

COMPUTATIONAL OPPORTUNITIES IN ALGEBRA, NUMBER THEORY AND COMBINATORICS

ABSTRACT

The following report was prepared by a group of 22 research mathematicians whose core areas are in Algebra, Number Theory, and Combinatorics (ANTC). These mathematicians met at the NSF headquarters in Arlington, VA in September, 2002 to discuss various issues having to do with computation in ANTC. This report is a compilation of our thoughts, discussions, and recommendations.

The sections in this report are as follows:

1. Introduction: the prevalence of ANTC in areas of theoretical and practical importance, and an introduction to the prominence of computation in ANTC issues.
2. ANTC problems with a computational component: theoretical problems in which computation is likely to have a significant impact.
3. Hardware: platforms which can be of use to ANTC.
4. Software: issues concerning the specialized software used by ANTC.
5. Web database: a repository for the useful products of computations.
6. Computers and proof: computers in the creation and verification of proofs.
7. Education and outreach: ANTC as a route to appreciating and doing higher mathematics.
8. Culture change: some current practices are contrary to the best interests for the future of mathematics.
9. Conclusion.
10. Recommendations.

We end with appendices listing current software packages, describing William Stein's computational cluster, proposing a vertically integrated summer research program for undergraduate students, and listing the authors of this document.

1. INTRODUCTION

Computation is a significant component both in ANTC research and in the applications of ANTC research to practical problems. We outline the role of computation in theoretical results, and then describe several real-world applications.

1.1. Computation in ANTC Research. Computation plays a significant role in the discovery of new results. Throughout most of the history of mathematics this involved calculation by hand. For example, Gauss conjectured the distribution of prime numbers by looking at a table of primes. The proof that his conjecture was correct was not completed until 100 years later, and that proof was a driving force behind the early development of complex analysis.

More recently, calculation involves a computer. Important conjectures, such as the Birch and Swinnerton-Dyer (BS-D) conjecture, were discovered from numerical data. The BS-D conjecture is a motivation for much current research, and this would not be possible without the computations that led to the conjecture.

Even more recently, computation has played a role in the proofs of mathematical theorems. A significant early example is the proof of the 4 color theorem (which has now been proven in several independent ways, all of which involve a computer). A more recent example is Hales' proof of the Kepler sphere packing conjecture.

The common thread here is that the traditional goals of progress in pure mathematics have been enhanced by the availability of computers. Below we list some of the ways in which computation plays a role in ANTC research. These are intended merely to hint at the variety of ways in which computation enters the picture; some of these examples are elaborated upon in the body of this document.

1.1.1. *Computational Tools.* Mathematicians use a great variety of approaches in their computations. At one end of the spectrum is researcher-written programs in a programming language such as Fortran or C. At the other end on the spectrum is highly specialized software which has built-in commands that perform high-level calculations. Examples are Macaulay (algebraic geometry), Pari (algebraic number theory) and GAP (group theory). In the middle are powerful general purpose packages such as Mathematica, Maple, and Magma.

1.1.2. *Discovery.* Many results have been discovered with the help of a computation, and later proven by traditional means. An illustrative example is Wilf's paper "Computer-Aided Discovery of a Theorem about Young Tableaux." A somewhat deeper example is the work of Eisenbud and Schreyer on the Bernstein-Gelfand-Gelfand correspondence. They give a completely concrete description of this correspondence, and a key step in their work was proving the general case of certain patterns which appeared in sheaf cohomology calculations they performed with the Macaulay 2 package.

1.1.3. *Confirmation and exploration.* Investigating examples is a fundamental technique for coming to grips with a difficult mathematical problem, and exploring new cases of a conjecture can lead to clues for understanding the general case. For instance, the Langlands program is a far-reaching series of conjectures linking algebra and analysis. Verifying special cases can be quite difficult and can lead to serious mathematics. Examples are Ash and McConnell's work on 3-dimensional Galois representations, and work of Sarnak's students on functional equations for L-functions of varieties.

A well-known example in this category is the conjecture of Birch and Swinnerton-Dyer on the ranks of elliptic curve. Another example involving elliptic curves is Bloch's conjecture that the rank of $K_2(E)$ is 1, where E is an elliptic curve over the rationals. Numerical calculations by Bloch and Grayson revealed that the conjecture had to be modified, and further calculation supports the current version of the conjecture which uses the Neron model of E rather than E itself.

1.1.4. *Useful databases of mathematical objects.* Enumeration of important classes of structures impacts research at a number of levels. It tests the power of existing theory; the data itself provides new insights into the structures, and elucidates their properties. The challenges in successfully completing the enumeration often lead to new theoretical and algorithmic breakthroughs, both mathematical and in the theory of databases.

For example, groups of prime power order have special importance since in a technical sense “most groups have prime power order”. Also all finite groups have Sylow subgroups of such orders which are critical to understanding their structure. Before modern computers were used for mathematical investigations, the largest (in format) book in many mathematics libraries was a book by Hall and Senior listing all groups of order 64 - a famous and difficult computation, regularly used for the information it contained. Now, through work of Havas, Newman, O’Brien, and Eick, we have available electronic databases of groups of all orders up to 1,000. Several computational and theoretical problems had to be overcome to achieve these classifications: how to guarantee all such groups were found, and distinguished up to isomorphism, how to ensure the accuracy of the lists, how to store and present the lists in a usable way.

Recently, Sturmfels and his collaborators have produced much valuable work in the emerging field of algebraic statistics, and also in Gröbner bases of toric varieties, which has applications to integer programming. Explicit examples, such as minimal generators of the toric ideal for various statistical models, will serve as a valuable resource.

1.1.5. *Changing the paradigm of existing computation.* Sims constructed what is now called the Lyons-Sims sporadic simple group in the early 1970s. This was the only way to prove existence, and it also proved uniqueness of a simple group with a list of properties obtained theoretically by Richard Lyons. So the impact of the actual construction was huge. The computational challenge for Sims was enormous: the group was acting on a set of size 8835156; it was not even possible to store two permutations of this size (as integer strings of this length), let alone multiply them together. The new theory developed to perform this computation led to the now-familiar concepts of base and strong generating set, that form the basis of all modern permutation group algorithms.

A similar endeavor was the construction of the sporadic group J_4 by Benson et al. The group was constructed on the computer as 112 by 112 matrices over the field with two elements, and the proof that the group constructed was actually J_4 was an intricate combination of computer algebra and abstract group theory.

1.1.6. *Proofs and counterexamples.* In addition to the computer assisted proofs described above, computation also plays a crucial role in finding counterexamples to conjectures. Examples are Elkies’ disproof of the Euler conjecture (on four 4th powers) and the disproof of the Mertens conjecture. More recent examples are the counterexample of Bruns-Gubeladze to the Hilbert cover problem of polyhedral cones, and De Loera-McAllister’s infinite family of counterexamples to the integrality conjecture for the coordinates of the vertices of Gelfand-Tsetlin polytopes.

Computer calculations can also be used to show negative results, such as the non-existence of a finite projective plane of order 10.

1.2. **ANTC Applications.** It is hard to overstate the importance of mathematics in the current and ongoing scientific revolution. Mathematics is the language of science, enabling the quantification, understanding, and exploitation of natural phenomena and ideas. Today, much of this language requires translation to the domain of computers in order to take advantage of the power of computation. Science today is digital and computational: biology

has been forever changed by genomics, medical diagnostics rely on an ever-increasing number of digital imaging techniques, neuroscience is being transformed by neuroinformatics, financial markets rise and fall on the analysis of digital time series, even sociology is being re-imagined by new network models. The list goes on and on, rife with subjects that are of importance for the security and intellectual and physical health of the nation.

All of this quantized and digitized information rides on the zeros and ones that are collected into the bits and bytes that stream through the Internet. It is due to the discrete nature of digital information that so much of the mathematics that has enabled these advances comes from the disciplines of Algebra, Number Theory and Combinatorics (ANTC). Generally speaking, algebra studies the implications and possibilities for various rules of symbol manipulations, including the familiar x 's and y 's of high school equation solving, as well as reams of digital data. Number theory investigates the laws and relations of numbers and their generalizations. Combinatorics studies the underlying regularity of complicated patterns, permitting the enumeration and subsequent organization of huge collections of objects, like the A's, C's, G's and T's of the genome.

For many important problems, we have long ago exhausted what can be done by taking a naive algorithm and running it on faster computers. Progress now requires increasingly efficient algorithms that depend on sophisticated mathematics, often derived from ANTC research:

Communications: Algebra and number theory are responsible for the communications protocols that ensure reliable and secure communication. These are the disciplines of coding theory and cryptography.

Coding theory develops error correcting codes, a means of encoding a digital message so that should its information be disturbed (either maliciously or naturally) the original message remains interpretable. Finite field arithmetic is responsible for the most widely used codes (Golay, Hamming, etc.). More recent developments of low density parity check codes use new algebraic tools from group theory and graph theory. New quantum codes rely on a marriage of number theory, topology, and algebra, as does Shor's quantum factoring algorithm.

Public-key cryptography is a triumph of number theory. The RSA cryptosystem is now the industry standard, but advances in computing power (including potential quantum computing) as well as requirements for more constrained environments (like palmtops) require systems employing more advanced algebra, in particular elliptic curve cryptosystems and their potential successor, hyperelliptic curve cryptosystems, as well as even more elaborate cohomological techniques.

Finance: Another novel use of algebra has recently arisen in applications to finance. Mortgage companies often trade large portfolios of mortgages. Their need to quickly and accurately compute a fair price for the collection requires numerical computation of integrals. Wall Street now uses optimal sample points generated using techniques from elementary number theory, algebraic curves over finite fields, and even class field theory of function fields.

Antenna Configurations: Recent advances have discovered a novel relation between good antenna configurations and linear algebra: configurations are equivalent to families of unitary matrices and their performance is assessed in terms of the determinants of differences between

these matrices. Industry has already used families derived from the algebraic structure of group representations, in which case the performance can be read directly from the character table.

Digital Signal Processing: The Fast Fourier Transform is an algorithm that underlies most of applied Fourier analysis, efficiently transforming temporal or spatial data into frequency data. Algebraic approaches relying on group cohomology theory (due to Auslander, Johnson and Johnson) have made possible the optimal implementation of these algorithms on a variety of platforms, an application that has proved to be of paramount importance in current national security applications.

Large-scale optimization and search techniques: A marriage of combinatorics, algebra and geometry are responsible for some of the most important techniques for finding optimal or near-optimal solutions to problems of resource allocation in the face of huge numbers of constraints. Thousand-city-tour instances of the traveling salesman problem, once thought impossible, are now solved routinely. This is a paradigm for the sort of logistical optimization faced by organizations like airlines. Recent advances in the very general tools of integer programming (one of the most oft-used optimization techniques) make use of developments in graph theory and combinatorics.

Additionally, combinatorial search techniques have helped make possible the real-time prospecting of the large databases of molecules and proteins, as well as the huge configuration spaces that arise in robotic motion planning.

Symbolic computation: ANTC is a direct contributor to the algorithms used in commercially available symbolic computation packages like Mathematica and Matlab, upon which the engineering community is heavily reliant. An example is “Analog Insydes”, an add-on package to Mathematica which is a tool for analysis and design of analog microelectronic circuits based on symbolic methods from algebra and combinatorics.

These packages enable rapid software and simulation prototyping as well as visualization tools so necessary for subjects like computational chemistry.

1.3. ANTC and Computation. ANTC not only enables computation, but is also a heavy user of computation as a tool for basic research. Symbolic manipulation packages and efficient computational platforms are the laboratories of many modern ANTC researchers, enabling the search for patterns in huge collections of complicated examples.

This is at the heart of the general fact that research in mathematics — and ANTC in particular — pushes technological development. Better mathematics gives better algorithms, faster computers, and more reliable communication. In particular, as many of our examples show, the new technology then requires new mathematics for its own exploitation and security. The symbiosis and feedback are inescapable.

ANTC researchers will make use of increasingly sophisticated computational tools as well as contribute to their development. In what follows we discuss what appears to be the issues on the horizon for this important area of research.

2. ANTC PROBLEMS WITH A COMPUTATIONAL COMPONENT

The spectacular increase in available computing power over the last 20 years or so has significantly increased the importance of computation as a mathematical research tool. Mathematicians use this power to experimentally detect previously invisible patterns and to complement “pure thought” in proving new theorems; the algorithms used are themselves fertile topics for mathematical research.

Much of the exciting new work in ANTC has an essential computational component. The excitement stems from the work’s intrinsic importance within mathematics as well as its direct and potential applications and interdisciplinary connections. By serving as a kind of laboratory (or test basin), computers can facilitate vital synergies among theory, experiment, and application. We see this blurring of boundaries as a healthy trend in ANTC, a trend destined to grow significantly in future years.

Though there is no crystal ball foretelling future applications of mathematics, a look at “mathematics in the making” – particularly when that has strong computational underpinnings – can often provide useful hints. We detail four examples drawn from diverse areas of ANTC, and conclude this section by illustrating the broad impact of modern computation in ANTC with a list of other areas that benefit considerably from computational power.

(1) Cryptographically applicable ANTC: factorization, discrete logarithms, and lattice reduction

Primes and prime factorization are one of the most venerable areas of ANTC and of all of mathematics, and continue to inspire extensive and fruitful research into both theory and computational practice. Over the past few decades, this research has gained new impetus from applications to cryptology, where large primes can be used to achieve such paradoxical-seeming tasks as public-key cryptography and secret sharing over a public communications channel. More recently, alternative approaches to these tasks have become available, but those too use mathematical structures from ANTC, such as elliptic curves, Jacobians of algebraic curves of higher genus, and high-dimensional lattices. Efficient algorithms for the computational problems posed by these applications now constitute a nexus of sophisticated mathematical techniques, algorithmic ingenuity, and practical applications with growing benefits to all three viewpoints.

(2) Zeta functions and related issues: Riemann Hypotheses, quantum chaos, and matrix models

There is constant need for efficient algorithms to compute zeros and values of the myriad of zeta and L-functions which arise in algebraic number theory, analytic number theory, and automorphic representations. Extensive numerical evidence now bolsters a generalized Riemann Hypothesis (RH) for the zeros of many such functions, a hypothesis whose proof promises revolutionary structural insights (as with the Weil-Deligne proof of RH for function fields) as well as concrete applications (such as primality testing which is both provably correct and faster than the recent AKS algorithm). Moreover, the numerical evidence hints at a deep connection with random matrix models in connection with the distribution of zeros. This promises a new and fertile interaction between several areas of ANTC, and has already led to advances in the theory of error-correcting codes and expander graphs. Recent work also connects this investigation with quantum unique ergodicity on hyperbolic surfaces

of arithmetic type, which again is amenable in part at least to computational exploration. The algorithms used to compute eigenvalues and eigenfunctions in these settings, which were developed to check agreement with various random matrix or wave models, also have potential or even immediate application to work on microwave cavities, semiconductors, and cosmology.

(3) Computational algebra: Groebner bases and applications

The theory of Groebner bases is a key computational tool in modern commutative algebra and algebraic geometry, the modern mathematical disciplines developed to solve classical problems such as solving simultaneous polynomial equations. (Such problems remain important not only in algebra but also in combinatorics – and even in such applications as designing robot-arm mechanisms!) Almost all computer algebra packages available today have an implementation of Groebner bases and many algorithms require the computation of a Groebner basis. This ubiquitous tool still holds many theoretical and practical secrets. The length of the computation can be extremely sensitive even to the order of the variables, and the art of choosing a good order has yet to be reduced to a reliable algorithm. Recent work in combinatorics also points to a deep connection between the theory of Groebner bases and certain NP-complete problems, suggesting that a better understanding of Groebner bases may even yield progress on the fundamental problem “P=NP?” of computational complexity theory.

(4) Combinatorial structures: Extended ATLASes of groups and their representations, combinatorial graphs and designs, etc.

Combinatorics is a natural source of problems and theories in which “pure thought” and computation combine synergistically to produce results beyond the reach of either approach on its own. One important class of such problems is the classification and study of highly regular structures. Some important examples are finite geometries, regular graphs, finite projective planes and more general designs, and error-correcting codes. Some of these classes of structure also have direct applications in computer science (expander graphs, error correction, etc.). In each case, continued progress is sure to rely on applying more computational insight and power to the problem. Another justly celebrated example of a highly regular structure is a finite simple group; while such groups have at last been classified, there is still much critical information about their structure, representations, etc. that remains to be elucidated, and is needed for further progress both in group theory and in other disciplines such as finite geometries and the topology of classifying spaces.

Further questions whose study can be expected to benefit greatly from computation include:

- The solution of general Diophantine equations;
- The arithmetic of modular forms and Galois representations;
- The study of Ehrhart polynomials and related formulas for counting integer points in polyhedra;
- The “true circuit conjecture” in toric algebra;
- Projective planes of non-prime-power order;
- Combinatorial topology;
- Unimodality conjectures in combinatorics;
- Zariski’s multiplicity conjecture;
- Applications of combinatorial game theory;

- Topological algorithms for graph theory;
- And various problems in algebraic geometry, such as:
 - The computational complexity of linear algebra,
 - The Eisenbud-Goto conjecture,
 - Green’s conjecture on canonical curves,
 - Incidence geometry (hyperplane arrangements and generalizations).

3. HARDWARE

Computers are tools of mathematical exploration and proof. Bigger, faster computers have changed the way mathematicians work. The need to make use of the capabilities of available computers, by both mathematics and the other sciences, has led to the development of new theorems, new algorithms, and new proof techniques.

Personal computers have changed mathematics in every way. What new mathematics would emerge if we had easy access to parallel computers? Modern, state-of-the-art parallel computers could be a fantastic resource for the ANTC community. These parallel machines have become much more affordable and maintainable in the last few years. Funding sources for a fast parallel network resource include FRG (Focused Research Group), Information Technology Research (ITR), Major Research Instrumentation (MRI), Academic Research Instrumentation (ARI), Scientific Computing Research Environments for the Mathematical Sciences (SCREMS).

There are two types of parallel machines based on readily available hardware. First, the easiest parallel computer to build is n PC’s linked together. William Stein has successfully accomplished this on a shoestring budget. This is not a true parallel computer because the 12 processors cannot communicate if they need to share information. However, if a computation is trivially parallelizable then it can be broken down into many subcomputations and run n times faster than on a single processor.

Second, if one does true parallel computation, one can convert n PC’s to a parallel machine under Linux, using code already created in the Beowulf project (see www.beowulf.org). Here is a simple description of a Beowulf:

“It’s a kind of high-performance massively parallel computer built primarily out of commodity hardware components, running a free-software operating system like Linux or FreeBSD, interconnected by a private high-speed network. It consists of a cluster of PCs or workstations dedicated to running high-performance computing tasks. The nodes in the cluster don’t sit on people’s desks; they are dedicated to running cluster jobs. It is usually connected to the outside world through only a single node. Some Linux clusters are built for reliability instead of speed. These are not Beowulfs.” [<http://www.canonical.org/~kragen/beowulf-faq.txt>]

There do exist national supercomputing facilities, but to date they are not well used by the ANTC community. These centers have historically been dominated by physics experiments. Mathematical software is not widely available, and it is non-trivial to write efficient parallel code. We would like to see the supercomputer centers more accessible to mathematicians in ANTC. At a minimum this requires support from systems experts. Some ANTC researchers have used supercomputers in their research (for example, Odlyzko and Hejhal), but at present

it is not common for mathematicians, even mathematicians who do extensive computations, to make use of these resources.

Funding for computer classrooms for teaching hands-on mathematical computation would be very beneficial. Computer cluster/classrooms have proven to be excellent teaching environments. Often departments hesitate to initiate new facilities but they are willing to maintain them. NSF sponsored computer clusters could revolutionize the way we teach mathematics. Funding sources for computer/classroom clusters could be through EHR (Education and Human Resources) or SCREMS.

4. SOFTWARE

Mathematical software is a major tool of the trade in modern mathematics. It is a means by which mathematicians can efficiently perform new computations without coding everything from scratch. This allows one to build on previous work, and to contribute to the collective body of knowledge. The development and maintenance of software should be a high priority for the ANTC community and funded at a high level by the NSF.

One concern is that most of the highest level computer algebra packages containing specialized tools for algebra and number theory are developed outside of the United States. The United States dominates in the development of operating systems and of software in other areas, but among the most widely used systems for high level algebra and number theory, only Macaulay2 is an American product. Magma, in particular, which is used extensively by security agencies both in and out of the US, is developed in Australia. While many American mathematicians contribute to foreign systems such as Magma, Pari and GAP, the major development is abroad. The situation may be partly due to the current atmosphere in the ANTC research community, which undervalues the importance of this activity. This is in sharp contrast with the state of affairs in other countries, for example, the EC, Australia, Canada and Japan. It is obviously not for lack of resources or talent, and the trend should be reversed. Indeed, the US can and should assume a more prominent role.

Currently the area is dominated by a few large general systems and many specialized systems. The existence of many specialized systems with diverse interfaces and philosophies makes their use cumbersome, with steep and independent learning curves. Most of these systems have interpreted languages with their own syntax, philosophies and idiosyncrasies. As it stands, becoming proficient in the use of systems such as Gap, Magma, or Macaulay2, is a huge investment of time and energy for most mathematicians. The front ends of these systems are typically not user friendly and often discourage non-expert users; just combining any two such systems can prove very difficult. It would be very useful to develop friendly front ends for some of the more popular systems to make the application of these tools much less intimidating for the novice user. Developing means for combining effectively the use of two or more systems should be a priority for funding.

A major concern for the ANTC community is ensuring the continuous maintenance and support of computer algebra systems that are established and useful. The systems represent huge investments of time and effort by many mathematicians. Maintaining the systems means more than just fixing bugs (although it must include this, of course). New algorithms need to be implemented and occasional major revisions in the systems need to be made, while an effort should be made to provide continuity, minimizing the demise of code individual users

may have written. If a large system should die for lack of funding, then the mathematical community would lose a major resource as well as the intellectual efforts of the numerous contributors to the system. The computer algebra systems are sufficiently important in the research of many mathematicians that it is time for NSF to consider helping with the costs of maintenance and upgrades, funding more than just the development of systems.

On the other hand, new systems for cutting edge applications and algorithms should also be supported by the funding agencies. Though it may be that many new computational tools in algebra and number theory will ultimately be included as packages in older and larger systems, innovative ideas are often developed by individuals and small teams of researchers working from basic principles.

In general, the ANTC community and NSF should expect from software development projects the following features:

- 1) The source code should be open.
- 2) It should be available on the web.
- 3) It should be well documented.
- 4) It should be maintained and kept up to date.

New and existing software should be developed with connectivity to other systems in mind.

Continued funding of proposals for software development projects should be contingent upon criteria such as: importance and usefulness, level of maintenance, and connectivity to or incorporation in other popular systems. The importance of the system can be documented by the number of users and the nature of the applications. The impact could be measured by the number of citations or a list of the problems that the system has helped to solve.

5. WEB DATABASE

Many areas of ANTC include a heavy component of example-driven computation. That is, results and conjectures are inspired by examples. A particularly famous example is the conjecture of Birch and Swinnerton-Dyer, which emerged from data on elliptic curves and their L -functions.

The collections of data produced in these calculations have a value far beyond their initial intent. For example, Dennis Hejhal has produced a wealth of Maass forms on $SL(2, \mathbb{Z})$, and used this data to study aspects of quantum chaos and quantum unique ergodicity. However, the same data would be of great value in many other studies. For example, there are predictions, based on random matrix theory, for the distribution of zeros and critical values of the L -functions associated to Maass forms. Thus, it would be of great benefit to the wider community if Hejhal's data was made available to anyone who had a use for it.

We suggest that a web-based repository would be a valuable asset to the ANTC community. We envision that initially the repository would include only data (see specific examples below). Later the site would expand to include useful code and working programs.

At present some collections of data exist on the web. Examples include tables of modular forms, elliptic curve data, number field data, character tables of finite quasi-simple groups, and L -functions. But the data lies in individual researchers' personal web pages, and each

data set has its own quirky format. Beyond this publicly available data, there is a huge repository of other data on researchers' personal computers. It is desirable to make this 'hidden' data available, but the prospect of converting the data to a usable format is daunting to most mathematicians.

Creating a central data repository, **with the data in a predefined standard format**, would make it easier for people with data to make it available, and it would also make it easier for other people to find and use the data.

Some of the problems which a web database will address:

- Currently it is hard to find out what data is available.
- The variety of interfaces make programs and data difficult to use.
- Data is in a variety of formats; it is tedious to convert to the desired form.
- The reliability of the data is difficult to determine.
- There are many gaps which need to be filled, and there is also inefficient duplication.
- Current sites are often aimed only at amateurs.

We envision that the database will be of use to mathematicians who just want to look at a few examples, as well as those who want to analyze a large number of cases and do statistical studies, and also students who want to explore. Having the data in one place will make it easy to find.

Before a number theory database can be implemented, the following issues must be addressed:

- develop a standard way of representing the data on the computer;
- have a place to put data;
- have data available in many formats (including an accepted standard);
- have data organized in a way that researchers can use it with a good cataloging system and sufficient documentation;
- verify the reliability of the data;
- keep data and tables up to date;
- give appropriate credit for data used from a database. This can be partially addressed with a good cataloging system (and increased cultural awareness of standards).

The first issue above is critical. If there is an adequate open standard, then everyone just needs to know how to convert from their own preferred format to the accepted standard. In this way everyone can share data with everyone else.

The common approach to data standardization on the web is to "mark up" the data with metadata. 'Mark up' means to surround by tags containing metadata, that is, information about the content or meaning of the data. For example, it is desirable to explicitly provide the information that " $f(x)$ " means " f is a function and x is the argument of that function," as opposed to "the number f times the number x ". That sort of distinction is needed if we want to search, store, and represent content on a computer.

The World Wide Web Consortium (W3C, www.w3c.org, the standards body of the web) provides XML (eXtensible Markup Language) as a general purpose way to create specialized markup languages. The first XML extension was MathML, the Mathematical Markup Language. MathML has the ability to represent the subtleties of mathematical typesetting and also to represent the meaning of simple mathematical expressions. MathML is

designed to handle expressions at approximately the level of introductory calculus, so it does not meet all of the needs of research mathematicians. A project called OpenMath (<http://www.openmath.org/>) may be of use here.

Note: we discussed at length the possibility of creating a simple website which merely has links to the data on other pages. It was decided that this will not provide a long-term solution to the problems of availability, integrity and accountability of the data, and the problem of accessibility and a uniform data standard.

In the future we envision extending this web database to include reusable computer code, as well as on-line programs which will perform a variety of calculations based on user input.

6. COMPUTERS AND PROOF

Computers will continue to take on a more prominent role in both the creation and verification of proofs.

Verifying computer assisted proofs. Our current system of peer review is not designed to handle the review of proofs that rely on a significant body of computer code. For example, Hales' proof of the Kepler sphere packing conjecture relies on about 40,000 new lines of code. The computer code is an essential part of the proof. However, the referees are adamant: checking computer code is not their responsibility. Several attempts to get students to check the computer code fizzled. Four years after the proof was submitted for publication, the peer review system has stagnated at an impasse.

In the end, the code is being checked for the journal by a programmer hired by the author. Does this really qualify as peer review?

Individual referees can hardly be blamed; checking computer code is a task which falls outside the training of most mathematicians. Rather the fault is with a system of peer review that was not designed for large scale computer projects.

Large-scale computer projects have developed their own methods of testing software. However, mathematicians may find many of these methods objectionable, because the standards of a software tester are much lower than those of a careful mathematician. As one popular software testing manual puts it, "It's your job to find and report significant bugs. But you won't find all of them.... If you think you can do that, you either have a very simple product or a very limited imagination."

Using computers to verify and improve proofs. Over the next 10 or 20 years, there does not seem to be a solution to the problem of code reliability, short of extreme measures such as banning the use of computers in pure mathematics.

However, in the long term there is a solution in the form of computer-verified formal proofs and formally verified code. Thus, a vastly increased use of computers may be necessary to overcome the current shortcomings of computer proofs.

The evolution of the proof of the four-color theorem is commendable in this regard. The trend has been toward increased reliability through the increased use of computers. Partial results on the four-color theorem had become so complex by 1965 that mathematicians (Karl Duerre and Heinrich Heesch) turned to computers to automate parts of the proof. The 1976 Appel-Haken proof required 1200 hours of computer time, but the proof itself was only

partially automated. The complete proof is found in 400 pages of microfiche in a back-cover insert to the Illinois Journal. During the month before the proof was announced, over 800 errors were found and corrected in the hand-calculations in the proof. Increased reliability came with the 1996 computer proof of Robertson et al. which eliminated much of the tedious hand-checking. A current initiative (George Gonthier and Benjamin Werner) seeks to give a complete formal computer verification of the four-color theorem.

Formal computer verification methods have developed to the point that significant theorems can be formally verified. It is now a realistic undertaking to give a formal verification of the four-color theorem. Goedel's first and second incompleteness theorems, the fundamental theorem of arithmetic, the fundamental theorem of algebra (a 3-year project), and Sylow's theorem have been formally verified. Current proposals include the prime number theorem (Bob Solovay's challenge) and the classification of semi-simple Lie algebras. The formal computer verification of the classification of finite simple groups is a possible long-term superproject. More ambitiously, the QED manifesto (<http://www.rbjones.com/rbjpub/logic/qedres00.htm>) seeks to express all proof by computer.

The role of ANTC in computer and proof. Our current computer algebra systems (GAP, Mathematica, Maple, and so forth) are inadequate tools for rigorous mathematical proof. Many of these tools do not have publicly available source code, so it is impossible to subject them to the scrutiny that would be required to include their results in a proof.

Over the past few years, a considerable amount of research has begun on the design of systems that integrate computer algebra systems with theorem verification systems. Shankar's work ("Metamathematics, Machines, and Goedel's Proof") shows the value of integrating metamathematical arguments into these systems as well. It is the right moment for pure mathematicians to formulate and convey their design requirements for such systems. If we delay, the systems will be designed and implemented without us (for better or for worse).

7. EDUCATION AND OUTREACH

The computational aspects of ANTC are among the more accessible facets of these areas. As such, computation can be a productive way to introduce students to higher mathematics. It is also a route to introducing the general public to the value of ANTC research. We encourage the ANTC community to seize on these opportunities to bring students into these areas and also to lead the wider community to see the value of this work.

Several researchers have developed innovative courses or programs which incorporate computational research, tools and experiments. These courses have several purposes. One is to form a research team of undergraduates, graduate students and faculty to investigate, theoretically and numerically, interesting conjectures. The second is to train students in using computers to build intuition and investigate problems. Finally, many talented mathematics students are lured away into computer science, finance, and other such fields. This poses a serious threat to the future of mathematics research and education. These courses help keep talented students in mathematics by allowing them, early on, to make contributions to important unsolved problems.

Some professors who are taking such an approach, either through courses or through REUs, include Steven J. Miller and Peter Sarnak at NYU and Princeton, Sara Billey at MIT, Nigel

Boston at the University of Wisconsin, Dennis Hejhal at the University of Minnesota, and David Farmer of the American Institute of Mathematics.

Programs such as the above have had great success in not only keeping students interested in mathematics, but also in generating interesting results. To build on these programs, one needs faculty who are knowledgeable about computational methods and resources. There should be support for faculty who develop innovative programs and research groups along these lines.

One goal would be to develop, on a large scale, a vertically integrated summer program on computational research incorporating undergraduates, graduate students, outreach faculty, and organizing faculty. This program should have ties to various courses around the country and that emphasize the integration of computational methods and tools in the curriculum. Furthermore, to facilitate the development of courses that incorporate research and computing, classrooms or laboratories equipped with sufficient computers for education and exploration are needed.

A proposal for a pilot vertically integrated summer program on computational research for undergraduate and graduate students is included in an Appendix.

Another goal would be to create a website for educating mathematicians, students, and the public about available math software. There are a number of websites that provide links for software downloads, but few provide enough information to make an informed decision as to which package is best suited for different tasks. One obstacle to performing world class computations often seems to be the poor choice of package. Indeed, many world class computations use specially written code. A site that included expert recommendations of packages, evaluations and comparisons of packages, instructions on interfacing packages to new code, links to software downloads, demos, and tutorials would go a long way in furthering research and education.

Finally, it would be helpful to have available a computer where researchers can log on and experiment to learn how to use various packages without having to worry about installing the software. Another major obstacle to choosing the right package is the upfront effort needed just to test a package and see whether it really suits the person's needs.

7.1. Educational outcomes of ANTC Computation.

7.1.1. *Experimental courses.* Several innovative courses using computation exist at the moment such as the “Undergraduate MathLab” at Princeton and NYU (taught by Steven J. Miller, Peter Sarnak, Andrew Wiles and others). These courses have websites housed in individual universities. Another such course is Sara Billey's course on hypergeometric functions at MIT.

For more information on these courses, see

- <http://www.math.princeton.edu/~mathlab/>
- <http://www.math.nyu.edu/Courses/V63.0393/>
- <http://www-math.mit.edu/~sara/classes/18.322.html>

7.1.2. *Innovative textbooks.* Recently several books and instructional materials have been published that show how one could use computation in mainstream courses.

- Arjeh Cohen, Hans Cuypers and Hans Sterk (Eindhoven) published an undergraduate text entitled ‘Algebra Interactive’. Its novel features are the interactive examples and tools for computing in the various algebraic structures. Many of these tools and examples use the computer algebra package GAP, a copy of which is provided on a compact disk with the book. Moreover, abstract notions are enlivened by Java applets, which go beyond the classical examples given in traditional textbooks.
- Steven J. Miller and Ramin Takloo-Bighash finished a first draft of a “textbook” for the Fall 2002 Undergraduate Mathematics Laboratory/Junior Research Seminar at Princeton University. The book contains all the lectures from the course, as well as notes from the background lecture series and results from Junior projects. The book is self-contained and can be used to run an undergraduate course on Diophantine problems. Included (both in the lectures and reports) are problems suitable for undergraduate research.
- “A SINGULAR Introduction to Commutative Algebra” by G.-M. Greuel and G. Pfister. This text introduces commutative algebra in a new style, taking into account modern developments such as algorithmic and computational aspects. As soon as a new concept is introduced, it is shown how to handle it by computer. The computations are exemplified with the computer algebra system Singular, which was developed by the authors.

8. CULTURE CHANGE

The increasing importance of computation in ANTC has changed the way many ANTC researchers think about their subject, the way they teach it, and the way they conduct their research. However this change in the culture of ANTC must be accompanied by other developments to ensure a healthy future for these subject areas.

8.1. Evaluating computational work in ANTC. Computational contributions to research appear to be viewed and assessed differently from theoretical research, as there are yet no accepted procedures and traditions by which these assessments can be made. This issue arises often when computational researchers in ANTC apply for promotion, tenure, and competitive research funding. Similar problems are faced by ANTC graduate students with computationally intensive research projects. Indeed, papers without formal theorems are often viewed as less valuable, when it is clear that conjectures are as important and influential as theorems.

The problems are especially acute for developers of large software packages, and for researchers who have chosen to devote much of their research agenda to computation. The design and development of software packages often involve a significant theoretical component, and there needs to be an agreed mechanism for assessing them.

Change in the culture of evaluating computational research, and computationally intensive theoretical research, is crucially important to acknowledge and encourage high quality, innovative, computational initiatives. Established assessment procedures would also help to distinguish high quality computational work from work that is ‘not so good’.

8.2. Publishing computational results. Publishing computationally intensive research in ANTC is still problematic, with many mainstream journals still reluctant to accept papers in the area. However, striking results, such as the Four Color Theorem, lie in this category.

A natural route is for computational researchers to publish their work in the few journals that currently accept them, such as *Experimental Mathematics* and *Mathematics of Computation*. Such articles should be accompanied by web access to the work that can be used by referees and readers to re-confirm the results. This requires extra effort by the researchers to make their computations easily usable by others. Also, all personnel with an essential involvement in the work should be included as authors.

The ANTC community requires agreed-on guidelines, or standards, for presenting and publishing computationally intensive research. These guidelines are necessary for referees and editors, as well as researchers, to guarantee quality. Such standards are also necessary first step toward the proper recognition of this work for promotion and tenure.

8.3. Publishing software packages. The publication and recognition of computational packages are equally serious issues. It goes without saying that computational packages used in theoretical research must be cited in research articles. This, however, presupposes the existence of a suitable publication representing a computational package that can be cited.

The GAP experience: For several years, the GAP Advisory Council has grappled with the problem of providing appropriate recognition for developers of software packages that form part of the GAP system (a computer system for Group computations). It was decided to offer software contributors the option of a refereeing procedure similar to the review process for journal articles. If a package is accepted, then the developers receive a certification from GAP, and recognition on both the GAP web site and in the GAP documentation. This accreditation process has not yet been tested widely in promotion and tenure evaluation assessments.

An electronic ANTC software journal was suggested for software packages. This would be governed by a high profile editorial board, including some high-profile members of the stature of Mumford, Shor, Knuth, or Lenstra. Packages would be refereed and, if accepted, the journal would publish a description of the package prepared by the author, along with web links to its location. Several issues still need to be addressed: appropriate procedures for ensuring the continued availability of packages, or at least accurate information about their status; and deciding whether the journal should house the entire package. Appropriate systems support would be crucial for the success of such a journal.

A second suggestion was to create an ANTC resource web site, similar to the math arXiv site, to contain such ANTC resources as: computer code, challenge problems, benchmark problems, links to innovative computational courses, books and instructional materials that use computation, novel computations that investigate open conjectures, and counterexamples that have been found using computers. This has some similarities to the web database described in a previous section.

8.4. ANTC and automated reasoning. The use of computer algebra software systems is now widespread in industry, education and scientific contexts¹. At the same time the use of formal methods in hardware and software development has made automated reasoning systems indispensable, in part because of ‘the complexity and the sheer size of the reasoning tasks involved’. Limitations on both the deductive power of existing computer algebra systems and the computing abilities of existing automated reasoning systems indicate that there is need for significant improvement, and in particular for the integration of computer algebra and automated reasoning systems. The aim is to combine ‘the reasoning capabilities of automated reasoning systems with the computational power of computer algebra systems’.

An illustration of recent developments along these lines is the funding of a core subgroup of the CALCULEMUS² interest group (by an IHP network grant under the EU 5th framework) to conduct research in areas including Integrating Computer Algebra and Proof Search (some case studies), Proof Planning, Interactive Theorem Proving, Computer Aided Formalization of Mathematics, and Open Mechanized Reasoning Systems.

8.5. Side effects of software distribution. A particular problem for authors whose packages are made publicly available is the receipt of repeated requests for help with installation of packages on individual machines. For an individual researcher-author this can be an overwhelming burden. (Commercial distributors include the cost of a technical support department in the software price.) Replying to such routine queries would be a welcome service that the journal suggested above could offer both to developers and users.

9. CONCLUSION

There are many instances in which researchers in ANTC could incorporate computational components in their work. The authors would very much like to see our community take the initiative to enhance our productivity through such avenues. As a group we need to be involved in a better organized approach to computation. We should educate our students in the ways of computation. We should support our colleagues’ endeavors in this arena. We should contribute to the common goal of usable web databases. We should experiment with supercomputers and with clusters, the Stein model or Beowulf. We should think carefully about the role that computers play in mathematical proof. We need to begin setting goals and learn how to achieve them. And we need to begin to ask for the things that we need to accomplish these goals.

Several of the authors intend to begin the process of writing grant proposals to move in these directions. At the start these initiatives are likely to include summer computational programs for undergraduates, the creation of a web database, the beginning of a new journal for software, the creation of Stein clusters, and the increased use of supercomputers. Later initiatives could include the development of a software system which meets the requirements outlined in the software section, the backing for someone in a mathematics department to

¹Paraphrasing Steve Linton and Roberto Sebastiani, Editorial: the integration of automated reasoning and computer algebra systems, *J. Symbolic Comp.* (2002) 34, 239.

²The CALCULEMUS Project: Systems for Integrated Computation and Deduction, see <http://www.calculemus.net/>

run a computational lab along the lines of other scientists, and an effort to create an institute devoted to mathematical computation.

If you in these matters, please send e-mail to *computation_in_ANTC@aimath.org*.

10. SUMMARY OF RECOMMENDATIONS

We recommend that the ANTC community, with the assistance of the NSF,

- recognize that basic research in ANTC is fundamental to many, if not most, of the developments in the so-called digital revolution, and so it merits the support of the NSF at a level comparable to subjects traditionally considered a part of applied mathematics;
- put a high priority on attacking long-term theoretical problems that can be positively affected by computation;
- work toward a culture change that fosters encouragement and appropriate rewards within mathematics departments and the mathematics community for contributions to software and algorithm development and for the increase of knowledge through the creation of worthwhile data;
- develop and maintain databases accessible by the internet and in a format easily used by a large variety of users and systems;
- maintain a prioritized database of good problems that should be attacked by computer;
- develop a collection of experts and a system whereby members of the ANTC community can receive and give assistance on computational matters;
- assist the development of mathematical computer expertise among undergraduates, graduate students, and faculty, through focused, vertically integrated programs, such as nonstandard classes during the academic year and REU-like programs during the summer;
- develop a collection of inexpensive high powered parallel computer systems that can be accessed remotely in an easy way;
- develop easy to use, open source, high level software for use in specialty fields of ANTC, maintain such products that have already been developed, and create tools which make it easy to use and combine those packages;
- work on the problem of how to referee proofs that are either computer generated or involve large computations.

It is our belief that progress and innovations along these lines will help ensure the continued vitality of research in ANTC in the rapidly changing scientific and economic environment of the 21st century.

APPENDIX I SOFTWARE PACKAGES

Here we provide a list of available software that mathematicians in ANTC tend to use. An asterisk indicates that the software is US based.

(1) Systems

- Caiss-stat*
- GAP
A high level system for group theory, combinatorics and other applications. Free software, can be downloaded from the network. Originally developed in Germany, currently in Scotland.
- Magma
A high level all-purpose system for algebra, number theory and combinatorics. Applications are to group theory, number theory, coding theory, commutative algebra and other areas. Non-profit, but proprietary software. Source code is available, but is usually released as a compiled program. An Australian product.
- Maple
- Mathematica*
- Matlab*
- PARI-GP

(2) Specialized systems for mathematics.

- Cocoa
A system for commutative algebra, developed in Italy.
- Lie2
A system for the combinatorics of Lie algebras and Lie groups. Proprietary software. Source code is available. Developed in Holland.
- Macaulay2*
A high level system for commutative algebra and algebraic geometry. Free software, can be downloaded from the network. An American product that was developed with the support of NSF.
- Magnus*
A system designed and specialized for investigations in infinite group theory. Created mostly in the US with substantial support from the NSF
- PORTA and Polymake
Two systems for calculations with polytopes.
- Singular
A system for polynomial computation with many features for global as well as for local commutative algebra and algebraic geometry, including libraries for symbolic-numerical polynomial solving. Developed in Germany.
- Symmetrica
A specialized system designed combinatorial analysis of problems concerned with symmetric groups. A German product.
- Tiger

APPENDIX II
THE MATHEMATICS EXTREME COMPUTATION CLUSTER AT HARVARD
March, 2003
by William Stein

In Spring 2002 I assembled and configured MECCA, which is a rack-mounted cluster of six fast computers for use by mathematicians who are doing demanding computational work. This article is about my experience building and maintaining MECCA. It should be of use to anyone considering undertaking or funding a similar project at their institution.

As a graduate student at Berkeley and a faculty member at Harvard, the computational resources available to me at my host universities consisted of scattered Sun workstations running at about one-fourth the raw speed of the current Pentium processors. These machines spent much of their time running Netscape and texing documents, so they were not suitable for demanding computations that could easily use all available resources. Sure, at each institution a senior faculty member had a powerful computer (McMullen at Berkeley, Elkies at Harvard), but that was for his own personal use.

In 2001, the Harvard sysadmin, Arthur Gaer, mentioned that the department was tentatively considering spending several tens of thousands of dollars (that it didn't yet have) on a single multi-processor Sun workstation to support computation-intensive work. My opinion was that such a workstation would be solid but hardly useful; the raw computational power would scarcely touch what two cheap Intel-based Linux boxes could do, though the Linux boxes would likely be less reliable.

I decided to build a cluster of dual processor machines running Linux. I did research and discussed possible configurations with Berkeley grad student Wayne Whitney and a Harvard undergrad named Alex Healy, and requested money. Finally, I secured a grant of \$6000 from Harvard, and Harvard alumnus William Randolph Hearst III gave me an additional \$14000, which made the budget \$20000.

I decided to assemble an Athlon-based system. The Athlon 2000MP is a multi-processor-ready Pentium-like CPU that Athlon claims has performance that is similar to a 2GHz Pentium IV. I selected the Athlon 2000MP processor in March because it was the fastest available budget-priced multi-processor capable CPU on the market. Intel's only fast multi-processor capable CPU was the Xeon, which was then much more expensive (the Xeon might be a good choice today). Six months later, Athlon has just announced the 2200MP, so I don't feel like Athlon 2000MPs are out of date.

In February 2002, I ordered six custom-built Athlon 2000MP machines in 2U-sized rack-mount cases from www.pcsforeveryone.com which is a local Cambridge "chop shop". They ordered the parts I wanted, assembled them, tested them, found surprisingly often that they were defective, got replacements, and finally delivered the individual computers. I still have occasional hardware reliability problems with one of the nodes, even after returning it for service under warranty, and it is currently off (a CPU fan had failed, so they replaced the CPU fan, but not the CPU, which is a cheap "solution" that didn't work).

Unwrapping the rack and putting the computers in it took Alex Healy a full afternoon. Once assembled, the machine had to be kept in my office, because the math department's server closet was tiny and currently full of equipment. It would be several months until we made

room in the server closet for the cluster. In the meantime, I kept a rack of noisy and hot computers running in my office. When students came to see me during office hours, they had to shout over the 30 cooling fans in MECCAHA.

And, the fuses kept blowing! My neighbor's office is on the same circuit as mine and when he returned from vacation and turned his computer on, the circuit breaker blew, so I had to call the electricians out to switch it back. I moved back to running only four machines, then once increased to five, again blowing the circuit.

MECCAHA's operating system is Redhat 7.2 with Linux kernel version 2.4.16 on all six nodes. MECCAHA also uses openMosix, which makes the rack of six computers appear to the user as a single computer with 12 processors and 13GB memory (though a single process should not use more memory than on any node). Under openMosix, jobs are automatically migrated from one node to another to dynamically balance the overall system load. Users only have accounts and login privileges for the master node, and never worry about logging into other nodes. I also configured MECCAHA to use the ext3 journaled filing system, so, e.g., I can pull the plug from the wall, plug it back in, and have MECCAHA back up in five minutes with absolutely no file system corruption.

For computations, people mainly use MAGMA, PARI, Python, C/C++, and Mathematica. Though Harvard has a Mathematica site license, I HATE administering Mathematica because the licenses regularly expire and limit the number of copies of Mathematica that can be run at once (there should be a way around the latter problem). MAGMA for Linux, on the other hand, requires no license and is free to me because I'm a MAGMA developer. Evidently, Maple is expensive, so we have only a limited Sun license for Maple in the math department.

Here is how I organize computation of a basis for the space of modular forms with level N and weight 2 for N between 1 and 1000. I run 12 jobs simultaneously that each look to see the next level that hasn't been computed, compute that level, and save the result. If it took 1 day to do this computation on my 1Ghz Pentium III last year, it will take only 1 hour to do it on MECCAHA. When I am in the throes of a big computation, having this kind of computational resource available to me is extremely exciting. Instead of waiting 1 day, I wait only an hour to generate more than enough data to stimulate theorem proving!

I've given MECCAHA accounts to nearly 80 mathematicians all over the world. Abuse of the system by users is rare but not unheard of. Somewhat surprisingly, the usage pattern comes in bursts. There are almost always at least two or three jobs running, but every so often many mathematicians simultaneously become inspired to run lots of computations all at once.

I am the only systems administrator of MECCAHA, and I typically spend under five hours a week on administrative responsibilities. I still haven't upgraded the Linux kernel or openMosix since March, but I probably should since there have been a few unexplained problems that might be fixed by a Linux and openMosix upgrade. I use a 30GB Onstream ADRx2 tape drive to make regular backups.

If I were to build a similar cluster from scratch again, I would probably buy more expensive and better warrantied pre-configured dual-processor rack mount nodes instead of custom designing the nodes myself. I definitely would not have kept the computer in my office. When first designing MECCAHA, I thought long about whether or not to stack a bunch of

conventional cases on shelves or to buy a rack and rack-mount cases. A rack costs nearly \$1000 and rack-mount cases cost more than double what ordinary cases cost. In retrospect, it would have been madness to buy conventional cases and shelves, because I've had to move the cluster around many times, and it barely fits in the tiny server closet. The \$1500 premium for a rack-mounted system was well worth it. I also deliberated between a fancy serial console or a KVM (keyboard, video, mouse) switch; I went with the \$500 KVM, which turned out to be an excellent choice.

The six nodes are networked via a switched 100Mbps ethernet network. I wish the network were faster, because it takes a few minutes to transfer 1 GB from one computer to another. Since user programs migrate between machines and frequently do use in excess of 1GB memory, this transfer time is significant. I purchased 100Mbps ethernet instead of 1Gbps ethernet, because I read that 1Gbps ethernet with Linux is not very reliable, and there can be significant latency problems. Since I didn't have the resources to experiment with many configurations, I opted for 100Mbps, which is very easy.

APPENDIX III
 PROPOSAL FOR A VERTICALLY INTEGRATED SUMMER PROGRAM
 October, 2002

This supplement to the FRG grant “L-functions: symmetry and zeros” will support a vertically integrated summer program in computational mathematics. This will be a pilot program for a larger future endeavor.

THE VERTICALLY INTEGRATED SUMMER PROGRAM

This program will take place at The American Institute of Mathematics (AIM) during summer, 2003, and will be administered by Brian Conrey, David Farmer, Steve Miller, and Michael Rubinstein. In addition to those 4 researchers, the participants in the program will be 6 undergraduate students, 2 graduate students, and Chris Hughes, who is an AIM postdoc.

The purpose of the program is to form teams of students and faculty to investigate fundamental questions in number theory which have been suggested by the ongoing work of Conrey’s FRG. In addition to the significant scientific progress which will be accomplished by these teams, the students will benefit by the opportunity to work closely with research mathematicians. This will give the students an exposure to the culture of research mathematics and will also provide them an opportunity to learn interesting mathematics and computational techniques which are not normally encountered in the classroom. Also, there will be a significant pedagogical benefit to the graduate students and postdocs as they learn to work on research projects with undergraduates.

The vertically integrated nature of the program, in which faculty work with postdocs and graduate students, and all three work with undergraduates, will provide valuable training which is not seen in a typical REU program. The faculty will set good examples of how to work with students on a research project, and this will assist the graduate students and postdocs as they prepare for the transition to the next level.

We expect that this program will inspire many of the students to be interested in continuing in mathematics to the research level. We also believe that it will help produce faculty who will continue to inspire students through successful undergraduate research experiences.

Background and future plans.

This program is a natural extension of previous work of the investigators:

Miller was involved in the creation and administration of an undergraduate research class on computational methods for mathematics students. He is experienced at helping students learn the basics needed to understand the background behind the problems to be investigated and to do the programming necessary to investigate the questions on a computer. These skills will be critical in the early weeks of the program. This program will also involve some new challenges for Miller because the summer program will be the full-time activity of the students, as opposed to one class in a full semester.

Conrey, Farmer, and Rubinstein have each worked with several undergraduate students on research projects. They have experience in working with individual students, which will be valuable in selecting appropriate projects for the summer. They have also been successful at introducing students to the process of documenting their work and giving talks on their

research. These skills will be critical in ensuring that the students receive maximum benefit from the program. However, the somewhat larger scope of this project and the team-based nature of the work is a new challenge which will give them valuable new experience.

We envision this program as a pilot project for a future program which will be approximately 5-10 times larger. We believe that computational investigations are an excellent way to introduce students to interesting mathematics, and that a vertically integrated program is an ideal way to prepare graduate students and postdocs to become faculty who will continue to work with undergraduate research students. Such a program will encourage students to pursue higher mathematics, and it will also help produce faculty who will continue to inspire students through interesting research projects.

At present the investigators have skills in all of the individual aspects of the planned program: selecting appropriate projects for students, working with students on computer programs, helping students understand where their project fits in with the larger picture, etc. What the investigators are lacking is the experience of putting together a large vertically integrated group to work on these projects. This pilot program will provide the experience necessary to allow them to plan a larger project.

Following the summer program the investigators will be in an ideal position to prepare a proposal for an expanded project.

Program details.

The program will take place at AIM in Palo Alto. AIM will dedicate one very large room, with desks, whiteboards, and several computers, which will be suitable for student use. The program will last for 8 weeks, from June 1 until July 25.

As soon as they arrive, students will *immediately* begin working on projects. The investigators' experience is that an ineffective way to begin an undergraduate research experience is to hand the student some books and tell them that they have to learn a lot before they can do anything. The projects we plan all have an element which can be investigated on a computer with essentially no preparation. This allows the student to accomplish something tangible on the first day. This leads the student to feel positive about their abilities, and it makes them very receptive to learning background so that they can put their work in the context of the larger picture. Throughout the summer we will maintain a mix of computer programming, analyzing data, background reading, and theoretical work.

The students will learn new mathematics and will gain an understanding of a mathematics research environment. In addition, the students will gain experience in writing computer code and in some cases they will learn about packaging code so that it can be used by other people. At the end of the program the students will TeX a report on their research and also give a talk to the group. The students are also expected to give a talk in the Fall to the other students at their home institution as well as at a student research conference, if appropriate.

An account of the summer program will be prepared by the investigators and they will maintain a website describing the program and offering advice to both students and faculty who are interested in undergraduate research. This will be a valuable resource for anyone interested in running a similar program.

At the beginning of the program the students will be surveyed to determine their mathematical background, attitude toward mathematics and computation, and future plans. This will be valuable for matching projects to students, and will also serve as a baseline for judging the effectiveness of the program. At the end of the summer the students will be surveyed to determine what mathematical and computational skills they have acquired, their understanding of the mathematical culture, their attitude toward mathematics, computation, and research, and their thoughts on the overall summer program. With only 8 students the information will necessarily be anecdotal, but it will serve to help us refine the program and to determine if a larger project is warranted. Finally, we will track the students after the end of the program, at least to the point at which they choose to go to graduate school or to industry.

Farmer, Miller, and Rubinstein have funds which are able to pay salary and travel for a total of 4 students. In this supplement we ask for support for 2 undergraduates and 2 graduate students, plus accommodations for the other 4 undergraduates.

Scientific program.

The investigators have identified a number of projects which are central to the theme of L -functions and Random Matrix Theory and also are suitable for study by the students in the summer program. All of the projects will involve numerical calculations, and most of them also have an associated theoretical aspect.

We will also run a seminar on random matrix theory and the connection with L -functions, which underlies all of the projects.

Low-lying zeros in one parameter families of elliptic curves. Random matrix theory has been fantastically successful at predicting the statistical behavior of the low-lying zeros of families of L -functions. We will investigate whether these predictions are robust enough to handle the case of L -functions associated to elliptic curves.

The advantage of studying families of elliptic curves (over other families of L -functions) is that, assuming the Birch and Swinnerton-Dyer (B-SD) conjecture, we can construct families of elliptic curves where we expect a multiple zero at the critical point. Do these forced zeros affect the distribution of the first few zeros above the critical point?

We can construct one-parameter families of elliptic curves with rank r over $\mathbb{Q}(t)$. Standard conjectures imply (as long as our family is sufficiently general), that half the curves will have even functional equation. By B-SD, each curve will have at least r zeros at the critical point. Thus, half the curves will vanish at least r times at the critical point, half at least $r + 1$. Is there extra vanishing, ie, how often are there at least $r + 2$ zeros? Experimentally, about 18% of curves in a given family are found to have extra vanishing, although the number of curves investigated is very small, and this could be the artifact of corrections from lower order sub-families. Also, what is the distribution of the first zero in these families?

Students will begin work by learning how to use available programs to calculate the critical vanishing, and the low-lying zeros, of elliptic curve L -functions. This will quickly put them in a position to look at interesting families. This project would also be a good opportunity for a student who was interested in doing a theoretical calculation to determine a random matrix theory prediction for the first zero in families with multiple critical zeros.

Weight 3/2 modular forms. It is possible that Delaunay’s heuristics [3] for Tate-Shafarevitch groups could be used in combination with random matrix models [2] to obtain a Lang-Trotter type of conjecture for certain weight 3/2 modular forms. There are still some complications that have to be worked out, the most significant of which is the contribution from some ‘exceptional’ primes. Experimentation should reveal some patterns which will allow one to make a guess as to the contribution from these primes.

The student could begin calculating right away because it is easy to describe a mechanical procedure for generating modular forms. Generating these functions will leave the student interested in understanding the theoretical ideas behind them. This is a project where a student is likely to make valuable contributions: the point is to conjecture a pattern from some numerical data, and students are quite good at that.

Zeros of derivatives. The zeros of the derivative of the ζ -function and the zeros of the derivative of the ξ -function are important theoretically, but very little is known about their distribution. Mezzadri [7] has made a random matrix prediction for the zeros of $\zeta'(s)$ that are very close to the critical line, but in general there is not even a precise conjecture.

To carry out the calculations, the student could make use of Rubinstein’s L -function package [10] and expand its functionality by adding a procedure to search for zeros of the derivative. Checking the predictions would require a careful comparison with the conjectures given in Mezzadri’s paper. It is also possible that a student could refine the work in Mezzadri’s paper by considering next-to-leading order terms.

There also are several questions about the zeros of ξ' which could be investigated both theoretically and numerically. This is relevant to the problem of Landau-Siegel zeros.

Parts of random matrices. Start with an $N \times N$ unitary matrix and take M of the eigenvalues, where $M < N$, (eg $M = N/2$ or $M = \log N$). Now consider the usual statistics (characteristic polynomial, traces of matrices, etc) calculated with the M eigenvalues, but averaged over the full $N \times N$ group. The question is: do the quantities behave like they are N/M independent copies of $M \times M$ matrices, or something else? If they do, how big must M be with respect to N ? The motivation for this question comes from work of Steve Gonek, Chris Hughes, and Jon Keating on an improved random model for the ζ -function.

The student could begin a numerical investigation by implementing standard techniques for generating random unitary matrices, and then just selecting a subset of the eigenvalues as described above. These questions may also be approachable theoretically, depending on how closely the calculation mirrors that of the standard case.

The L -function calculator. Rubinstein is nearing completion of his L -function package [10], and the remaining work involves optimizing key subroutines. Working on this project would be a fantastic experience for a student who was already adept at C++ and who had a keen interest in high-performance computing.

REFERENCES

- [1] B. Conrey, D. Farmer, P. Keating, M. Rubinstein and N. Snaith, *Integral Moments of L -Functions*, <http://arxiv.org/pdf/math.NT/0206018>

- [2] B. Conrey, J. Keating, M. Rubinstein, and N. Snaith. *On the frequency of vanishing of quadratic twists of modular L -functions*. Proceedings of the Millennium Number Theory Conference.
- [3] Christophe Delaunay, *Heuristics on Tate-Shafarevitch groups of elliptic curves defined over \mathbb{Q}* , Experiment. Math. **10** (2001), no. 2, 191–196.
- [4] H. Iwaniec, W. Luo and P. Sarnak, *Low lying zeros of families of L -functions*, Inst. Hautes Études Sci. Publ. Math. **91**, 2000, 55 – 131.
- [5] N. Katz and P. Sarnak, *Random Matrices, Frobenius Eigenvalues and Monodromy*, AMS Colloquium Publications **45**, AMS, Providence, 1999.
- [6] N. Katz and P. Sarnak, *Zeros of zeta functions and symmetries*, Bull. AMS **36**, 1999, 1 – 26.
- [7] Francesco Mezzadri, *Random matrix theory and the zeros of $\zeta'(s)$* , preprint.
www.arxiv.org/PS_cache/math-ph/pdf/0207/0207044.pdf
- [8] S. J. Miller, *1- and 2-Level Densities for Families of Elliptic Curves: Evidence for the Underlying Group Symmetries*, Ph.D. Thesis, Princeton University, 2002, <http://www.math.princeton.edu/~sjmiller/thesis/thesis.ps>.
- [9] M. Rubinstein, *Evidence for a spectral interpretation of the zeros of L -functions*, Ph.D. Thesis, Princeton University, 1998, <http://www.ma.utexas.edu/users/miker/thesis/thesis.html>.
- [10] Michael Rubinstein *The L -function calculator*. www.ma.utexas.edu/~miker.
- [11] Z. Rudnick and P. Sarnak, *Zeros of principal L -functions and random matrix theory*, Duke Journal of Math. **81**, 1996, 269 – 322.

APPENDIX IV
THE AUTHORS

- Eric Bach, bach@cs.wisc.edu
- Gilbert Baumslag, gilbert@groups.sci.ccny.cuny.edu
- Louis Billera, billera@math.cornell.edu
- Sara Billey, sara@math.mit.edu
- Nigel Boston, boston@math.wisc.edu
- Jon Carlson, jfc@sloth.math.uga.edu
- Brian Conrey, conrey@aimath.org
- William Duke, wdduke@ucla.edu
- Noam Elkies, elkies@math.harvard.edu
- David Farmer, farmer@aimath.org
- Thomas Hales, hales@pitt.edu
- Dennis Hejhal, hejhal@math.umn.edu
- Steven Joel Miller, sjmiller@math.princeton.edu
- Cheryl Praeger, praeger@maths.uwa.edu.au
- Dan Rockmore, rockmore@cs.dartmouth.edu
- Fernando Rodriguez-Villegas, villegas@math.utexas.edu
- Michael Rubinstein, miker@math.utexas.edu
- Carla Savage, savage@cayley.csc.ncsu.edu
- William Stein, was@math.harvard.edu
- Mike Stillman, mike@math.cornell.edu
- Rekha Thomas, thomas@math.washington.edu
- Roger Wiegand, rwiegand@math.unl.edu