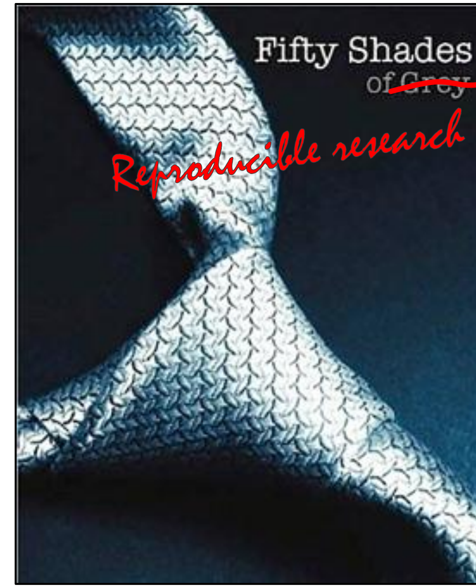
Interactively reproducible: All figures, tables, and data in a paper can be reproduced with the original data, or I can supply my own data through a web service and get new graphs and results.

Turn-key reproducible: All figures, tables, and data in a paper can be reproduced by compiling and running the program at the click of a button.

Publicly reproducible: All figures, tables, and data in the paper would be possible to reproduce for someone else, but they'd have to manually compile the program and all of its dependencies.

Privately reproducible: All figures, tables, and data in a paper would be possible to reproduce, albeit with a great deal of effort, by myself or one of my co-authors.

Irreproducible: It would not be possible for me to recreate the results I published.



"The person most likely to reproduce
your work is your own future self"
-- Sergey Fomel at ICERM workshop

- The most convincing reason for me to be reproducible, is that somewhere down the line:
  - I will have to re-do the graph with different axes because a reviewer asked,
  - I will have to reinterpret the data for an updated conclusion,
  - I will write a journal paper based on a conference paper,
  - I will (hopefully☺) write a book or book chapter based on previous results,
  - …

More reasons for reproducible research

- Reproducible papers are cited almost 5 times as often!
- Easier to get collaborations started
- Start-up-time for Masters and Ph.D. students cut down from months/years to weeks.
- Fraudulent research is not reproducible☺, reproducible research should not be fraudulent.
- ACM journals starting with "stamps of approval" for reproducible research
- …

[1] Code Sharing Is Associated with Research Impact in Image Processing, Patrick Vandewalle, Computing in Science & Engineering, 2012

The importance of reproducible research

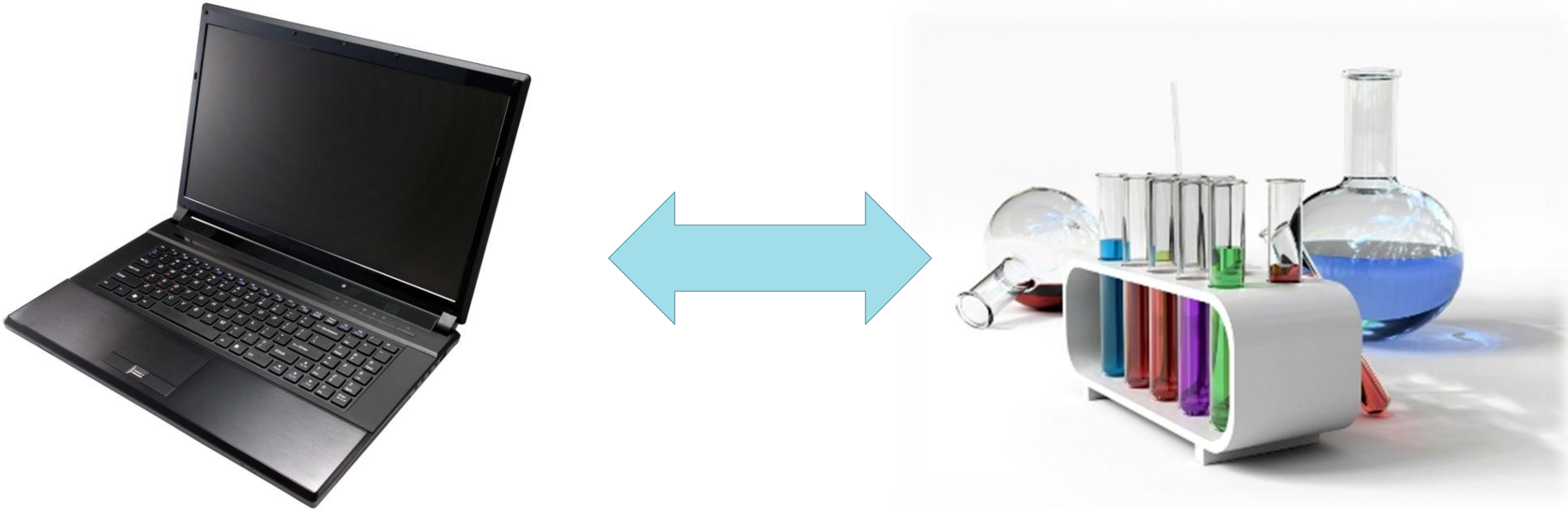
Computational science cannot be elevated to a third branch of the scientific method until it generates routinely verifiable knowledge.
--Donoho, et al. 2009

Science is the art of building pyramids of knowledge, one small block of knowledge at a time

"If I have seen further it is by standing on the shoulders of giants."
-- Isaac Newton

All Gizah Pyramids, CC-BY-SA 2.0, Ricardo Liberato

Inspired by slides presented by Sergey Fomel and Victoria Stodden at ICERM, 2012



A chemist treats the laboratory with thorough respect, having strict security procedures, careful note taking during experiments, etc. So should we also treat our computers, as it is the laboratory of computational science.

Modern Scientific Software Development

Commercial versus academic software development

- Commercial software development is driven by the motive of **making money**
  - **Inherent requirement to quality**: Customer must feel that the software is worth the money it costs

- Academic software development is driven by the motive of **publishing papers**
  - **Inherent requirement to work at least once**. After the publication has been completed, further research and development can rapidly die out.

The sad truth

"In academia, software quality, user interfaces, documentation, testing and reproducibility, will all be sacrificed at the altar of publications"

Working reproducibly must become a central part of your everyday work cycle: it is not enough to think of it as a post publication step (which never happens, anyway…)

The sad truth

- Many (Most?) scientists have not been taught software development practices
  - Researchers are taught as physicists, chemists, biologists, etc.
  - "**Self-taught** software developers" with **little or no formal training**

- Many hold the belief that software development is something you can pick up, whilst {mathematics, chemistry, …} is something that you must study.
  - Might be true for simple Matlab programming
  - But it's also like lab safety: not something you can pick up for advanced and demanding experiments

- Academic software has a tendency to grow into huge monsters over the years
  - Example: NASTRAN: Developed at NASA in late 1960-ies. 1 million lines of code, Fortran, still sold commercially.
  - **Software quality: Pay me now or pay me later**

Scientific Software Development

- Commercial software development has progressed rapidly:
  - Version control, issue trackers, ...
  - Extreme programming, pair programming, scrum, ...
  - Test driven development, continuous builds, ...
  - Dedicated testing and quality assurance teams
  - ...
- Many, but perhaps not all, of these tools and methodologies are great aids in reproducible research.
  - Some tools are also specifically created for scientific software: lab notebooks ala Ipython, Sage, etc.

Version control & issue trackers

- **Version control**
  - Subversion, mercurial, git, …
  - A simple kind of backup system.
  - Stores what a file looks like at a given time
  - By regularly checking in changes to your program, you get a history of the source code development.
  - Also a tool for collaboration

- **Issue trackers**
  - Bugzilla, trac, …
  - A kind of registry for bugs and planned developments for your source code
  - Makes sure you have a history of fixed issues, and that you don't forget an important bug.

Version control and provenance

- Provenance is imperative for reproducible research
  - Provenance is the history that describes what has happened to the data and source code used to produce results

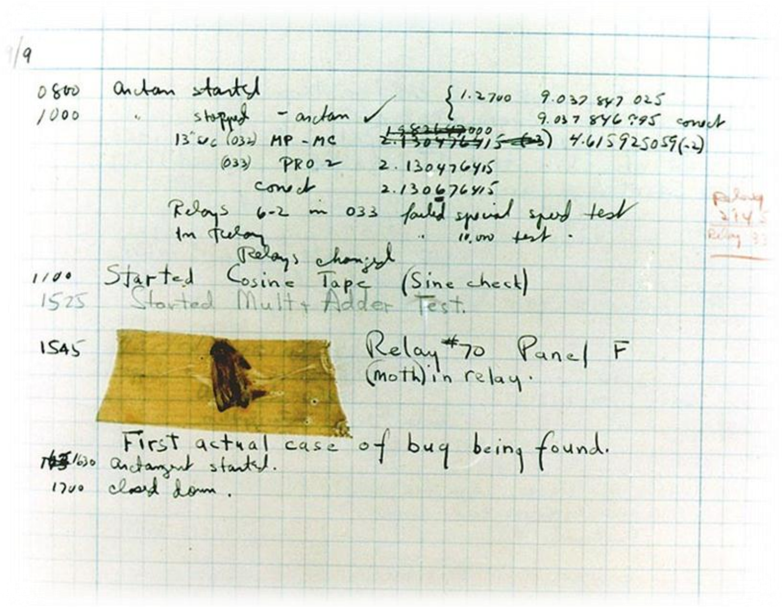
- In its simplest form:
  What version of the software and what version of the data produced the results?

- The use of version control software gives provenance for source code
  - More difficult with dependencies (version of operating system, libraries, etc.)
  - More difficult with large data-sets, as most version control systems are not really designed for it

- Important: must be able to match output results to a given version of the data and source code

Software testing

- Software testing is important for having trust in computer programs
- The simplest kind of test, a regression test, will check that the program output does not change
- Feature tests and unit tests that test specific features and parts of the software give the expected output
- Testing of fixed bugs to make sure they do not reappear
- More advanced tests include verification and validation



First computer bug, Harvard Mk. II, 1947

Continuous builds

- One danger of using testing is that tests quickly are disabled or ignored as the paper deadline approaches…
- Continuous builds tackle this by automatically testing each new revision in the source code repository, and creating a report on the results
- Makes it easy to get a history of development, and discover when and why something suddenly changed.

Summary

- Reproducible research does not come for free
  - The pros outweigh the cons: pay me now or pay me later
  - Requires that we include reproducibility in our daily work-flows

- Modern software development practices are important tools
  - Without version control, it is extremely difficult to have reproducible research
  - Many techniques and tools make software development faster, less error prone, and makes it easy to collaborate

Further references

- **ICERM Reproducibility in Computational and Experimental Mathematics**
  http://icerm.brown.edu/tw12-5-rcem
  http://wiki.stodden.net/ICERM_Reproducibility_in_Computational_and_Experimental_Mathematics

- **Reproducible Research: Tools and Strategies for Scientific Computing**
  http://stodden.net/AMP2011/

- **Best Practices for Scientific Computing**
  Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Katy Huff, Ian M. Mitchell, Mark Plumbley, Ben Waugh, Ethan P. White, Paul Wilson
  (Submitted on 1 Oct 2012 (v1), last revised 29 Nov 2012 (this version, v3))
  http://arxiv.org/abs/1210.0530