Reproducibility in Computational Science

Randall J. LeVeque
Applied Mathematics
University of Washington

Reproducibility of one's own results

- Version control,
- Automated regression testing,
- Workflow software,
- Etc.

All this is important, but not the focus of this talk.

Want to concentrate on reproducing results of others.

Outline

- Top 10 reasons not to share your code (and why you should anyway).

- What does reproducibility mean?
- What do/can/should journals do?
- What do/can/should funding agencies do?
- Code and data repositories?
- Scaling to Petascale/Exascale?
- Discussion

Top 10 Reasons to Not Share Your Code

(and why you should anyway)

Randall J. LeVeque
Applied Mathematics
University of Washington

Talk from SIAM CSE Conference
March, 2011

Alternate reality

Imagine a world in which mathematics papers contain:

- Lemmas, Theorems, Corollaries
- No proofs

Alternate reality

Imagine a world in which mathematics papers contain:

- Lemmas, Theorems, Corollaries
- No proofs

Nobody expects to see a proof in a publication,
or to ever have to submit one.

Alternate reality

Imagine a world in which mathematics papers contain:

- Lemmas, Theorems, Corollaries
- No proofs

Nobody expects to see a proof in a publication,
or to ever have to submit one.

This is the way it's always been and there are lots of good theorems in the literature, so why change?

Alternate reality

Imagine a world in which mathematics papers contain:

- Lemmas, Theorems, Corollaries
- No proofs

Nobody expects to see a proof in a publication,
or to ever have to submit one.

This is the way it's always been and there are lots of good theorems in the literature, so why change?

Suppose a group of cranks started suggesting papers should contain proofs...

Alternate reality

Imagine a world in which mathematics papers contain:

- Lemmas, Theorems, Corollaries
- No proofs

Nobody expects to see a proof in a publication,
or to ever have to submit one.

This is the way it's always been and there are lots of good theorems in the literature, so why change?

Suppose a group of cranks started suggesting papers should contain proofs...

Many objections would be raised...

Alternate reality (no proofs in papers)

Some objections...

Alternate reality (no proofs in papers)

Some objections...

1. The proof is too ugly to show anyone else.

Alternate reality (no proofs in papers)

Some objections...

1. The proof is too ugly to show anyone else.

1a. It would be too much work to rewrite it neatly so others could read it.

Alternate reality (no proofs in papers)

Some objections...

1. The proof is too ugly to show anyone else.

1a. It would be too much work to rewrite it neatly so others could read it.

1b. It's a one-off proof for this particular theorem, not intended for others to see or use.

Alternate reality (no proofs in papers)

Some objections...

1. The proof is too ugly to show anyone else.

1a. It would be too much work to rewrite it neatly so others could read it.

1b. It's a one-off proof for this particular theorem, not intended for others to see or use.

1c. My time is much better spent proving another result and publishing more papers rather than putting more effort into this one, which I've already proved.

Alternate reality (no proofs in papers)

2. I didn't work out all the details.

Alternate reality (no proofs in papers)

2. I didn't work out all the details.

2a. Some tricky cases I didn't want to deal with, but the proof works fine for most cases, such as the ones I used in my examples.

Alternate reality (no proofs in papers)

2. I didn't work out all the details.

2a. Some tricky cases I didn't want to deal with, but the proof works fine for most cases, such as the ones I used in my examples.

2b. I discovered some cases actually don't work, but as long as I don't mention it nobody will notice.

Alternate reality (no proofs in papers)

2. I didn't work out all the details.

2a. Some tricky cases I didn't want to deal with, but the proof works fine for most cases, such as the ones I used in my examples.

2b. I discovered some cases actually don't work, but as long as I don't mention it nobody will notice.

2c. I didn't actually prove the theorem, my student did.

Alternate reality (no proofs in papers)

2. I didn't work out all the details.

2a. Some tricky cases I didn't want to deal with, but the proof works fine for most cases, such as the ones I used in my examples.

2b. I discovered some cases actually don't work, but as long as I don't mention it nobody will notice.

2c. I didn't actually prove the theorem, my student did.

And... the student has since disappeared, along with the proof, but I'm sure it was correct!

Alternate reality (no proofs in papers)

3. The proof is valuable intellectual property.

Alternate reality (no proofs in papers)

3. The proof is valuable intellectual property.

3a. It took years to prove this theorem, why should I give the proof away freely.

Alternate reality (no proofs in papers)

3. The proof is valuable intellectual property.

3a. It took years to prove this theorem, why should I give the proof away freely.

3b. The same idea can be used to prove other theorems.
I deserve at least 5 more papers before sharing the proof.

Alternate reality (no proofs in papers)

3. The proof is valuable intellectual property.

3a. It took years to prove this theorem, why should I give the proof away freely.

3b. The same idea can be used to prove other theorems.
I deserve at least 5 more papers before sharing the proof.

3c. Someone else might use the ideas in my proof without giving me proper credit.

Alternate reality (no proofs in papers)

3. The proof is valuable intellectual property.

3a. It took years to prove this theorem, why should I give the proof away freely.

3b. The same idea can be used to prove other theorems.
I deserve at least 5 more papers before sharing the proof.

3c. Someone else might use the ideas in my proof without giving me proper credit.

3d. The idea is so great I can commercialize and sell the proof.
(see Dijkstra's **Mathematics, Inc.**)

Alternate reality (no proofs in papers)

4. Technical difficulties.

Alternate reality (no proofs in papers)

4. Technical difficulties.

4a. Including proofs would make math papers *much* longer. Journals wouldn't want to publish them.

Alternate reality (no proofs in papers)

4. Technical difficulties.

4a. Including proofs would make math papers *much* longer. Journals wouldn't want to publish them.

4b. Referees would never want to have to read proofs. It would be too hard to determine correctness of long proofs and finding referees would become impossible.

Alternate reality (no proofs in papers)

4. Technical difficulties.

Alternate reality (no proofs in papers)

4. Technical difficulties.

4c. The proof uses sophisticated mathematical machinery that most readers/referees don't know. (Their hardware/software cannot fully execute the proof.)

Alternate reality (no proofs in papers)

4. Technical difficulties.

4c. The proof uses sophisticated mathematical machinery that most readers/referees don't know. (Their hardware/software cannot fully execute the proof.)

So there's no point publishing it if most people only read the theorems and ignore the proofs.

Alternate reality (no proofs in papers)

4. Technical difficulties.

4c. The proof uses sophisticated mathematical machinery that most readers/referees don't know. (Their hardware/software cannot fully execute the proof.)

So there's no point publishing it if most people only read the theorems and ignore the proofs.

4d. My proof uses other theorems with unpublished (proprietary) proofs, so it won't help to publish my proof — readers still will not be able to fully verify correctness.

Back to the real world...

Back to the real world...

Papers in numerical mathematics and computational science often contain:

- Algorithms, often incompletely described,
- Graphs or tables demonstrating properties claimed,
- Lots of pretty pictures,
- No actual code with all the details.

Back to the real world...

Papers in numerical mathematics and computational science often contain:

- Algorithms, often incompletely described,
- Graphs or tables demonstrating properties claimed,
- Lots of pretty pictures,
- No actual code with all the details.

"Yes", you may say, "but codes are different than proofs."

Back to the real world...

Papers in numerical mathematics and computational science often contain:

- Algorithms, often incompletely described,
- Graphs or tables demonstrating properties claimed,
- Lots of pretty pictures,
- No actual code with all the details.

"Yes", you may say, "but codes are different than proofs."

Let's examine some code issues...

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

- It may still be very valuable to aid in the readers' understanding.

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

- It may still be very valuable to aid in the readers' understanding.
- It's an important part of the scientific record.

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

- It may still be very valuable to aid in the readers' understanding.
- It's an important part of the scientific record.
- Bugs are often found when cleaning it up!

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

- It may still be very valuable to aid in the readers' understanding.
- It's an important part of the scientific record.
- Bugs are often found when cleaning it up!
- You will be glad you did sometime down the road.

Reasons for not sharing code

It's not software, it's a research code that isn't worth cleaning up for others to see.

- It may still be very valuable to aid in the readers' understanding.
- It's an important part of the scientific record.
- Bugs are often found when cleaning it up!
- You will be glad you did sometime down the road.
- People understand that not all code is software. Much more embarrassing things appear on the web.

comments on this story

Stories by subject

Lab life

This article elsewhere

Blogs linking to this article

Add to Connotea

Add to Digg

Add to Facebook

Published online 13 October 2010 | *Nature* **467**, 753 (2010) | doi:10.1038/467753a

Column: World View

Publish your computer code: it is good enough

Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says Nick Barnes.

Nick Barnes

www.nature.com/news/2010/101013/full/467753a.html

Reasons for not sharing code

Intellectual property: Giving away code is very different from describing in detail how an experiment is done.

"It is more like inviting all other scientists to use my lab."

Reasons for not sharing code

Intellectual property: Giving away code is very different from describing in detail how an experiment is done.

"It is more like inviting all other scientists to use my lab."

- "Giving away" a proof is similar.

Reasons for not sharing code

Intellectual property: Giving away code is very different from describing in detail how an experiment is done.

"It is more like inviting all other scientists to use my lab."

- "Giving away" a proof is similar.
- Often that's what we're paid to do. Grant support to develop algorithms requires making them available.

Reasons for not sharing code

Intellectual property: Giving away code is very different from describing in detail how an experiment is done.

"It is more like inviting all other scientists to use my lab."

- "Giving away" a proof is similar.
- Often that's what we're paid to do. Grant support to develop algorithms requires making them available.

  So we're not "giving it away".

Reasons for not sharing code

Intellectual property: Giving away code is very different from describing in detail how an experiment is done.

"It is more like inviting all other scientists to use my lab."

- "Giving away" a proof is similar.
- Often that's what we're paid to do. Grant support to develop algorithms requires making them available.
  So we're not "giving it away".

(Codes developed in industry of labs may be different.)

Reasons for not sharing code

Lack of credit:

People will use parts of my code without attribution.

Reasons for not sharing code

Lack of credit:

People will use parts of my code without attribution.

- If culture changes so all publications are accompanied by code, this will be impossible to hide.

Reasons for not sharing code

Lack of credit:

People will use parts of my code without attribution.

- If culture changes so all publications are accompanied by code, this will be impossible to hide.
- Publishing it with paper gives it a timestamp and provenance.

Reasons for not sharing code

Lack of credit:

People will use parts of my code without attribution.

- If culture changes so all publications are accompanied by code, this will be impossible to hide.
- Publishing it with paper gives it a timestamp and provenance.
- Making your algorithms more understandable leads to more users and citations.

Reasons for not sharing code

Lack of credit:

People will use parts of my code without attribution.

- If culture changes so all publications are accompanied by code, this will be impossible to hide.
- Publishing it with paper gives it a timestamp and provenance.
- Making your algorithms more understandable leads to more users and citations.
- Trying to use a research code for a new problem is often impossible without involvement of the authors.

  $\Longrightarrow$ new collaborations.

Reasons for not sharing code

Impossible to run by others:

Require proprietary software, or

Only runs on supercomputers, or

Requires too many dependencies, or

May work today but probably won't in 5 years.

Reasons for not sharing code

Impossible to run by others:

Require proprietary software, or

Only runs on supercomputers, or

Requires too many dependencies, or

May work today but probably won't in 5 years.

- Ability to view the part of the code directly related to paper is often most important.

Reasons for not sharing code

Impossible to run by others:

Require proprietary software, or

Only runs on supercomputers, or

Requires too many dependencies, or

May work today but probably won't in 5 years.

- Ability to view the part of the code directly related to paper is often most important.
- Techniques like Virtualization can help with some technical issues.

Reasons for not sharing code

Lack of credit:

I get no credit in my department / field for the time required.

Reasons for not sharing code

Lack of credit:

I get no credit in my department / field for the time required.

- This will change with time, but only if we get started.

Reasons for not sharing code

Lack of credit:

I get no credit in my department / field for the time required.

- This will change with time, but only if we get started.
- Set a good example and push for change.

Meaning of reproducible computational science?

Some people mean: (replicable a better term?)

Full code is available to download and run.
Can obtain exactly the same results as in paper.
(And this will be true forever.)

Meaning of reproducible computational science?

Some people mean: (replicable a better term?)

Full code is available to download and run.
Can obtain exactly the same results as in paper.
(And this will be true forever.)

Technical problems: (Not to mention social ones...)

- Code has many dependencies...
  Compilers, software packages, visualization tools, ...
- Some of these may be commercial/proprietary
- May only run on special hardware (e.g. Leadership class)
- All software evolves...
  Even compiler changes can affect results.

Meaning of reproducible computational science?

Another possible definition (as in experimental science):

> Paper describes the algorithm sufficiently well that a reader can implement and obtain essentially the same results.

Meaning of reproducible computational science?

Another possible definition (as in experimental science):

> Paper describes the algorithm sufficiently well that a reader can implement and obtain essentially the same results.

Many paper fail at this.

Meaning of reproducible computational science?

Another possible definition (as in experimental science):

> Paper describes the algorithm sufficiently well that a reader can implement and obtain essentially the same results.

Many paper fail at this.

Middle ground:

Encourage sharing portions of code directly related to the new work being published.

The ability to inspect code greatly improves chances of reproducing results, or using ideas effectively in future work.

Journal policies

Some journals require sharing of data and/or code.

For example Science, as of February, 2011:

> *"All data necessary to understand, assess, and extend the conclusions of the manuscript must be available to any reader of Science. All computer codes involved in the creation or analysis of data must also be available to any reader of Science. After publication, all reasonable requests for data and materials must be fulfilled. "*

http://www.sciencemag.org/site/feature/contribinfo/prep/gen_info.xhtml#dataavail

Funding agency policies

NSF now requires a Data Management Plan for all new proposals.

"Data" includes computer code.

Funding agency policies

NSF now requires a Data Management Plan for all new proposals.

"Data" includes computer code.

DOE SciDAC requires open access to codes funded.

Funding agency policies

NSF now requires a Data Management Plan for all new proposals.

"Data" includes computer code.

DOE SciDAC requires open access to codes funded.

NIH requires access to data and publications.

Supplementary materials in journals

Many journal allow supplementary material or other on-line resources.

Often only for animations, datasets, etc.

Sometimes also for code.

Supplementary materials in journals

(e.g. SIAM, Society for Industrial and Applied Mathematics)

Possible approach:

- Allow authors to upload static snapshot of code with submission of article.
- Possibly choose whether or not to be refereed.
- Can also include link to another website for evolving code, bug fixes, wiki for feedback, etc.

Supplementary materials in journals

(e.g. SIAM, Society for Industrial and Applied Mathematics)

Possible approach:

- Allow authors to upload static snapshot of code with submission of article.
- Possibly choose whether or not to be refereed.
- Can also include link to another website for evolving code, bug fixes, wiki for feedback, etc.

Available repositories:

- Authors' website (not stable!)
- Open source / commercial hosting services, such as
  www.sourceforge.net, code.google.com, bitbucket.org, github.com,

  May disappear, or change business model.

Stable Public Data Repositories?

Could be used by all journals, rather than many different solutions.

Include version control?

Include wiki for posting bugs, comments, etc.?

- “Reputation system” to complement peer review
- Works very well for large open source projects.

Stable Public Data Repositories?

Could be used by all journals, rather than many different solutions.

Include version control?

Include wiki for posting bugs, comments, etc.?

- "Reputation system" to complement peer review
- Works very well for large open source projects.

Note: open source infrastructure already exists!

See...

www.sourceforge.net, code.google.com, bitbucket.org, github.com,

Virtualization

One approach to archiving full software environment

For example **www.virtualbox.org** runs on Linux, Mac, Windows.

Download Virtual Machince and open in VirtualBox to obtain a full environment with OS, compilers, software, visualization, etc.

Virtualization

One approach to archiving full software environment

For example **www.virtualbox.org** runs on Linux, Mac, Windows.

Download Virtual Machince and open in VirtualBox to obtain a full environment with OS, compilers, software, visualization, etc.

Problem: VM is large (≥ 1GB), even if code of interest is small.

Possible solution: Provide standard VMs that can be used for broad classes of code?

Virtualization

One approach to archiving full software environment

For example **www.virtualbox.org** runs on Linux, Mac, Windows.

Download Virtual Machince and open in VirtualBox to obtain a full environment with OS, compilers, software, visualization, etc.

Problem: VM is large (≥ 1GB), even if code of interest is small.

Possible solution: Provide standard VMs that can be used for broad classes of code?

How stable is VirtualBox? Need open source version?

Discussion

- Top 10 reasons not to share your code (and why you should anyway).
- What does reproducibility mean?
- What do/can/should journals do?
- What do/can/should funding agencies do?
- Code and data repositories?