UNIVERSITY *of* DELAWARE

# The three Rs of work in scientific papers: repeatability, replicability and reproducibility

**Michela Taufer**

University of Delaware

## Acknowledgments

- SCC Team: Hai Ah Nam, Stephen Harrell, Scott Michael, and Kristopher Garrett
- SC Reproducibility Advisory Board: Wilf Pinfold, Victoria Stodden, Michael Heroux, Fernando Perez
- ACM and the participants in ACM meetings in May 2015- 2016, in particular Grigori Fusin and Bruce Childers

## Reproducible Accuracy

- From Van Nostrand's Scientific Encyclopedia

  Reproducibility: "closeness of agreement among repeated simulation results under the same initial conditions over time"

  Accuracy: "conformity of a resulted value to an accepted standard (or scientific laws)"

- Repeatability (Same team, same experimental setup)
  - The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.
- Replicability (Different team, same experimental setup)
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- Reproducibility (Different team, different experimental setup)
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

From: https://www.acm.org/publications/policies/artifact-review-badging

- Repeatability (Same team, same experimental setup)
  - The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.
- Replicability (Different team, same experimental setup)
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author’s own artifacts.
- Reproducibility (Different team, different experimental setup)
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

From: https://www.acm.org/publications/policies/artifact-review-badging

- Repeatability (Same team, same experimental setup)
  - The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.
- Replicability (Different team, same experimental setup)
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- Reproducibility (Different team, different experimental setup)
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

From: https://www.acm.org/publications/policies/artifact-review-badging

## ACM Badges

- Artifacts Evaluated:
  - Applied to papers whose associated artifacts have successfully completed an independent audit
- Artifacts Available:
  - / ose auth ive k ically acc r
- Resu
  - / hich the p essfully c or team other than the author

Artifacts Evaluated
acm
Functional

Artifacts Evaluated
acm
Reusable

From: http://www.acm.org/publications/policies/artifact-review-badging

## ACM Badges

- Artifacts Evaluated:
  - Applied to papers whose associated artifacts have successfully completed an independent audit
- Artifacts Available:
  - Applied to papers whose author-created artifacts have been placed on a publically accessible archival repository
- Results Validated:
  - Applied to papers in which the main paper have been successfully obtaine ɔr team other than the author

From: http://www.acm.org/publications/policies/artifact-review-badging

## ACM Badges

- Arti
  - ose ass
    d an ir
- Arti
  - ose au have
    ically
    repository

acm
Results Replicated
acm
Results Reproduced

- Results Validated:
  - Applied to papers in which the main results of the paper have been successfully obtained by a person or team other than the author

From: http://www.acm.org/publications/policies/artifact-review-badging

## Reproducibly Initiate at SC16 *(Part 1)*

- Reproducibility highlighted through the Student Cluster Competition (SCC)
- Call for SCC reproducibility application proposal
    - Target Tech Papers and Gordon Bell Papers in SC13, SC14, and SC15 proceedings.
- Authors self-identify reproducibility and submit SCC proposal
- SCC committee volunteers assess reproducibility & suitability for SCC competition application of submitted papers
- SCC committee selected one paper to become a SCC application

## SC16 Selected Paper

Patrick Flick, Chirag Jain, Tony Pan, and Srinivas Aluru.
“A parallel connectivity algorithm for de Bruijn graphs in metagenomic applications,” SC15.

UNIVERSITY *of* DELAWARE

## Reproducibly Initiate at SC16 *(Part 2)*

- SCC committee works with authors of selected paper to reproduce paper and create suitable competition application and competition parameters
  - Replicate work in paper (before competition)
  - Extend work at competition (active authors' support)

→SIGHPC certificate of appreciation at SC16

## SCC Challenge at SC16

- Students received the paper and the code before SC16
- Students' challenges:
  - Replicate the work for a different dataset
  - Analyze impact of different architecture
  - Analyze impact of downscale
  - Write a report (structure provided)
  - Discuss the experience in the exit interview
- Assignment counts for 25% of the entire competition
- Best reports will be collected in a special issue of PARCO

## SCC Challenge at SC17

- Select the paper for the SCC application @SC17 from the SC16 papers
- Invite authors to express availability to join the initiative at the time of the paper submission
- Invite authors of accepted papers to submit *Artifact Descriptor*
  - 2-page appendix describing the execution of the work in the paper
  - 3 SC16 finalist papers → selected paper announced on Dec 7

UNIVERSITY of DELAWARE

# Reproducibly Initiate at SC17

- Pierre Jolivet and Pierre-Henri: “Tournier Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers”
- Duane Merrill and Michael Garland: “Merge-based Parallel Sparse Matrix-Vector Multiplication”
- Sangeetha Abdu Jyothi, Ankit Singla, P. Brighten Godfrey and Alexandra Kolla: “Measuring and Understanding Throughput of Network Topologies”
- W. Michael Brown, Andrey Semin, Michael Hebenstreit, Sergey Khvostov, Karthik Raman and Steven J. Plimpton: “Increasing Molecular Dynamics Simulation Rates with an 8-Fold Increase in Electrical Power Efficiency”
- Ehab Abdelhamid, Ibrahim Abdelaziz, Panos Kalnis, Zuhair Khayyat and Fuad Jamour: “ScaleMine: Scalable Parallel Frequent Subgraph Mining in a Single Large Graph”
- Markus Hoehnerbach, Ahmed E. Ismail and Paolo Bientinesi: “The Vectorization of the Tersoff Multi-Body Potential: An Exercise in Performance Portability”
- Shaden Smith, Jongsoo Parky and George Karypis: “An Exploration of Optimization Algorithms for High Performance Tensor Completion”
- Maria Dimakopoulou, St´ephane Eranian, Nectarios Koziris and Nicholas Bambos: “Reliable and Efficient Performance Monitoring in Linux”
- Preeti Malakar, Venkatram Vishwanath, Christopher Knight, Todd Munson and Michael E. Papka: “Optimal Execution of Co-analysis for Large-scale Molecular Dynamics Simulations”

# Artifact Description: Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers

Pierre Jolivet*
*CNRS
IRIT–ENSEEIHT
pierre.jolivet@enseeiht.fr

Pierre-Henri Tournier[‡§]
[‡]Laboratoire J.-L. Lions, UPMC
[§]Inria Paris, ALPINES research team
tournier@ann.jussieu.fr

## Appendix

### A. Abstract

This description contains the information needed to launch some experiments of the SC16 paper "Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers". More precisely, we explain how to compile and run the modified PETSc examples used in section IV. The results from section V can be reproduced using a finite element library interfaced with HPDDM, but this artifact description is not focused on that part of the paper.

### B. Description

*1) Check-list (artifact meta information):*

- **Algorithm:** GCRO-DR with right, left, or variable preconditioning
- **Program:** C binary, C and C++ libraries
- **Compilation:** icpc version 16.0.2.181 (gcc version 4.9.1 compatibility) with the -O3 flag
- **Output:** time to solution and number of iterations
- **Experiment workflow:** install PETSc, clone HPDDM, compile the HPDDM C library, compile the modified PETSc examples, run the binaries, observe the results
- **Experiment customization:** number of MPI processes, threads, and grid points, standard parameters of Krylov methods...
- **Publicly available?:** yes

*2) How delivered:* HPDDM can be cloned from GitHub using the following URL: https://github.com/hpddm/hpddm. The examples taken from the PETSc distribution are in the folder `examples/petsc`.

*3) Hardware dependencies:* none. Note that we will link binaries with shared objects. As such, systems with severe limitations when it comes to dynamic loading—e.g., IBM BlueGene/Q—are not covered in this document (but it is not a problem to use static libraries instead).

*4) Software dependencies:* HPDDM requires a C++11 compliant compiler such as: g++ 4.7.2 and above, clang++ 3.3 and above, icpc 15.0.0.090 and above, or pgc++ 15.1 and above. icpc 16.0.1.150 and below and pgc++ are partially bugged when activating C++11 support. Please apply the following patch to the sources of HPDDM first if you are using one of these compilers:

```
$ sed -i\ '' 's/type\* = nullptr/type* = (void*)0 \
  /g; s/static constexpr const char/const char   \
  /g' include/*.hpp examples/*.cpp
```

BLAS and LAPACK are needed for dense linear algebra computations but can be automatically downloaded by PETSc. PETSc is available at the following URL: http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-lite-3.7.3.tar.gz. Do not forget to turn off debugging and error detection if you need to compile PETSc (`--with-debugging=0 --with-errorchecking=0`).

### C. Installation

1) clone HPDDM and enter the newly created directory

```
$ git clone https://github.com/hpddm/hpddm
$ cd hpddm
```

2) create an appropriate `Makefile.inc` by defining:
   a) `MPICXX`, a C++ compiler wrapping an MPI implementation
   b) `CXXFLAGS`, to activate C++11 support and such
   c) `BLAS_LIBS`, to link with BLAS and LAPACK
   d) or if you are using Intel Math Kernel Library, define `MKL_INCS` and `MKL_LIBS` instead

   Here are some minimalist `Makefile.inc` examples, be sure to link PETSc and HPDDM with the same BLAS and MPI implementations as they are not all ABI compatible:

   i. for Linux-based systems with the legacy BLAS

```
MPICXX    = mpic++
CXXFLAGS  = -std=c++11 -O3 -fPIC
BLAS_LIBS = -L/usr/lib -lblas -llapack
```

   ii. for Linux-based systems with Intel MKL and GOMP

```
MPICXX   = mpic++
CXXFLAGS = -std=c++11 -O3 -fPIC
MKL_LIBS = -lgomp -L${MKLROOT}/lib/intel64          \
  -lmkl_core -lmkl_intel_lp64 -lmkl_gnu_thread
MKL_INCS = -I${MKLROOT}/include
```

   iii. for macOS systems with Apple BLAS

```
MPICXX    = mpic++
CXXFLAGS  = -std=c++11 -O3 -fPIC
BLAS_LIBS = -framework Accelerate
```

   iv. for macOS systems with Intel MKL and IOMP

```
MPICXX   = mpic++
CXXFLAGS = -std=c++11 -O3 -fPIC
MKL_LIBS = -L/opt/intel/lib -L/opt/intel/mkl/lib  \
  -liomp5 -lmkl_core -lmkl_intel_lp64             \
  -lmkl_intel_thread
MKL_INCS = -I/opt/intel/mkl/include
```

3) compile the C library

```
$ LIST_COMPILATION=c make lib
```

4) copy the modified PETSc examples into your PETSc installation

```
$ cp examples/petsc/ex32.c examples/petsc/ex56.c  \
  ${PETSC_DIR}/src/ksp/ksp/examples/tutorials
```

5) store the working directory in an environment variable and make sure that the shared library can be found, e.g., for some systems:

```
$ export HPDDM_DIR=`pwd`
$ export LD_LIBRARY_PATH=`pwd`/lib:\
  ${LD_LIBRARY_PATH}
```

### D. Experiment workflow

Now that the HPDDM C library is compiled, PETSc toolchain will be used for generating the binaries. For that matter, change directory and compile the modified PETSc examples:

```
$ cd ${PETSC_DIR}/src/ksp/ksp/examples/tutorials
$ make ex32 ex56 CFLAGS="-I${HPDDM_DIR}/interface \
  -L${HPDDM_DIR}/lib -lhpddm_c"
```

Most HPDDM and PETSc options may be set via command line, so there is almost no need to recompile either the library or the binaries. In the rest of the artifact description, we will explain the most important options to set up in order to reproduce the results of the paper.

### E. Evaluation and expected result

To make sure that everything runs smoothly, here are two commands (one for each modified PETSc example) that should run on most platforms (from laptops to supercomputers):

```
$ mpirun -np 8 ./ex32 -hpddm_recycle_same_system  \
  -ksp_pc_side right -ksp_rtol 1.0e-6             \
  -hpddm_recycle 10 -hpddm_krylov_method gcrodr   \
  -hpddm_gmres_restart 30 -da_refine 2
$ mpirun -np 8 ./ex56 -ne 9 -ksp_pc_side right    \
  -ksp_rtol 1.0e-6 -hpddm_gmres_restart 30        \
  -hpddm_krylov_method gcrodr -hpddm_recycle 10
```

The output for example 32 should include the following lines:

```
      PETSc (GMRES)               HPDDM (GCRO-DR)
1       81      0.005241     1       64      0.005964
2       65      0.003395     2       28      0.001851
3       77      0.003898     3       27      0.001860
4       65      0.003308     4       28      0.001987
-------------------------    -------------------------
        288     0.015842             147     0.011662
```

The first column is the index of the linear system solved, the second column is the number of iterations needed to reach convergence, and the third column is the time to solution (excluding setup) in seconds. The last line is the sum of all rows.

The output for example 56 should include the following lines (which have the same structure as described previously):

```
      PETSc (GMRES)               HPDDM (GCRO-DR)
1       128     0.018176     1       70      0.014209
2       77      0.010872     2       60      0.014578
3       98      0.013834     3       79      0.018486
4       106     0.014781     4       38      0.009578
-------------------------    -------------------------
        409     0.057663             247     0.056851
```

### F. Experiment customization

In the paper, numerical experiments were carried out with the two previously compiled examples but with the following adjusted parameters: grid size, preconditioner type, dimension of recycled Krylov subspaces. All PETSc options were disclosed in the paper, but due to double-blind review policy, HPDDM options were omitted. Here are the exact command lines including both sets of options:

- for section IV-B

```
$ mpirun -np 8192 ./ex32 -ksp_rtol 1.0e-8               \
  -pc_type gamg -ksp_type fgmres -da_refine 2           \
  -mg_levels_ksp_type gmres -da_grid_x 4210             \
  -mg_levels_ksp_max_it 3 -da_grid_y 4210               \
  -hpddm_krylov_method gcrodr -hpddm_recycle 10         \
  -hpddm_gmres_restart 30 -hpddm_tol 1.0e-8             \
  -hpddm_variant flexible -pc_gamg_square_graph 2       \
  -hpddm_recycle_strategy B                             \
  -hpddm_recycle_same_system                            \
  -pc_gamg_threshold 0.0725
$ mpirun -np 8192 ./ex32 -ksp_rtol 1.0e-8               \
  -pc_type gamg -ksp_type fgmres -da_refine 2           \
  -mg_levels_ksp_type gmres -da_grid_x 4210             \
  -mg_levels_ksp_max_it 1 -da_grid_y 4210               \
  -hpddm_krylov_method gcrodr -hpddm_recycle 10         \
  -hpddm_gmres_restart 30 -hpddm_tol 1.0e-8             \
  -hpddm_variant flexible -pc_gamg_square_graph 2       \
  -hpddm_recycle_same_system                            \
  -hpddm_recycle_strategy B                             \
  -pc_gamg_threshold 0.076
```

- for section IV-C

```
$ mpirun -np 8000 ./ex56 -ne 399 -ksp_rtol 1.0e-8 \
  -ksp_type fgmres -pc_type gamg                  \
  -mg_levels_ksp_type cg -mg_levels_ksp_max_it 4  \
  -hpddm_krylov_method gcrodr -hpddm_recycle 10   \
  -hpddm_gmres_restart 30 -hpddm_tol 1.0e-8       \
  -hpddm_variant flexible -hpddm_recycle_strategy A
$ mpirun -np 8000 ./ex56 -ne 399 -ksp_rtol 1.0e-8 \
  -ksp_type lgmres -pc_type gamg                  \
  -ksp_lgmres_augment 10 -ksp_pc_side right       \
  -mg_levels_ksp_type chebyshev -hpddm_recycle 10 \
  -hpddm_krylov_method gcrodr -hpddm_tol 1.0e-8   \
  -hpddm_gmres_restart 30 -hpddm_variant flexible \
  -hpddm_recycle_strategy A
```

The list of all available PETSc (resp. HPDDM) options may be displayed by appending the option `-help` (resp. `-hpddm_help`) to the command line arguments. Alternatively, these options are also described at the following URL: http://www.mcs.anl.gov/petsc/documentation (resp. https://github.com/hpddm/hpddm/blob/master/doc/cheatsheet.pdf)

### G. Notes

For GCRO-DR, `-hpddm_recycle_strategy A` (resp. `B`) means solving the generalized eigenvalue problem in fig. 1 (line 33) with the right-hand side matrix $W$ defined eq. (3a) (resp. eq. (3b)).

In the last experiment of section IV-C, we use FGCRO-DR instead of GCRO-DR with right preconditioning because this leads to less operations (at a cost of additional storage, which is typical of flexible iterative methods).

## What Next?

- Purse transparency in published work!
- Collecting your feedback:
  *reproducibility@info.supercomputing.org*