NOMINATION FORM
for the 2008
JENKS PRIZE IN COMPUTER ALGEBRA SOFTWARE ENGINEERING

Deadline for Receipt: May 1, 2008

Send the nominations to jenksprize@sigsam.org

1. Name of Nominator: Robert H. Lewis
2. Nominator's Institution:

   Fordham University, New York NY

3. Nominator's E-mail Address: rlewis@fordham.edu

---

4. Full Name of Nominee (as it should appear on the award plaque): Robert H. Lewis
5. Nominee's Affiliation:

   Fordham University, New York NY

6. Nominee's Postal Address: Department of Mathematics Fordham University 441 E. Fordham Rd. Bronx NY 10458
7. Nominee's E-mail Address: rlewis@fordham.edu

---

8. DESIGN AND IMPLEMENTATION: A bibliography of three to five papers that describe the work for which this person is nominated. If there are no formal publications, please list any other work that is available or can be made available to help the prize committee evaluate the software.

   The Fermat web site, http://home.bway.net/lewis/
   Computer Algebra System Fermat, software and documentation; in the CD accompanying
   "Computer Algebra Handbook," J. Grabmeier, E. Kaltofen, V. Weispfenning (editors),
   Springer-Verlag, 2002.
   Computer Algebra System Fermat, software and documentation; in CD accompanying
   "Computer Programming", July 2002, published by Gruppo Editoriale Infomedia (Italy),
   R.Boni, editor.
   Lewis, Robert H. and Michael Wester, "Comparison of Polynomial-Oriented Computer
   Algebra Systems", SIGSAM Bulletin 33 (4), p. 5-13, Dec. 1999.
   Comparisons with other CA systems: http://www.bway.net/~lewis/fermat/FerTest.html

---

9. DOCUMENTATION: A list of documentation sources for users of the software.

   The Fermat web site http://home.bway.net/lewis/ has manuals:
   http://home.bway.net/lewis/fermat/windm.html
   http://home.bway.net/lewis/fermat/ferman.ps
   Computer Algebra System Fermat, software and documentation; in the CD accompanying
   "Computer Algebra Handbook," J. Grabmeier, E. Kaltofen, V. Weispfenning (editors),
   Springer-Verlag, 2002.

---

10. APPLICATIONS AND IMPACT: A list of publications that cite the use of the software and will help the committee judge its scientific/technical impact. Refereed publications are most helpful, but all will be considered. Other evidence of the impact of the software should be listed and described as well.

    These are in roughly chronological order, most recent first.

[1] Béla Paláncz, Robert H. Lewis, Piroska Zaletnyik, and Joseph Awange, "Computational Study of the 3D Affine Transformation Part I. 3-point Problem." First version online at the Mathematica website, revised version at http://home.bway.net/lewis/affine1.pdf. March 2008.
[2] R. H. Lewis, Comparing Acceleration Techniques for the Dixon and Macaulay Resultants, Proceedings of the ACA 2007 Conference (Applications of Computer Algebra), to appear.
[3] Shaun van Ault, PhD Thesis, Ohio State University, 2008. personal communication March

2008. http://www.math.ohio-state.edu/~ault/

[4] R. H. Lewis, Heuristics to Accelerate the Dixon Resultant, Mathematics and Computers in Simulation 77, Issue 4, p. 400-407, April 2008.

[5] Nazar Khan, "Silhouette-Based 2D-3D Pose Estimation Using Implicit Algebraic Surfaces", Master Thesis, Saarland University, Germany, 2007. advisor: Bodo Rosenhahn.

[6] Lewis, Robert H. and E. A. Coutsias, "Algorithmic Search for Flexibility using Resultants of Polynomial Systems." Proceedings of the ADG 2006 Conference (Automatic Deduction in Geometry), Pontevedra Spain. Lecture Notes in Computer Science, Vol. 4869, p. 68 - 79. Botana, F., Recio, T. (Eds.) Springer, Berlin, 2007.

[7] Pearson, Jane M., University of Wales, Aberystwyth. personal communications 2006 – 2008.

[8] Bozoki, Sandor and Robert H. Lewis, "Solving the Least Squares Method Problem in the AHP for 3 x 3 and 4 x 4 Matrices," Central European Journal for Operations Research **13** p. 255 – 270, September 2005.

[9] K.G. Chetyrkin, M. Faisst, C. Sturm, and M. Tentyukov. e-Finite Basis of Master Integrals for the Integration-By-Parts Method. January 2006. 28pp. http://arxiv.org/pdf/hep-ph/0601165

[10] M. Czakon. "The Four-loop QCD Beta-Function and Anomalous Dimensions." DESY-04-223, SFB-CPP-04-62, Nov 2004. 14pp. Published in Nucl.Phys.B 710:485-498, 2005. http://arxiv.org/pdf/hep-ph/0411261

[11] M. Czakon, J. Gluza, T. Riemann. "Master Integrals for Massive Two-Loop BHABHA Scattering in QED." DESY-04-222, SFB-CPP-04-61, Dec 2004. 21pp. Published in Phys.Rev.D71:073009, 2005. http://arxiv.org/pdf/hep-ph/0412164

[12] Lewis, Robert H. Top secret mathematical topic, Internal publication, Institute for Defense Analysis. 18 pages, April 2005.

[13] Yuen, David S. Siegel modular cusp forms (2004). personal communication.

[14] Lewis, Robert H., Invited One hour lecture and demonstration, "Computer Algebra System Fermat," at the conference: Computer Algebra for the Working Mathematician, City College of New York, May 15 - 16, 2003.

[15] Little, John. "Solving the Selesnick-Burrus Filter Design Equations Using Computational Algebra and Algebraic Geometry", Advances in Applied Mathematics, **31** (2003), p. 463-500.

[16] Lewis, Robert H. and Stephen Bridgett, "Conic Tangency Equations and Apollonius Problems in Biochemistry and Pharmacology," Mathematics and Computers in Simulation 61(2) (2003) p. 101-114.

[17] Lewis, Robert H. and Peter F. Stiller, "Solving the recognition problem for six lines using the Dixon resultant," Mathematics and Computers in Simulation 49 (1999) p. 205-219.

[18] Lewis, Robert H. and Michael Wester, "Comparison of Polynomial-Oriented Computer Algebra Systems", SIGSAM Bulletin 33 (4), p. 5-13, Dec. 1999.

[19] Lewis, Robert H. "The Six Line Problem and Resultants," presented to the "Grand Challenges" session at IMACS, Hawaii, July 1997.

[20] Lewis, Robert H. and Guy D. Moore. "Computer Search for Nilpotent Complexes," Experimental Mathematics **6**:3 (1997) p. 239-246.

[21] Brumer, Armand. "The Rank of Jo(N)," Asterisque **228** (1995) p. 41-68.

[22] Lewis, Robert H. and Sal Liriano. "Isomorphism Classes and Derived Series of Almost-Free Groups," Experimental Mathematics **3** (1994) p.255-258.

11. A 1-3 page description of the nominee's accomplishments in software engineering applied to computer algebra and an assessment of the importance and impact of the work for which this person or team is nominated.

Computer Algebra System Fermat

References [1] - [22] are above, [23] - [36] are below.

Fermat is an interactive system for mathematical experimentation. It is a super calculator-computer algebra system, in which items being computed can be integers (of arbitrary size), rational numbers, real numbers, complex numbers, modular numbers, finite field elements, multivariable polynomials, rational functions, or polynomials modulo other polynomials. The main areas of application are multivariate rational function arithmetic and matrix algebra over rings of multivariate polynomials or rational functions. Fermat does not do simplification of transcendental functions or symbolic integration.

A session with Fermat usually starts by choosing rational or modular "mode" to establish the *ground field* (or *ground ring*) F as Z or Z/n. On top of this may be attached any number of symbolic variables $t_1, t_2, ..., t_n$, thereby creating the polynomial ring $F[t_1, t_2, ..., t_n]$ and its quotient field. Further, some polynomials p, q,... involving some of the t_i can be chosen to mod out with, creating the quotient ring $F(t_1, t_2, ...) \ / < p, q, ... >$. Finally, it is possible to allow *Laurent polynomials*, those with negative as well as positive exponents. Once the computational ring is established in this way, all computations are of elements of this ring. The computational ring can be changed later in the session.

In an earlier version, called FFermat, the basic number type is real (or complex) numbers or "floats" of 18 digits. That version allows for numerical computing techniques, has extensive graphics capabilities, no sophisticated polynomial g.c.d.

algorithms, and is available only for Mac OS.

Fermat runs on Mac OSX, OS9, Linux, Unix, and Windows95/98/Me/XP etc. Fermat was originally written in Pascal for a DEC Vax, then for Mac OS during 1985 - 1996. It was ported to Windows in 1998. In 2003 it was translated into C and ported to Linux (Intel machines) and Unix (Sparc/Sun). It is about 91000 lines of C code. The polynomial g.c.d. procedures, which call each other in a highly recursive manner, are about 8000 lines.

Fermat has extensive built-in primitives for array and matrix manipulations, such as submatrix, sparse matrix, determinant, normalize, column reduce, row echelon, Smith form, and matrix inverse. It is consistently faster than some well known computer algebra systems, especially in multivariate polynomial g.c.d. It is also space efficient. Comparisons [28] show it to use much less space than other systems.

The basic data item in Fermat is a multivariate rational function or *quolynomial*. The numerator and denominator are polynomials with no common factor. Polynomials are implemented recursively as general linked lists, unlike some systems that implement polynomials as lists of monomials. To implement (most) finite fields, the user finds an irreducible monic polynomial in a symbolic variable, say $p(t_1)$, and commands Fermat to mod out by it. This may be continued recursively, $q(t_2, t_1)$, etc. Low level data structures are set up to facilitate arithmetic and g.c.d. over this newly created ground field. Two special fields, $GF(2^8)$ and $GF(2^{16})$, are more efficiently implemented at the bit level.

To help implement the Dixon resultant technique [25], special features have been added to the determinant function [4], [27]. These provide a dramatic increase in the speed of resultant calculations with systems of polynomial equations that exhibit symmetry. Several experienced researchers have told me that Grobner basis techniques cannot take advantage of symmetry.

Fermat provides a complete programming language. Programs and data can be saved to an ordinary text file that can be examined as such, read during a later session, or read by some other software system.

I envision the users of Fermat to be rather sophisticated in both mathematics and programming. (However, one needn't do *any* programming to use Fermat.) At many places in the design and implementation of Fermat I had to balance the conflicting goals of flexibility and safety. That is, whether to allow the user certain freedoms or language features that might perhaps be abused, or to circumscribe the user in the name of safety. Since I regard the users as sophisticated, I have usually chosen freedom. It is easy to interrupt a long computation, examine data, save to a file, make changes, and resume the computation. In this and other ways Fermat is very user-friendly.

Fermat is shareware. Executable binaries and manuals can be downloaded from the web site, [28]. Pages there compare Fermat to other CA systems on tests such as determinant, rational function arithmetic, and g.c.d. of multivariate polynomials [31]. Some of these tests have been conducted by independent researchers [30].

Fermat has been crucial to the success of several projects. The authors were unable to make progress with large well known computer algebra systems. [20], [21] and [22] were pure mathematics applications in which matrix normalizations or characteristic polynomial were important. [7], [8], [15], [16], and [17] needed to solve systems of polynomial equations, and used the Dixon resultant method [25] with Fermat. [9], [10] and [11] use Fermat's very efficient rational function arithmetic. [13] used Fermat's row reduction of integer matrices.

More information is available on my web site [28].

History and Summary

Following from my PhD (Cornell University) in algebraic topology, and my MS in computer science (University of North Carolina, Chapel Hill) I was by 1981 exploring what would later be called "experimental mathematics." I wrote computer programs to compute the homology groups of CW complexes, and the homology modules (not just groups) of their universal covers. I used these early programs (circa 1982), written in Pascal, to search for the "simplest" low dimensional nilpotent complexes.

By 1985 I realized that Pascal and all other high level languages were inadequate to the task. Serendipitously having just finished teaching a computer science course in programming language design, I began writing a software system I would soon name "Fermat." From 1988 - 1992 I developed and used Fermat during three National Science Foundation R.E.U. projects that I ran. Each of these involved a summer program for undergraduates who worked on the computer search for three dimensional nilpotent complexes. The search was a success. This project is summarized in the 1997 paper [20] with Guy Moore. The use of Fermat was crucial, as we found it to be much faster than Maple in Smith form and column normalization, the key steps in the computation of homology.

I attended the seminar in computational group theory run by Gilbert Baumslag in the fall of 1990. This successful - thanks to Fermat - project yielded the 1994 paper with Sal Liriano [22] in which we settled a conjecture of Baumslag.

During 1996 - 1998 I was employed as a computer algebra consultant by the Department of Naval Research. There I was introduced to systems of polynomial equations and resultants. I helped to solve a significant problem in computer image analysis, called the "six line problem" [17]. Fermat was instrumental in this solution. Apparently, until the year 2000, no other system could work out one crucial step, and as far as I know Fermat remains by far the fastest system on these computations with multivariate rational functions [31].

In 2001-2002 I worked with John Little of Holy Cross and Ivan Selesnick of Polytechnic University, Brooklyn, on signal processing problems and wavelet design. Once again, Fermat could solve systems of polynomial equations that no other system could touch. Little published this in [15].

In 2002 I worked with a colleague in the U. K. on a computational chemistry project that falls under the general category of "rational drug design." One approach is to compute the medial axis or surface (a sort of skeleton) of the space around molecules. This leads to Apollonius-type sphere, ellipsoid, and line problems, in which one wants to compute the polynomial equations that various parameters of a tangent sphere must satisfy[16]. Apparently, only Fermat, using a variation of the Dixon resultant technique, can solve equations of this complexity.

In 2002-2003 I worked with S. Bozoki to solve a problem in decision theory [8]. Once again, this involved solving a system of polynomial equations with the combination Fermat-Dixon.

From 2002-2005 I was a consultant at the Center for Computing Sciences (a part of the Institute for Defense Analysis). We produced a Unix/Linux version of Fermat written in C, and have worked on various mathematical problems [12].

Since 2003, physicists Oleg Tarasov and Michael Czakon have used Fermat to simplify large rational functions in nine or more polynomial variables. The ASCII file containing their first examples occupied 258K. Fermat simplified them completely to 84K. They reported to me that Mathematica crashed trying to do this and Maple went on for 8 hours before they killed it. Fermat did it in 7 seconds. See [10], [11]. They continue to use Fermat well into 2008.

In 2004 David S. Yuen [13] of Lake Forest College used Fermat to help prove a theorem in Siegel modular cusp forms. He needed to row-reduce 2000 by 2500 integer matrices where the size of the entries range form 1 to 200 digits. On February 25 he posted to sci.math.symbolic: "Of course, as soon as I try RowReduce in Mathematica, my computer would just run out of memory. Yes, Fermat was able to do this row reduction. Fermat was also great in that it gave progress reports on how far along the row reduction it was (this was indispensable so that I knew when to abort hopelessly long calculations)." This is a documentable example of the user-friendliness of Fermat. Yuen's work with Fermat is continuing with his colleague Cris Poor [29], [35].

From 2006 – 2008 Jane Pearson and her colleagues in Wales have used Fermat to investigate non-linear polynomial systems of ordinary differential equations, and the related problem of limit cycles bifurcated from the critical point [7].

From 2005 – 2008 Evangelos Coutsias and I have used Fermat to investigate computational chemistry. We are extending ideas of [24] looking for flexible molecules [26], [6]. Recently the renowned computational chemist Michael Barnett has expressed interest to me [23] in resultant techniques as an alternative to Grobner bases.

In 2007 Fermat was used in part of Nazar Kahn's Masters Thesis [5], on a computer vision problem somewhat similar to [16].

In 2007 I used Fermat's builtin features, not found in any other system that I am aware of, to analyze the Macaulay resultant in new ways [2].

In 2008 Béla Paláncz and his colleagues in Hungary found that Mathematica's Dixon resultant routines were inadequate to solve a system of polynomial equations arising in Global Positioning System theory. Fermat easily solved the system [1].

In 2008 Shaun van Ault found that Fermat's Smith normal form routines were far superior to those of the system he had been using, GAP [3].

In 2008, algebraic topologist Mark Behrens [32] wrote "I enjoy using your program to perform informal calculations in certain Hopf algebras. The Hopf algebra in this posting is used to compute the E_2 term of a spectral sequence that computed the stable homotopy groups of spheres. I very much enjoy your program for its ease of use and lack of hubris!"

Through 2008, Alexander Smirnov and M. Tentioukov, Institute for Theoretical Particle Physics, Karlsruhe, have used Fermat as a fundamental part of their software. See especially FiRE and FLink. They write [33]: "Fermat is extremely fast in working with polynomials. One of the most powerful features of Fermat is the fast evaluation of the GCD - the greatest common divisor." See also the paper on Feynman integral reduction [34].

J. H. Kuhn, M. Steinhauser and M. Tentyukov used Fermat in their particle physics research [36].

Assessment

Admittedly, this may be hard for me to do. Clearly Fermat is greatly superior to Maple, Mathematica, and GAP in polynomial and rational function arithmetic [28], and matrix normal forms [13], [3]. It is approached, as far as I know, only by Magma, which is very expensive. Fermat is essentially free. See [18] for comparisons with Singular, Magma, CoCoA, MuPad, and Pari-GP. As far as I know, using Fermat-Dixon I can solve systems of polynomial equations that no one else can solve. I am aware that the designers of other systems have been influenced by Fermat, such as David Saunders, and the designers of Magma. The growing list (above) of users and collaborators since 2002 shows the accelerating interest in Fermat.

As of this date, Fermat has about 150 registered users. No doubt there are more who have not bothered to register (it is not necessary to do so to download the software). Because Fermat is a specialized system, it obviously has nowhere near the user base of the large (and expensive) systems like Maple, Mathematica, and Magma. Comparing it to other small systems, like CoCoA and Singular that are designed for Grobner bases, it would seem that Fermat is much better at a much larger list of capabilities. For example, I have not tried Singular recently, but Singular used to be terrible at multivariate rational function arithmetic [18]. NTL does only univariate polynomials, and, quoting from its web site, "NTL has no pretensions about being an interactive computer algebra system: it is a library for programmers." I am not aware that any of these other small systems is very good at Smith form or column reduction.

It would seem that Fermat is not approached by any other system for the combination of price, features, and performance.

[23] Michael Barnett, http://www.princeton.edu/~allengrp/ms/ personal communication 2006 -

2008.
[24] E.A. Coutsias, C. Seok, M.J. Wester and K.A. Dill. Resultants and Loop Closure. International Journal of Quantum Chemistry, 106(1), 176-189, 2005.
[25] D. Kapur, T. Saxena, and L. Yang, Algebraic and geometric reasoning using Dixon resultants. In: Proc. of the International Symposium on Symbolic and Algebraic Computation. A.C.M. Press (1994).
[26] Robert H. Lewis. Algorithmic Search for Flexibility Using Resultants of Polynomial Systems. ECCAD 2006, Drexel University.
[27] Robert H. Lewis. Exploiting Symmetry with the Dixon Resultant, ACA 2004, Lamar University, July 21 - 23, 2004.
[28] Fermat web site, http://home.bway.net/lewis/
[29] Cris Poor, personal communication, March 2006, March 2007.
[30] Fermat is overall best at polynomial gcd, according to two independent researchers. D. Robertz and V. Gerdt. http://home.bway.net/lewis/fermat/gcdcomp
[31] http://home.bway.net/lewis/fermat/FerTest.html
[32] Mark Behrens, http://www-math.mit.edu/~mbehrens/. April 29, 2008. ALGTOP-L mail list. Personal communication.
[33] A. Smirnov, http://www-ttp.particle.uni-karlsruhe.de/~asmirnov/
[34] A. Smirnov, http://arxiv.org/pdf/0807.3243v3 Algorithm FIRE — Feynman Integral REduction.
[35] M. Oura, C. Poor, and D. S. Yuen, "Towards the Siegel Ring in Genus Four." International Journal of Number Theory, August 2008.
[36] J. H. Kuhn, M. Steinhauser and M. Tentyukov, "Massless Four-Loop Integrals and the Total Cross Section in e+ e- Annihilation." High Performance Computing in Science and Engineering '07; Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2008.

12. A list of three-five persons who are being asked to write letters in support of this nomination. It is the nominator's responsibility to contact these persons and see to it that the letters are received by the committee by the nomination deadline of May 1.

Evangelos A. Coutsias
Jane Pearson
David Yuen
Michal Czakon
Michael Wester
Bela Palancz
Manfred Minimair
Erich Kaltofen

This form is meant to be a guide for nominators. The information that it requests should be supplied as completely, accurately, and effectively (i.e., effective in support of the nomination) as possible. However, the form is not intended to preclude the inclusion of other relevant information. The impact of the software is of particular importance and the nomination packet should clearly indicate that to the committee. Careful preparation of strong nomination packets is important.