Accurately Citing Software and Algorithms Used in Publications

Matthew G. Knepley
Computation Institute
University of Chicago
Chicago, IL, USA
knepley@ci.uchicago.edu

Jed Brown, Lois Curfman McInnes, and Barry Smith
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL, USA
[jedbrown,mcinnes,bsmith]@mcs.anl.gov

August 31, 2013

Abstract

Properly citing academic publications that describe software libraries and algorithms is the way that open source scientific library users "pay" to use the free software. With large multifaceted libraries and applications that use several such libraries, even the conscientious user ends up citing publications in error or missing relevant publications. Some open source developers list appropriate citations on their website or in their documentation. Based on a recent addition to the PETSc numerical software libraries, we suggest an alternative model where the library itself generates the bibtex items based on **exactly** what algorithms and portions of the code are used in the application.

The PETSc numerical libraries [1, 2, 6] implement hundreds of published algorithms and can use over 50 optional external software packages. When users publish results based on a simulation involving PETSc, how do they know what papers they should cite as relevant

and essential to their simulation? We have recently added a new feature to PETSc to make this determination straightforward. PETSc contains the function

```
PetscCitationsRegister(const char bibtexentry[],bool *set)
```

which may be called anywhere in the source code to indicate that a fragment of code or algorithm is being run that is worthy of a specific citation. If the application is run with the `-citations [filename]` option, then at the conclusion of the simulation all indicated bibtex entries are printed and available for the user. The `set` flag is used so that each bibtex item gets recorded only once. Software packages and algorithms that are compiled into the library, but are not used in the simulation do not get listed. For optional external packages to which PETSc interfaces, such as hypre [4], we call `PetscCitationsRegister()` in the "wrapper" function that calls the underlying hypre library, and we cite the publication requested by the hypre developers. `PetscCitationsRegister()` calls can also be incorporated directly into appropriate places in the application code. In addition, if parts of the libraries or application codes are generated by external source code generators, such as Orio [5], these generators can also insert appropriate citations.

We believe approaches such as this should be adopted by the entire open source scientific software community to ensure that full and accurate citations are made for libraries used in scientific applications. The current hit-or-miss approach generally results in many fewer citations than are appropriate for many scientific software libraries. This situation can have detrimental effects on funding, future library development, and even scientific careers.

We have chosen a simple initial implementation of this idea in PETSc. We store the entire bibtex item, as a C character string, for each citation in a linked list that is then traversed and printed at the conclusion of the run. A disadvantage of this strategy is that the publication information is hardwired into the source code and may not get updated upon publication change. An alternative would be to store a "universal identifier" for each citation and either translate that identifier at the conclusion of the run or require the user to translate the identifiers themselves. Because requiring the application simulation code to have access to a database of universal identifiers (in order to do the translation at runtime) poses difficulties, for example, on large batch systems with no network access, we did not adopt that approach. Requiring the user to translate from universal identifiers to the actual bibtex entries presents one more hurdle that needs to be passed for correct citations, and we believe that the fewer hurdles the better.

In addition to registering the bibtex data, it may make sense to register other information—for example, the portion of the source code related to the citation, the change-set of the source code that introduced the cited functionality, or notes on the implementation. We have not explored such possibilities.

We hope this paper opens a discussion into automated efficient ways of ensuring the proper citation of software libraries and algorithms in scientific publications. We conclude with a simple demonstration of this capability in PETSc using two alternative external packages for solving linear systems, hypre [4] and SuperLU [3].

```
~/Src/petsc/src/snes/examples/tutorials  master $ ./ex19 -ksp_monitor -citations -pc_type hypre -pc_hypre_type boomeramg
lid velocity = 0.0625, prandtl # = 1, grashof # = 1
    0 KSP Residual norm 2.465542831974e-01
    1 KSP Residual norm 1.263805353152e-02
    2 KSP Residual norm 1.438657128267e-03
    3 KSP Residual norm 4.175579235181e-05
    4 KSP Residual norm 1.967044070973e-06
    0 KSP Residual norm 2.365268970528e-05
    1 KSP Residual norm 1.296077770265e-06
    2 KSP Residual norm 1.105601409840e-07
    3 KSP Residual norm 1.152875256249e-09
    4 KSP Residual norm 6.113550034643e-11
Number of SNES iterations = 2
If you publish results based on this computation please cite the following:
================================================================================
@TechReport{petsc-user-ref,
  Author = {Satish Balay and Jed Brown and Kris Buschelman and Victor Eijkhout and William D.  Gropp and
           Dinesh Kaushik and Matthew G. Knepley and Lois Curfman McInnes and Barry F. Smith and Hong Zhang},
  Title = {{PETS}c Users Manual},
  Number = {ANL-95/11 - Revision 3.4},
  Institution = {Argonne National Laboratory},
  Year = {2013}
}
@InProceedings{petsc-efficient,
  Author = {Satish Balay and William D. Gropp and Lois Curfman McInnes and Barry F. Smith},
  Title = {Efficient Management of Parallelism in Object Oriented Numerical Software Libraries},
  Booktitle = {Modern Software Tools in Scientific Computing},
  Editor = {E. Arge and A. M. Bruaset and H. P. Langtangen},
  Pages = {163--202},
  Publisher = {Birkh{\"{a}}user Press},
  Year = {1997}
}
@manual{hypre-web-page,
  title  = {{\sl hypre}: High Performance Preconditioners},
  organization = {Lawrence Livermore National Laboratory},
  note  = {\url{http://www.llnl.gov/CASC/hypre/}}
}
================================================================================
~/Src/petsc/src/snes/examples/tutorials  master $ ./ex19 -ksp_monitor -citations -pc_type lu -pc_factor_mat_solver_package superlu
lid velocity = 0.0625, prandtl # = 1, grashof # = 1
    0 KSP Residual norm 2.358581702743e-01
    1 KSP Residual norm 7.147839725241e-17
    0 KSP Residual norm 2.309061316849e-05
    1 KSP Residual norm 2.989519344266e-21
Number of SNES iterations = 2
If you publish results based on this computation please cite the following:
================================================================================
...
@article{superlu99,
  author  = {James W. Demmel and Stanley C. Eisenstat and John R. Gilbert and Xiaoye S. Li and Joseph W. H. Liu},
  title = {A supernodal approach to sparse partial pivoting},
  journal = {SIAM J. Matrix Analysis and Applications},
  year = {1999},
  volume  = {20},
  number = {3},
  pages = {720-755}
}
================================================================================
```

Acknowledgments

The authors were supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research under Contract DE-AC02-06CH11357.

References

[1] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Web page*. `http://www.mcs.anl.gov/petsc`, 2013.

[2] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhauser Press, 1997, pp. 163–202.

[3] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 720–755.

[4] R. Falgout, *hypre Web page*. `http://www.llnl.gov/CASC/hypre`.

[5] A. Hartono, B. Norris, and P. Sadayappan, *Annotation-based empirical performance tuning using Orio*, in Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on, IEEE, 2009, pp. 1–11.

[6] B. Smith, *Encyclopedia of Parallel Computing*, Springer, 2011, ch. PETSc, the Portable, Extensible Toolkit for Scientific computing.