Reproducible software vs. reproducible research

Fernando Pérez
http://fperez.org
Fernando.Perez@berkeley.edu

Helen Wills Neuroscience Institute, UC Berkeley

AAAS Annual Meeting, Washington, DC.
Feb 19, 2011
Outline

1. A contrast of cultures
2. Technical ideas: tools matter
3. Incentives and rewards: changing our practices
Outline

1. A contrast of cultures
2. Technical ideas: tools matter
3. Incentives and rewards: changing our practices
Who am I?

**Background**
- Particle physics (theory/computation): numerical QCD
- Applied mathematics: algorithm development for PDEs
- Neuroscience: algorithms and tools for brain imaging

**A common thread: computational tools**
- IPython: interactive Python
- Matplotlib: visualization
- Numpy: numerics
- Scipy: scientific algorithms
- Nipy: neuroimaging tools
Reproducible research practices!

Reproducibility at publication time?  
It’s already too late.

Learn from a community (open source) where reproducibility is an everyday practice (by necessity)
Reproducible research practices!

Reproducibility at publication time?
It’s already too late.

Learn from a community (open source) where reproducibility is an everyday practice (by necessity)
Reproducible research practices!

Reproducibility at publication time?
It’s already too late.

Learn from a community (open source) where reproducibility is an everyday practice (by necessity)
Fast adaptive algorithms in the non-standard form for multidimensional problems

Gregory Beylkin *, Vani Cheruvu, Fernando Pérez

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526, USA
Received 6 June 2007; accepted 2 August 2007
Available online 14 August 2007
Communicated by Vladimir Rokhlin

Abstract
We present a fast, adaptive multiresolution algorithm for applying integral operators with a wide class of radially symmetric kernels in dimensions one, two and three. This algorithm is made efficient by the use of separated representations of the kernel. We discuss operators of the class $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \geq 0$ and $0 < \alpha < 3/2$, and illustrate the algorithm for the Poisson and Schrödinger equations in dimension three. The same algorithm may be used for all operators with radially symmetric kernels.
a periodic analogue of the Hilbert transform. In order to find its representation in multiwavelet bases, we compute

$$r_{ii'}^{j:l} = 2^{-j} \int_{-1}^{1} K(2^{-j}(x+l))\Phi_{ii'}(x) \, dx = 2^{-j} \int_{-1}^{1} \cot(\pi 2^{-j}(x+l))\Phi_{ii'}(x) \, dx,$$  \hspace{1cm} (12)$$

where $\Phi_{ii'}(x)$, $i, i' = 0, \ldots, k - 1$ are cross-correlation functions described in Appendix A.4 and $l = 0, \pm 1, \pm 2, \ldots, 2^j - 1$. We compute $r_{ii'}^{j:l}$ using the convergent integrals

$$r_{ii'}^{j:l} = 2^{-j} \sum_{k=i'-i}^{i'+i} c_{ii'}^{k} \int_{0}^{1} \Phi_{k,0}^+(x) \left( \cot(\pi 2^{-j}(x+l)) + (-1)^{i+i'} \cot(\pi 2^{-j}(-x+l)) \right) \, dx,$$

where $\Phi_{k,0}^+$ is a polynomial described in Appendix A.4. In our numerical experiment, we apply (11) to the periodic function on $[0, 1]$,

$$f(x) = \sum_{k \in \mathbb{Z}} e^{-a(x+k-1/2)^2},$$
Tables and pretty figures. Now in color!

Table 1
Results from evaluating (13) with our algorithm

<table>
<thead>
<tr>
<th>$p$</th>
<th>Scales</th>
<th>$N_{\text{blocks}}$</th>
<th>$\epsilon$</th>
<th>$E_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[2,3,4]</td>
<td>8</td>
<td>$10^{-3}$</td>
<td>$1.5 \times 10^{-4}$</td>
</tr>
<tr>
<td>8</td>
<td>[2,4,5]</td>
<td>12</td>
<td>$10^{-6}$</td>
<td>$1.3 \times 10^{-7}$</td>
</tr>
<tr>
<td>11</td>
<td>[2,4,5]</td>
<td>14</td>
<td>$10^{-9}$</td>
<td>$1.1 \times 10^{-10}$</td>
</tr>
<tr>
<td>14</td>
<td>[3,4,5]</td>
<td>16</td>
<td>$10^{-12}$</td>
<td>$4.4 \times 10^{-13}$</td>
</tr>
</tbody>
</table>

Notes. The order of the basis $p$ is adjusted as a function of the requested precision $\epsilon$. The second column indicates scales present in the adaptive tree for the input. The third column shows the total number of blocks of coefficients in this tree. The last column ($E_2$) shows the actual error of the computed solution in the $\ell^2$ norm.

Fig. 3. Results of applying the cotangent kernel to a periodized Gaussian using basis of order $p = 14$ (the last row in Table 1). The pointwise error is shown on the right for a requested accuracy of $\epsilon = 10^{-12}$. 

Chance of reproducing results for third parties?

$\mathcal{O}(10^{-\text{something very big}})$
Registration-wall software

STRFlab
Spatio-temporal receptive field lab

Download STRFlab
Please register to download STRFlab:

Name
Institute
Lab
Email

Comments

Sign up for mailing list: [ ]

Download
BSMART: A Matlab/C Toolbox for Analyzing Brain Circuits

BSMART, an acronym of Brain-System for Multivariate AutoRegressive Timeseries, is an open-source software package for analyzing that was born out of a collaborative research effort between Dr. Hualou Liang at Drexel University, Dr. Steven Bressler at Florida Atla Ding at University of Florida. BSMART can be applied to a wide variety of neuroelectromagnetic phenomena, including EEG unique feature of the BSMART package is Granger causality that can be used to assess causal influences and directions of directions of different time scales. Based upon a MAR model, a plethora of spectral quantities such as auto power, partial power, coherency coherence and Granger causality can be immediately derived. The approach has been fruitfully used to characterize, with high resolution, functional relations within large scale brain networks.

The BSMART is currently undergoing beta test, freely available under the GNU public license (download BSMART). It is supported by Neurological Disorders and Stroke (NINDS) through the NIH Neuroinformatics / Human Brain Projects.

The BSMART is described in:


Please refer to this article when publishing results obtained from this toolbox. For any queries or comments please contact Hualou Liang.
Contrast: FOSS better than scientific research?
FOSS: Free and Open Source Software

Public distributed version control: provenance tracking
Adapt magic commands to new history system.

This grew from issue ipython/IPython#245. Various magic commands weren't working properly with the new history system: %edit, %macro, and %hist.

Among various minor troubles, selecting a range of lines (`%macro test 2-5`) numbered from the beginning of the history, so didn't match up with the current line numbers. I've approached this by adding a session_offset attribute to the history manager. This has the added benefit that we no longer need to store a blank history entry so we can count lines from 1.

Along the way, I simplified and modernised parts of the code, including using `basestring` over `StringTypes` and `.isdigit()` over an equivalent regex.
Pull requests: back and forth discussion

```
@@ -77,6 +80,9 @@ class HistoryManager(object):
     # pre-processing. This will allow users to retrieve the input just as
     # it was exactly typed in by the user, with %hist -r.
     self.input_hist_raw = []
+
+     # Offset so the first line of the current session is #1
+     self.session_offset = -1
```

Since this is a new attribute, it should be listed at the class level, for details see: [http://ipython.scipy.org/doc/nightly/html/development/coding_guide.html#attribute-declarations-for-objects](http://ipython.scipy.org/doc/nightly/html/development/coding_guide.html#attribute-declarations-for-objects)
Branches: exploratory work with control
Automated tests: validation/reproducibility
The VTK Build Dashboard: immediate feedback

9 files changed by 5 authors as of Friday, February 18 2011 20:00:00 EST

<table>
<thead>
<tr>
<th>Site</th>
<th>Build Name</th>
<th>Update</th>
<th>Configure</th>
<th>Build</th>
<th>Test</th>
<th>Build Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Files</td>
<td>Error</td>
<td>Warn</td>
<td>Min</td>
<td>Error</td>
</tr>
<tr>
<td>p90r03.pbm.ihost.com</td>
<td>AIX00F614-xIIC</td>
<td>8</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>4.4</td>
</tr>
<tr>
<td>londinium.kitware</td>
<td>Arch-GCC-4.5-x86_64-debug</td>
<td>8</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>londinium.kitware</td>
<td>Arch-GCC-4.5-x86_64-release</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>amber1.kitware</td>
<td>Debian4-x64-gcc</td>
<td>8</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>1.2</td>
</tr>
<tr>
<td>ID</td>
<td>Type</td>
<td>Status</td>
<td>Priority</td>
<td>Milestone</td>
<td>Owner</td>
<td>Summary + Labels</td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
<td>--------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>2018</td>
<td>Defect</td>
<td>Accepted</td>
<td>Critical</td>
<td>Release0.7.0</td>
<td><a href="mailto:smi...@gmail.com">smi...@gmail.com</a></td>
<td>terms vs factors NeedsBetterPatch smichr</td>
</tr>
<tr>
<td>2120</td>
<td>Defect</td>
<td>Accepted</td>
<td>Critical</td>
<td>Release0.7.0</td>
<td><a href="mailto:smi...@gmail.com">smi...@gmail.com</a></td>
<td>fix doclest or quality testing to recognize docests NeedsBetterPatch smichr Testing</td>
</tr>
<tr>
<td>2133</td>
<td>Defect</td>
<td>Started</td>
<td>Critical</td>
<td>Release0.7.0</td>
<td><a href="mailto:matt...@gmail.com">matt...@gmail.com</a></td>
<td>Merge new polynomials manipulation module Polynomial NeedsBetterPatch mppap NeedsReview</td>
</tr>
<tr>
<td>2151</td>
<td>Defect</td>
<td>Accepted</td>
<td>Critical</td>
<td>Release0.7.0</td>
<td><a href="mailto:Ronan.L...@gmail.com">Ronan.L...@gmail.com</a></td>
<td>BasicMeta.keep_sign Series</td>
</tr>
<tr>
<td>1276</td>
<td>Defect</td>
<td>Started</td>
<td>High</td>
<td>Release0.7.0</td>
<td>----</td>
<td>solve(-1 + x**2 + 0.1111111111111111*(1.0000000000000 + 2.0000000000000*x)**2, x) fails EasyToFix Polynomial Solvers</td>
</tr>
<tr>
<td>1721</td>
<td>Defect</td>
<td>Accepted</td>
<td>High</td>
<td>Release0.7.0</td>
<td><a href="mailto:Ronan.L...@gmail.com">Ronan.L...@gmail.com</a></td>
<td>Rename class 'Real'</td>
</tr>
<tr>
<td>1735</td>
<td>Defect</td>
<td>New</td>
<td>High</td>
<td>Release0.7.0</td>
<td><a href="mailto:Ronan.L...@gmail.com">Ronan.L...@gmail.com</a></td>
<td>Rename .func attribute (.args too?)</td>
</tr>
<tr>
<td>1919</td>
<td>Enhancement</td>
<td>Accepted</td>
<td>High</td>
<td>Release0.7.0</td>
<td>Vinzent.Steinberg</td>
<td>unify behavior of var() and symbols() NeedsReview smichr mppap</td>
</tr>
<tr>
<td>51</td>
<td>Enhancement</td>
<td>Started</td>
<td>Medium</td>
<td>Release0.7.0</td>
<td><a href="mailto:matt...@gmail.com">matt...@gmail.com</a></td>
<td>RootOf for polynomial equations Polynomial NeedsReview mppap</td>
</tr>
<tr>
<td>326</td>
<td>Defect</td>
<td>Started</td>
<td>Medium</td>
<td>Release0.7.0</td>
<td><a href="mailto:matt...@gmail.com">matt...@gmail.com</a></td>
<td>sympy.roots_.sturm(...) hangs Polynomial EasyToFix NeedsReview mppap</td>
</tr>
<tr>
<td>527</td>
<td>Enhancement</td>
<td>Started</td>
<td>Medium</td>
<td>Release0.7.0</td>
<td>----</td>
<td>guessing what functions, like integrate, roots, factor, apart (and many more), should do with the given expression NeedsReview mppap</td>
</tr>
</tbody>
</table>
Nitetime: time-series analysis for neuroscience

Auditory processing in grasshoppers

Extracting the average time-series from one signal, time-locked to the occurrence of some type of event in another signal is a very typical operation in the analysis of time-series from neuroscience experiments. Therefore, we have an additional example of this kind of analysis in

Event-related fMRI

In the following code-snippet, we demonstrate the calculation of the spike-triggered average (STA). This is the average of the stimulus wave-form preceding the emission of a spike in the neuron and can be thought of as the stimulus ‘preferred’ by this neuron.

We start by importing the required modules:

```python
import numpy as np
import nitime.timeseries as ts
import nitime.analysis as tsa
import nitime.viz as viz
```

Two data files are used in this example. The first contains the times of action potentials
plt.show() is called in order to display the figures

```python
plt.show()
```

The data used in this example is also available on the [CRCNS data sharing web-site](http://example.com).  


---

**Example source code**

You can download [the full source code of this example](http://example.com). This same script is also included in the Nitime source distribution under the `doc/examples/` directory.
Outline

1. A contrast of cultures
2. Technical ideas: tools matter
3. Incentives and rewards: changing our practices
Version control everywhere
Git: the tool you didn’t know you needed

Reproducibility?

- Tracking and recreating every step of your work
- In the software world: it’s called Version Control!

Git: an enabling technology. Use Version control for everything

- Paper writing
- Grant writing
- Everyday research

Advantages of pervasive DVCS

- Tracking of everyday results. A “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.
## Version control everywhere

**Git: the tool you didn’t know you needed**

### Reproducibility?
- Tracking and recreating every step of your work
- In the software world: it’s called Version Control!

### Git: an enabling technology. Use Version control for everything
- Paper writing
- Grant writing
- Everyday research

### Advantages of pervasive DVCS
- Tracking of everyday results. A “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.
Version control everywhere
Git: the tool you didn’t know you needed

Reproducibility?
- Tracking and recreating every step of your work
- In the software world: it’s called Version Control!

Git: an enabling technology. Use Version control for everything
- Paper writing
- Grant writing
- Everyday research

Advantages of pervasive DVCS
- Tracking of everyday results. A “time machine” view.
- Distributed backup.
- Explore lines of research/writing.
- Collaborate with colleagues.
The dna network graph

All branches in the network using msporny/dna as the reference point. Read our blog post about how it works.

Last updated: 1 day ago
Outline

1. A contrast of cultures
2. Technical ideas: tools matter
3. Incentives and rewards: changing our practices
Incentives and rewards

In Open Source

- Individual attribution in commit logs.
- Volunteers find reward in community.
- For some, it’s part of their job.
- No hidden work before “publication”: the *process* is open.

Academia: a naïve transplant won’t work

- **Success ⇐⇒** individual authorship.
- Fears of scooping from open development.
- Low/no requirements from journals
  - But changing! E.g. *Biostatistics, Open Research Computation*.
- Similarly for funding agencies.
  - Also changing: new NSF data management requirements.
Adopt a *habit* of reproducibility
Make version control as routine as email

- Git for your next *grant*
- Git in your next in-house *research project*.
- Disk is cheap! Separate repositories for:
  - **Libraries**: automated tests and docs *during the development process*.
  - **In-house tools** shared across project but of less generic use.
  - **Project/dataset** specific repositories.

- **Write** your next *paper* with a repository that can produce all results/figures.
- **Publish** your next *paper* with the code/data repository for it
  - Properly licensed, see V. Stodden’s standard.
Use your influence to improve the situation

**FINAL NIH STATEMENT ON SHARING RESEARCH DATA**

...Reviewers will **not** factor the proposed data-sharing plan into the determination of scientific merit or priority score. [emphasis mine]

This must change!

- **Grant review panel**
  - Credit proposals that do a good job on this front, note those that don’t.

- **Hiring/tenure/promotion committee**
  - credit good computational work.

- **Teaching:**
  - students must treat computing as rigorously as any other aspect of the research.
Use your influence to improve the situation

FINAL NIH STATEMENT ON SHARING RESEARCH DATA

...Reviewers will not factor the proposed data-sharing plan into the determination of scientific merit or priority score. [emphasis mine]

This must change!

- Grant review panel
  - Credit proposals that do a good job on this front, note those that don’t.

- Hiring/tenure/promotion committee
  - credit good computational work.

- Teaching:
  - students must treat computing as rigorously as any other aspect of the research.
Open Research Computation is an open access journal that publishes articles describing the development, capacities, and uses of software for researchers in any field. The journal also encourages submissions that review or describe developments relating to software based research tools. All software source code published in *Open Research Computation* is made available under an Open Source Initiative compliant license.

Submit your manuscript and benefit from:

• High visibility for articles through unrestricted online access
• No limits on article length, additional files, colour figures or movies
• Immediate open access publication on acceptance
• Expert peer review

[www.openresearchcomputation.com](http://www.openresearchcomputation.com)