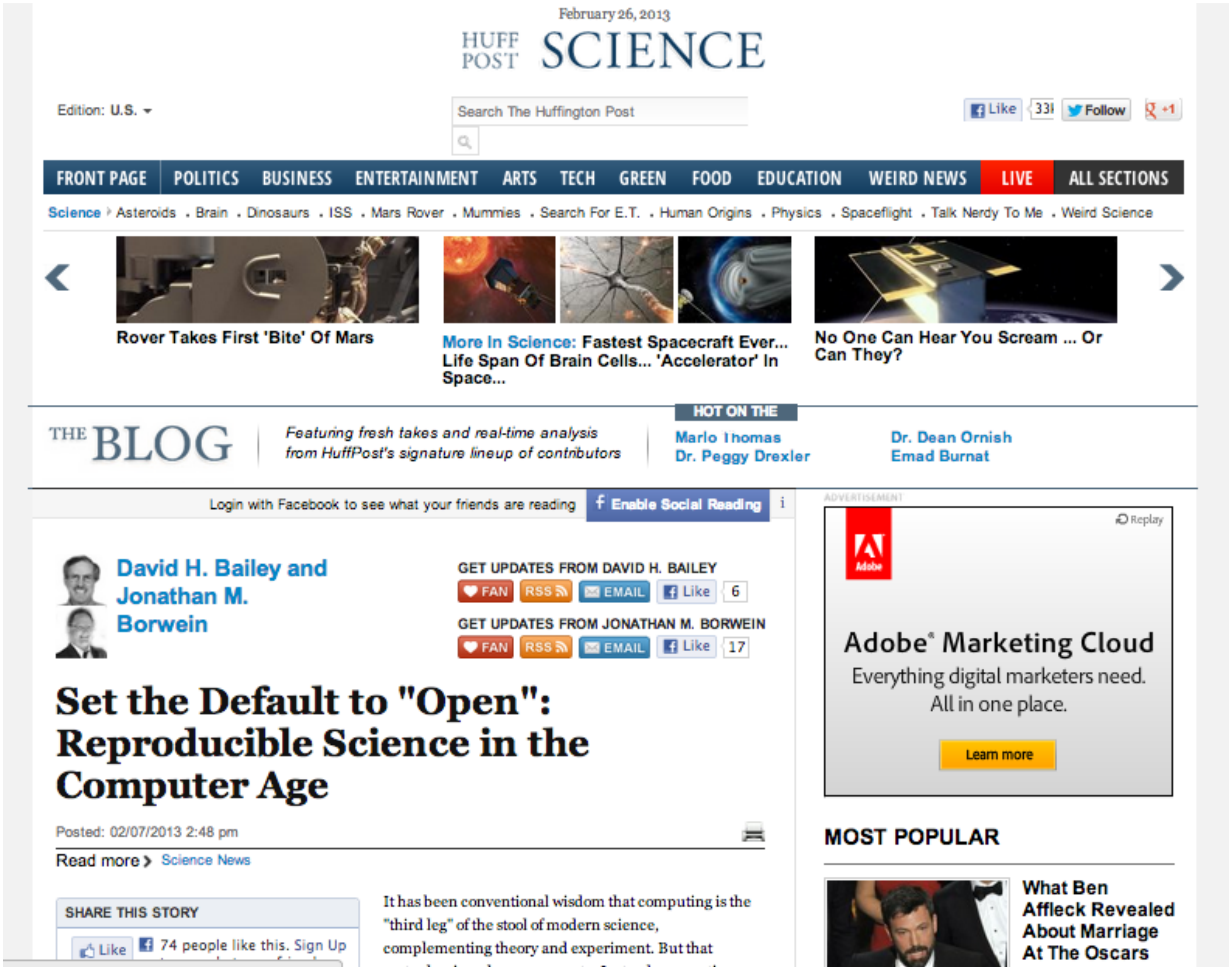
Workshop: Reproducibility in Computational and Experimental Mathematics

Institute for Computational and Experimental Research in Mathematics

Dec 10-14 2012

icerm.brown.edu



February 26, 2013

HUFF POST SCIENCE

Edition: U.S.

Search The Huffington Post

Like 33 Follow +1

FRONT PAGE POLITICS BUSINESS ENTERTAINMENT ARTS TECH GREEN FOOD EDUCATION WEIRD NEWS LIVE ALL SECTIONS

Science › Asteroids . Brain . Dinosaurs . ISS . Mars Rover . Mummies . Search For E.T. . Human Origins . Physics . Spaceflight . Talk Nerdy To Me . Weird Science

Rover Takes First 'Bite' Of Mars

More In Science: Fastest Spacecraft Ever... Life Span Of Brain Cells... 'Accelerator' In Space...

No One Can Hear You Scream ... Or Can They?

THE BLOG

Featuring fresh takes and real-time analysis from HuffPost's signature lineup of contributors

HOT ON THE

Marlo Thomas
Dr. Peggy Drexler
Dr. Dean Ornish
Emad Burnat

Login with Facebook to see what your friends are reading Enable Social Reading

ADVERTISEMENT

Replay

Adobe

Adobe® Marketing Cloud

Everything digital marketers need. All in one place.

Learn more

David H. Bailey and Jonathan M. Borwein

GET UPDATES FROM DAVID H. BAILEY

FAN RSS EMAIL Like 6

GET UPDATES FROM JONATHAN M. BORWEIN

FAN RSS EMAIL Like 17

Set the Default to "Open": Reproducible Science in the Computer Age

Posted: 02/07/2013 2:48 pm

Read more › Science News

SHARE THIS STORY

Like 74 people like this. Sign Up

It has been conventional wisdom that computing is the "third leg" of the stool of modern science, complementing theory and experiment. But that

MOST POPULAR

What Ben Affleck Revealed About Marriage At The Oscars

This report summarizes discussions that took place during the ICERM Workshop on Reproducibility in Computational and Experimental Mathematics, held December 10-14, 2012. The main recommendations that emerged from the workshop discussions are:

1. It is important to promote a culture change that will integrate computational reproducibility into the research process.
2. Journals, funding agencies, and employers should support this culture change.
3. Reproducible research practices and the use of appropriate tools should be taught as standard operating procedure in relation to computational aspects of research.

The workshop discussions included presentations of a number of the diverse and rapidly growing set of software tools available to aid in this effort. We call for a broad implementation of these three recommendations across the computational sciences.

For example, workshop participants recommend that software and data be "open by default" unless it conflicts with other considerations. Proposals involving computational work might be required to provide details such as:

- Extent of computational work to be performed.
- Platforms and software to be utilized.
- Reasonable standards for dataset and software documentation, including reuse (some agencies already have such requirements [8]).
- Reasonable standards for persistence of resulting software and dataset preservation and archiving.
- Reasonable standards for sharing resulting software among reviewers and other researchers.

In addition, we suggest that funding agencies might add "reproducible research" to the list of specific examples that proposals could include in their requirements such as "Broader Impact" statements. Software and dataset curation should be explicitly included in grant proposals and recognized as a scientific contribution by funding agencies. Templates for data management plans could be made available that include making software open and available, perhaps by funding agencies, or by institutional archiving and library centers. [6]

A number of suggestions were made regarding best practices for publications of research results. To aid in reproducibility, the available materials should ideally contain:

- A precise statement of assertions to be made in the paper.
- A statement of the computational approach, and why it constitutes a rigorous test of the hypothesized assertions.
- Complete statements of, or references to, every algorithm employed.
- Salient details of auxiliary software (both research and commercial software) used in the computation.
- Salient details of the test environment, including hardware, system software and the number of processors utilized.

[9] Indeed, one needs to know which precise functions were called, with what parameter values and environmental settings?

- Salient details of data reduction and statistical analysis methods.
- Discussion of the adequacy of parameters such as precision level and grid resolution.
- Full statement (or at least a valid summary) of experimental results.
- Verification and validation tests performed by the author(s).
- Availability of computer code, input data and output data, with some reasonable level of documentation.
- Curation: where are code and data available? With what expected persistence and longevity? Is there a site for site for future updates, e.g. a version control repository of the code base?
- Instructions for repeating computational experiments described in the paper.
- Terms of use and licensing. Ideally code and data “default to open”, i.e. a permissive re-use license, if nothing opposes it.
- Avenues of exploration examined throughout development, including information about negative findings.
- Proper citation of all code and data used, including that generated by the authors.

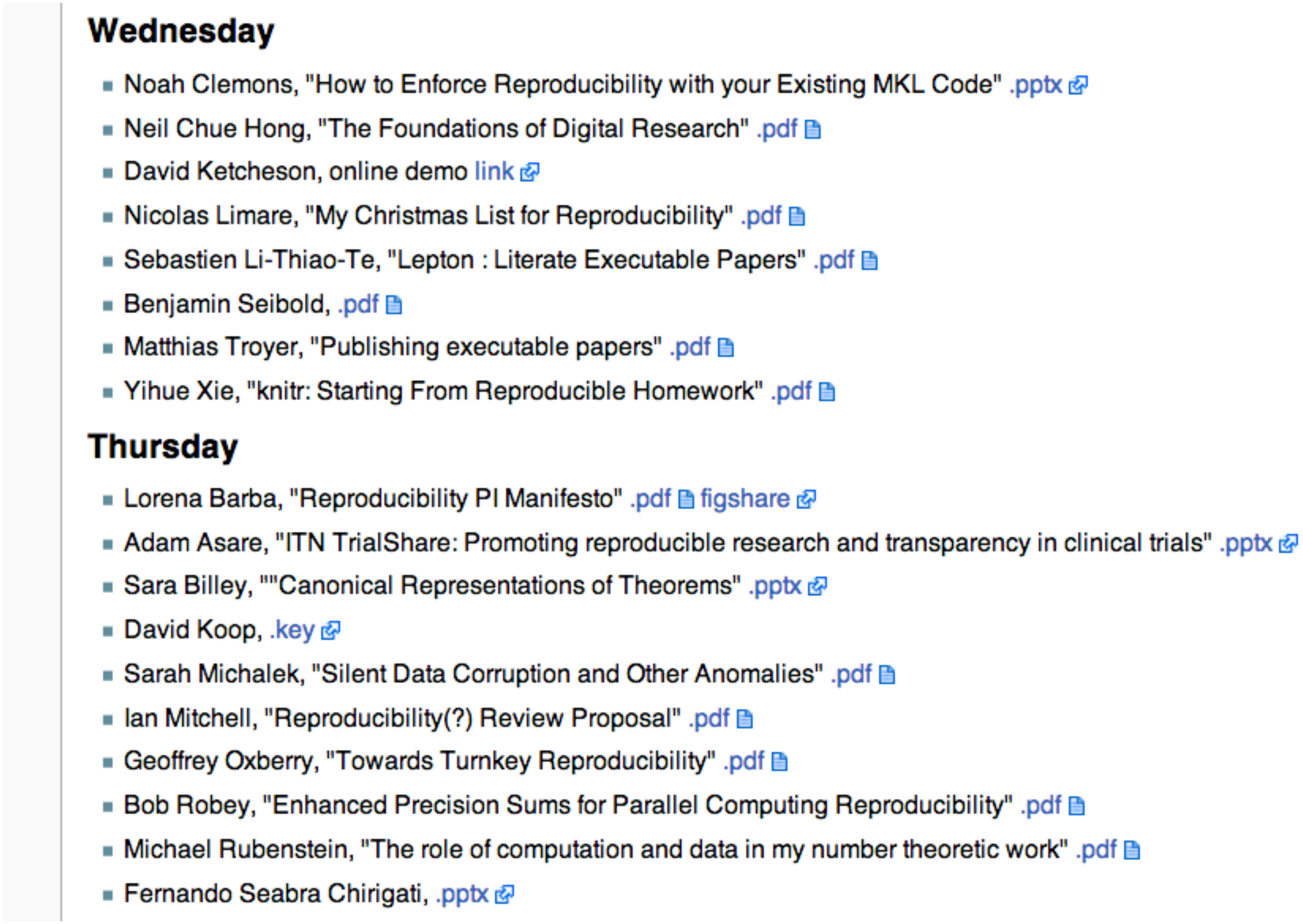
Literate programming, authoring, and publishing tools. These tools enable users to write and publish documents that integrate the text and figures seen in reports with code and data used to generate both text and graphical results. In contrast to notebook-based tools discussed below, this process is typically not interactive, and requires a separate compilation step. Tools that enable literate programming include both programming-language-specific tools such as WEB, Sweave, and knitr, as well as programming-language-independent tools such as Dexy, Lepton, and noweb. Other authoring environments include SHARE, Doxygen, Sphinx, CWEB, and the Collage Authoring Environment.

Tools that define and execute structured computation and track provenance. Provenance refers to the tracking of chronology and origin of research objects, such as data, source code, figures, and results. Tools that record provenance of computations include VisTrails, Kepler, Taverna, Sumatra, Pegasus, Galaxy, Workflow4ever, and Madagascar.

Integrated tools for version control and collaboration. Tools that track and manage work as it evolves facilitate reproducibility among a group of collaborators. With the advent of version control systems (e.g., Git, Mercurial, SVN, CVS), it has become easier to track the investigation of new ideas, and collaborative version control sites like Github, Google Code, BitBucket, and Sourceforge enable such ideas to be more easily shared. Furthermore, these web-based systems ease tasks like code review and feature integration, and encourage collaboration.

Tools that express computations as notebooks. These tools represent sequences of commands and calculations as an interactive worksheet with pretty printing and integrated displays, decoupling content (the data, calculations) from representation (PDF, HTML, shell console), so that the same research content can be presented in multiple ways. Examples include both closed-source tools such as MATLAB (through the publish and app features), Maple, and Mathematica, as well as open-source tools such as IPython, Sage, RStudio (with knitr), and TeXmacs.

Tools that capture and preserve a software environment. A major challenge in reproducing computations is installing the prerequisite software environment. New tools make it possible to exactly capture the computational environment and pass it on to someone who wishes to reproduce a computation. For instance, VirtualBox, VMWare, or Va



Wednesday

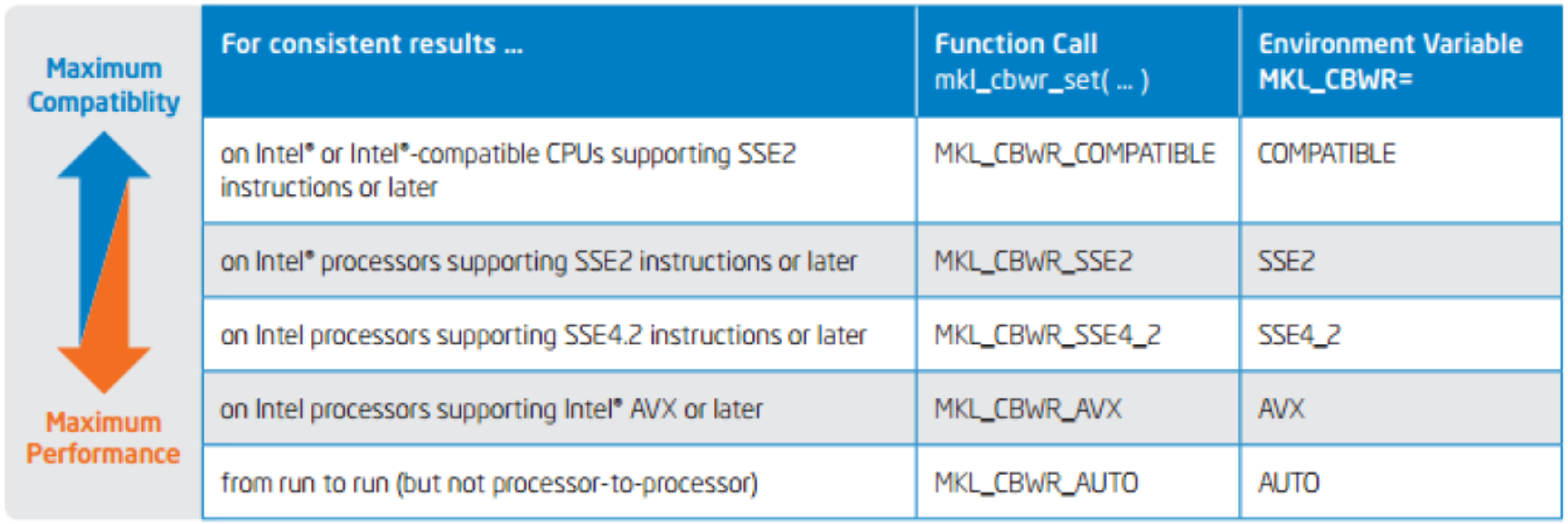
- Noah Clemons, "How to Enforce Reproducibility with your Existing MKL Code" .pptx
- Neil Chue Hong, "The Foundations of Digital Research" .pdf
- David Ketcheson, online demo link
- Nicolas Limare, "My Christmas List for Reproducibility" .pdf
- Sebastien Li-Thiao-Te, "Lepton : Literate Executable Papers" .pdf
- Benjamin Seibold, .pdf
- Matthias Troyer, "Publishing executable papers" .pdf
- Yihue Xie, "knitr: Starting From Reproducible Homework" .pdf

Thursday

- Lorena Barba, "Reproducibility PI Manifesto" .pdf figshare
- Adam Asare, "ITN TrialShare: Promoting reproducible research and transparency in clinical trials" .pptx
- Sara Billey, ""Canonical Representations of Theorems" .pptx
- David Koop, .key
- Sarah Michalek, "Silent Data Corruption and Other Anomalies" .pdf
- Ian Mitchell, "Reproducibility(?) Review Proposal" .pdf
- Geoffrey Oxberry, "Towards Turnkey Reproducibility" .pdf
- Bob Robey, "Enhanced Precision Sums for Parallel Computing Reproducibility" .pdf
- Michael Rubenstein, "The role of computation and data in my number theoretic work" .pdf
- Fernando Seabra Chirigati, .pptx

3 Types of Non-Reproducibility in Intel® Math Kernel Library

- Run to Run – same processor
- Runs between different Intel processors
- Runs between different IA-compabible processors



Maximum Compatiblity

Maximum Performance

For consistent results ...	Function Call mkl_cbwr_set(...)	Environment Variable MKL_CBWR=
on Intel® or Intel®-compatible CPUs supporting SSE2 instructions or later	MKL_CBWR_COMPATIBLE	COMPATIBLE
on Intel® processors supporting SSE2 instructions or later	MKL_CBWR_SSE2	SSE2
on Intel processors supporting SSE4.2 instructions or later	MKL_CBWR_SSE4_2	SSE4_2
on Intel processors supporting Intel® AVX or later	MKL_CBWR_AVX	AVX
from run to run (but not processor-to-processor)	MKL_CBWR_AUTO	AUTO

Lepton : Literate Executable Papers

Lepton is a tool to do research as opposed to publishing reproducible research results. It deals with :

- **everyday tasks** such as programming and writing technical reports
- **reviewing** the methods and results by collaborators and in the long term
- **re-using** source code, input data, research results

Further references :

- Website `http://www.math.univ-paris13.fr/~lithiao/ResearchLepton/Lepton.html`
  with program for download, manual, faq, examples
- 2 conference papers
  - Sébastien Li-Thiao-Té. Literate program execution for reproducible research and executable papers. *Procedia Computer Science*, 9(0):439 – 448, 2012. ICCS 2012.
  - Sébastien Li-Thiao-Té. Literate program execution for teaching computational science. *Procedia Computer Science*, 9(0):1723 – 1732, 2012. ICCS 2012.
- A poster + demonstration at ICERM

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OLE

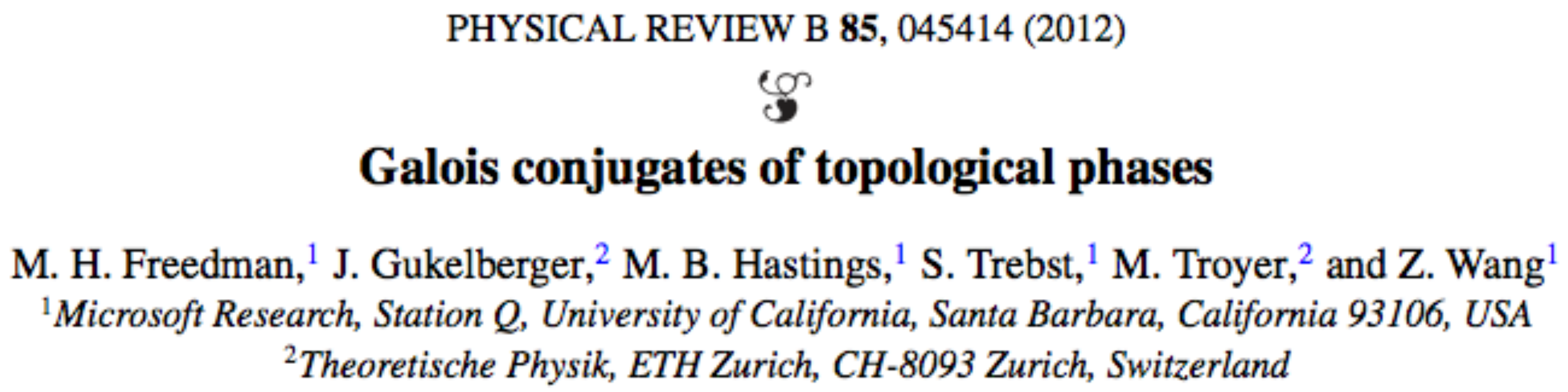
DARPA

DPHYS
Department of Physics
Institute for Theoretical Physics

Publishing executable papers

Matthias Troyer and Jan Gukelberger (ETH Zurich)
Michael H. Freedman (Microsoft)

with help from the VisTrails team,
especially David Koop, Emanuele Santos, and Juliana Freire



PHYSICAL REVIEW B **85**, 045414 (2012)

Galois conjugates of topological phases

M. H. Freedman,[1] J. Gukelberger,[2] M. B. Hastings,[1] S. Trebst,[1] M. Troyer,[2] and Z. Wang[1]
[1]*Microsoft Research, Station Q, University of California, Santa Barbara, California 93106, USA*
[2]*Theoretische Physik, ETH Zurich, CH-8093 Zurich, Switzerland*

Conclusions of an observer (vjc)

- Community of numerical scientists interested in reproducible research concepts is growing and energetic
- Tools for fostering reproducibility of results in science are growing
  - Some indication of duplicative parallel development
  - Learning curves need to be flattened
- Provenance/persistence/culture issues are significant and require considerable investment