Available online at www.sciencedirect.com
SCIENCE DIRECT®
ADVANCES IN Applied Mathematics
Advances in Applied Mathematics 31 (2003) 532–543
www.elsevier.com/locate/yaama

Liebe Opa Paul, ich bin auch ein experimental Scientist! ☆

Doron Zeilberger

Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA

Received 9 October 2002; accepted 28 December 2002

Dedicated to the memory of my grandfather Paul Alexander

Abstract

Contrary to popular belief, Math *is* an experimental science. Hence I am an experimental scientist, just like my beloved grandfather, Dr. Paul Alexander (1870–1942, Dr. Phil. Rer. Nat., Chemie, 1897).

Leipzig

This is my first visit to Leipzig. My main reasons for coming here are *personal*: to look up the graves and dwellings of my great-grandparents, Salomon and Rebecka Alexander, and to explore the city and the university where my grandfather, Paul Alexander, grew up and studied. But I thought that it would be nice to combine the business of family pilgrimage with the pleasure of giving a math talk.

Even though this talk is supposed to be on *math* rather than on *family history*, let me nevertheless spend a few minutes telling you about my grandfather Paul Alexander, to whom this talk is dedicated.

☆ First version: October 9, 2002. Edited version of the transcript of a talk (entitled: 'Determinants and Computer Algebra') given at the Max Planck Institute, Leipzig, August 5, 2002, 15:00–16:00. Accompanied by the Maple package CLD available from http://www.math.rutgers.edu/~zeilberg/programs.html. Supported in part by the NSF.

E-mail address: zeilberg@math.rutgers.edu.
URL: http://www.math.rutgers.edu/~zeilberg/.

0196-8858/$ – see front matter
doi:10.1016/S0196-8858(03)00025-3

A few words about Paul Alexander

1. Paul got his Dr. Phil. in 1897, here in Leipzig. His dissertation was entitled: *"Über die Einwirkung von o-Nitrobenzylchlorid auf Natriummalonsäureäthylester."*
2. Paul was the inventor of efficient *verfahrens* for the regeneration (recycling) of caoutchouc (rubber).
3. Paul had many patents, e.g., US # 844077, issued in 1907 entitled *"Process for the production of aqueous caoutchouc solution and the regeneration of rubber waste."*
4. He contributed several articles to the famous Ullmann *Enzyklopädie* of Industrial Chemistry.

Paul's academic *vater* was the great chemist *Johaness Wislicensus* (1835–1902), who made many important contributions to chemistry. One was the suggestion that there are *geometrical isomers* exemplified by the two forms of the lactic acid. He inspired, and later enthusiastically endorsed the revolutionary theory of Le-Bell and J. van t' Hoff (who got the very first chemistry Nobel, in 1901). He was, most probably, a very nice guy!

But not *all* Leipzig professors were so nice. One especially nasty specimen was the great organic chemist *Hermann Kolbe*, who was Wislicensus's predecessor at the Leipzig chemistry chair. Kolbe, known for his very acerbic wit, commented on the Le-Bell–van t' Hoff theory as follows:

> "... There is an overgrowth of the weed of the seemingly learned and ingenious but in reality trivial and stupefying natural philosophy... which had been dressed up in modern fashion and rouged freshly like a whore whom one tries to smuggle into good society where she does not belong..."

It would be unfair to ridicule Kolbe for the *substance* of his critique, since now we have the benefit of hindsight, and it is not his fault that history proved him wrong. Science does need its share of conservatives to serve as bouncers to guard us against wild speculations like cold fusion and organic transistors. Nevertheless, one can be critical without being mean, and often we are critical because we feel like being mean.

Controversy is much more prevalent in science than it is in math, but even math has its share. As late as 1903 there were still people who did not accept non-Euclidean Geometry. We all heard about the Kronecker–Cantor and Hilbert–Brouwer feuds. More recently we witnessed the heated debate concerning the role of 'theoretical math,' as well as the Andrews–Zeilberger mini-controversy about semi-rigorous mathematics. As math will become more scientific, we should expect more controversy, which will make things more interesting!

Yet another Leipziger was *Hermann Hankel*, who said:

> "In most sciences one generation tears down what another has built, and what one has established, another undoes. In MATHEMATICS ALONE each generation adds a new story to the old structure."

In other words, math is a peaceful, non-violent, cumulative enterprise. Hankel also expressed another "obvious truth" about our profession:

> "Mathematics is purely intellectual, a pure theory of forms, which has for its objects not the combination of quantities and images, but things of thought to which there *could* correspond effective objects or relations, even though such a correspondence is *not necessary*" (my emphasis).

But even though Hankel was probably much nicer than Kolbe, his statements too are starting to be wrong. Because of the *Computer*, Mathematics is becoming an empirical, descriptive, and experimental science, just like Chemistry! 'Modern' Math, that is supposedly *a priori*, will soon join the ranks of Aristotelian physics that was also *a priori*. Unlike the latter, however, it will not be labelled false, but, perhaps worse, would be considered utterly trivial, since computer-generated math will be able to discover, and prove, much deeper results.

The first mathematical area that took advantage of the computer revolution was *Numerical Math*.

Here is a quotation form yet another Leipziger, do you know who?, that in a very major International Congress said:

> "In the former times there were obvious reasons why n was rather small. This is why *Numerical Math* did not appear as a discipline of its *own* before the help of electronic *computers* was available."

That same person also said, later at the same talk:

> "Large scale computations are those which are *almost* too large to be computed on *present* machines" (emphasis added).

I am sure that most of you will recognize these words as belonging to your esteemed colleague, Wolfgang Hackbusch, who in his insightful plenary ICM 1998 talk presented the state of the art in numerical math and scientific computation, and stated that in order to be able to solve very large problems, one has to make some *compromises*. The first compromise is to abandon the *exact* and settle for the *approximate* (what he called ϵ-oriented). Another compromise is to abandon the *general* and settle for the *special*, i.e., trying to solve special classes that often come up in practice, e.g., *sparse* systems, or his own favorite, H-systems.

He also talked about *Algorithmic Paradigms*. In particular, about *Hierarchy*, *Adaptivity*, and (*De*)*composition*.

All this sounds like a *sound* methodology for a *science* that has a strong empirical and experimental flavor. In addition, numerical mathematicians do *numerical experiments* on a regular basis to test their algorithms, and the empirically observed performance is often much better than the theoretical, *a priori*, prediction.

I strongly feel that Hackbusch's talk [H] should be required reading to all *pure* mathematicians, especially to those, like myself, who try to get as much as possible out of the computer. We too should be able to develop *algorithmic paradigms* and *research methodologies* of our own. A good start is by adapting to our needs the already acquired wisdom from numerics, as described by Hackbusch.

So far the use of computers in 'pure' math, with a few exceptions, was rather *methodologically* boring. It consisted mainly in testing conjectures.

Much more interesting, from the *methodological* point of view, are *computer-assisted proofs*. The most famous being the Appel–Haken Four Color Theorem. Here there was a *human-machine dialog* that helped *design* the proof, and once the proof was proposed (a certain explicit unavoidable set of reducible configurations), it was verified by computer.

There is also a rapidly growing effort in *automated proving*. These can be roughly divided to *logic-based* programming, pursued in AI and computational logic, that uses *resolution, tableaux* and other methods, and *ansatz-based* programming, in which the objects are known to belong to a well-defined algebraic class that possesses *canonical* forms, or at least *normal forms*, making it decidable whether $A = B$ or $A = 0$, respectively.

A famous example is *Euclidean Geometry*, in which, thanks to Rene Descartes, the objects are *rational functions* in the parameters, and in more complicated situations *ideals*, for which the Buchberger algorithm supplies a canonical form. There is also *WZ theory*, that is an *algorithmic proof theory* for *hypergeometric summation and integration* identities. However, in both these cases the algorithms themselves were created by humans, and while it is true that they can prove many results that previously required ad hoc human proofs, the very existence of these algorithms makes these 'computer-generated proofs,' and hence also the results that they prove, trivial in some sense, since we are *guaranteed* to get a proof or refutation, time- and space-limitations permitting.

But what makes research so exciting is that it is a *gamble*. You do not know, beforehand, whether you will succeed or fail. You also want to allow for *serendipity*, the possibility that in your computer's attempts to prove Goldbach it will discover something even more interesting. So what we desperately need are *Algorithmic Paradigms for Computer-Generated Research*.

In other words, we need methodologies for creating new algorithms that will enable computers to discover, and *prove*, new results, without knowing, beforehand, whether it will succeed, but with a fair chance that it will.

For the sake of simplicity, let us focus on proving *identities*. These are mathematical statements whose format is $A = B$. The traditional way is to try and manipulate A, finding another object A_1 that 'looks different' but is really the same. The problem is that there are usually several choices. Then one can try to find A_2, getting a string $A = A_0 = A_1 = A_2 = \cdots$, and if in *luck*, or one has a good intuition, or the problem is not very deep, one gets to B. Since at every stage there are several choices, and there is no upper bound for the number of steps, this method leads to exponential explosion.

If both A and B belong to a class of mathematical objects for which there exists a *canonical form*, and there is also an *algorithm* $A \to c(A)$, for reducing any object to its canonical form, then all one has to do is compute $c(A)$ and $c(B)$ and see whether they are the same.

But what if you do not know B? In other words, you have an input A that is 'ugly,' and belongs to a *general ansatz*, but you hope that there exists a *nice* B, such that $A = B$.

By ‘nice’ I mean belonging to a suitable *subansatz*, a *specific ansatz*. Can the computer find B? (if it exists?).

Let us assume that the objects of the *specific ansatz* have a natural complexity, such that one can express the objects of any given, finite, complexity, in generic form, with *undetermined* coefficients.

To make this non-trivial, we must assume that A and B are really infinite classes, i.e., $A(n)$ and $B(n)$, where n is an integer parameter. We assume that for any specific n_0, $A(n_0)$ and $B(n_0)$ are computable, where the latter is in terms of undetermined coefficients. Then by plugging-in enough values for n_0 and solving the system $A(n_0) = B(n_0)$, $n_0 = 1, 2, \ldots, L$, for sufficiently large L, for the unknown undetermined coefficients of $B(n)$, and if the computer finds a solution, then we have a genuine new theorem, that the computer discovered from scratch. Once conjectured, it should be routine to prove that $A(n) = B(n)$ by plugging into the defining equation of A in the *general ansatz*.

Many times it is not possible to prove that $A(n) = B(n)$ directly. Then one looks for more general objects $A'(n,r)$, that does belong to a general ansatz, and such that $A(n) = A'(n,0)$. If that general ansatz contains a subansatz of nice objects $B'(n,r)$, one may try to find it, prove algorithmically that $A'(n,r) = B'(n,r)$, and finally deduce that $A(n,0) = B'(n,0)$, where $B'(n,0)$ is nice since $B'(n,r)$ is.

If the above seems a bit vague, I hope that the case-study below, of *automated* (symbolic) determinant-evaluation, using the *Dodgson ansatz* for the *general* ansatz, and the *hyperhypergeometric* ansatz for the *specific* (nice) ansatz, would make this approach crystal clear.

Trying to abstract from the well-known explicit evaluation of the determinant of the Hilbert matrix

The Hilbert matrix

$$A(n)_{i,j} := \frac{1}{i+j+1}, \quad 0 \leqslant i,\ j \leqslant n-1,$$

is dear to numerical analysts because it is a famous example of a badly-conditioned matrix. Its determinant has a well-known explicit evaluation

$$\det(A(n)) = \prod_{i=1}^{n-1} \frac{i!^4}{(2i+1)!(2i)!}.$$

Let us call the right-hand side $b(n)$. What is nice about $b(n)$ is that the ratio: $c(n) := b(n+1)/b(n)$, is a hypergeometric sequence namely, $c(n) = n!^4/((2n+1)!(2n)!)$. But a *hypergeometric sequence* is precisely one whose consecutive ratio is a rational function. In this case $d(n) := c(n)/c(n-1)$ equals $n^2/(4(2n+1)(2n-1))$.

How would *anyone* start to prove it? A natural way would be by induction on n. However, having only one parameter is too restrictive.

My favorite way to evaluate determinants [Z1,Z2] is

Reverend Charles Lutwidge Dodgson's determinant condensation rule

It states the following. For any n by n matrix A, let $A_r(i,j)$ denote the r by r minor consisting of r contiguous rows and columns of A, starting with row i and column j. In particular, $A_n(1,1) = \det A$. Then, according to Dodgson [D],

$$A_n(i,j)A_{n-2}(i+1,j+1) \\ = A_{n-1}(i,j)A_{n-1}(i+1,j+1) - A_{n-1}(i+1,j)A_{n-1}(i,j+1). \qquad \text{(Lewis)}$$

The desired determinant is $A_n(1,1)$. In many cases, $A_n(i,j)$ turns out, conjecturally at first, to have an explicit expression, involving single and double products. Whenever this is the case the proof of the conjectured evaluation is completely routine, by induction on n, by checking that (Lewis) is satisfied by that conjectured expression, and by checking the trivial initial conditions for $n = 0$ and $n = 1$. Finally, to get an explicit expression for the original determinant, all one has to do is plug in $i = 1$ and $j = 1$.

In [AE2] this method was used to get computer-assisted proofs of numerous determinant identities. But my goal is to make things *completely* automatic, and human-free.

To keep things simple (after all, the main point here is to introduce a *research methodology*, not to find exciting new results), let us focus on Hankel matrices, which have the form $(h(i+j))$ for some sequence $h(r)$ (that for us would have to be an explicit expression).

So given a discrete function $h(r)$ (say a hypergeometric sequence), we have the *General Problem* of evaluating the $n \times n$ determinant

$$A(n,r) := \det(h(r+i+j)), \quad 0 \leqslant i,\ j \leqslant n-1. \qquad \text{(Hankel)}$$

Even if we are only interested in $A(n,0)$, we still need the r, as will become apparent shortly. For Hankel matrices (Hankel), Dodgson's rule specializes to:

$$A(n,r) = \frac{A(n-1,r)A(n-1,r+2) - A(n-1,r+1)^2}{A(n-2,r+2)}. \qquad \text{(HankelDod)}$$

Now, in some sense, this is *already* an answer, since it displays $A(n,r)$ in the ansatz of double sequences satisfying *partial non-linear recurrence equations with constant coefficients*. Indeed since $A(0,r) = 1$ and $A(1,r) = h(r)$, (HankelDod), gives a quick way to crank out the sequence $A(n_0,r)$ for $n_0 = 0,1,2,\dots,N_0$ for any desired N_0. Of course, one can argue that the very definition is already an 'answer' just declare the class of determinants of hypergeometric determinants a legitimate ansatz! But we would like to do better.

Inspired by the Hilbert matrix, for which $A(n,0)$ turned out to be hyperhypergeometric in n, it turns out (experimentally, at first), that $A(n,r)$ also has this property for each r. Not only that, the ratio-of-ratios $(A(n,r)/A(n-1,r))/(A(n-1,r)/A(n-2,r))$ is not only a rational function of n, but of *both* n and r. Furthermore, it also turns out that it is also hyperhypergeometric in r, i.e., $(A(n,r)/A(n,r-1))/(A(n,r-1)/A(n,r-2))$ is another rational function of (n,r). Finally the 'mixed-ratio' $(A(n,r)/A(n-1,r))/(A(n,r-1)/A(n-1,r-1))$ is also a rational function of (n,r).

A new ansatz is Born: hyperhypergeometric double-sequences

Definition. A double sequence $B(n,r)$ is *hyperhypergeometric* if the three discrete functions

$$B_{11}(n,r) := \frac{B(n,r)B(n-2,r)}{B(n-1,r)^2}, \qquad B_{12}(n,r) := \frac{B(n,r)B(n-1,r-1)}{B(n-1,r)B(n,r-1)},$$
$$B_{22}(n,r) := \frac{B(n,r)B(n,r-2)}{B(n,r-1)^2},$$

are all rational functions of (n,r). Hence, hyperhypergeometric double-sequences may be identified with *triples of rational functions* (B_{11}, B_{12}, B_{22}) satisfying the obvious *compatibility conditions*:

$$\frac{B_{11}(n,r)}{B_{11}(n,r-1)} = \frac{B_{12}(n,r)}{B_{12}(n-1,r)}, \qquad \frac{B_{22}(n,r)}{B_{22}(n-1,r)} = \frac{B_{12}(n,r)}{B_{12}(n,r-1)}. \qquad \text{(Compatibility)}$$

In addition we have to specify the initial conditions $b_{00} = B(0,0)$, $b_{01} = B(0,1)$, $b_{10} = B(1,0)$.

So suppose you have a conjectured hyperhypergeometric expression $B(n,r)$ for the family of Hankel determinants $A(n,r) := \det(h(r+i+j))$, $0 \leqslant i, j \leqslant n-1$. By Dodgson's rule, it is enough to verify that $B(0,r) = 1$, $B(1,r) = h(r)$, and

$$B(n,r) = \frac{B(n-1,r)B(n-1,r+2) - B(n-1,r+1)^2}{B(n-2,r+2)}. \qquad \text{(HankelDod}')$$

By taking ratios, this is equivalent to, in terms of the rational functions B_{11}, B_{12}, B_{22}:

$$\frac{B_{12}(n-1,r+2)B_{12}(n-1,r+1)}{B_{11}(n,r)} - \frac{B_{12}(n,r+2)B_{12}(n-1,r+2)}{B_{22}(n,r+2)B_{11}(n,r+1)} = 1, \qquad \text{(VerifyHankelDod)}$$

which Maple (or Mathematica, etc.) can verify routinely, and hence *prove* the conjecture.

But what about *discovering* the identity in the first place? *Can a computer do that*? *All by itself*?

You bet it can! Now that we have a well-defined *haystack*, the ansatz of hyperhypergeometric double-sequences, we can let the computer compile a table of $A(n,r)$ for $n,r \leqslant L$ for some finite L, either by using the determinant definition, or more efficiently, by using (HankelDod), starting with $A(0,r) = 1$, $A(1,r) = h(r)$. Then we let our beloved computer compute the iterated ratios

$$A_{11}(n,r) := \frac{A(n,r)A(n-2,r)}{A(n-1,r)^2}, \qquad A_{12}(n,r) := \frac{A(n,r)A(n-1,r-1)}{A(n-1,r)A(n,r-1)},$$
$$A_{22}(n,r) := \frac{A(n,r)A(n,r-2)}{A(n,r-1)^2},$$

for $2 \leqslant n, r \leqslant L$.

Then, assuming that $A(n,r)$ is indeed hyperhypergeometric, we put B_{11}, B_{12}, B_{22} in generic form for rational functions in (n,r) with *undetermined coefficients*, where the top and bottom of each are generic polynomials of a guessed degree d. Now by plugging-in, we have $(L-1)^2$ equations

$$B_{11}(n_0,r_0) - A_{11}(n_0,r_0) = 0, \quad 2 \leqslant n_0, r_0 \leqslant L.$$

Clearing denominators, and setting the numerator equal to zero, will give us a system of *linear* equations in the unknown 'undetermined' coefficients of B_{11}. Similarly for B_{12} and B_{22}. If the computer finds a solution, then we are done! If it does not, we can make the guessed degree one higher, and try again. We can keep upping the degree until we succeed or give up. Of course, no one said that $A(n,r)$ must be hyperhypergeometric, it was only our *conjecture* that it might. So humans still have to *decide* which ansatzes to try, but once that decision is made, and the program already exists, the computer does *everything* from α to ω: conjecture the expression $B(n,r)$ (in its equivalent form as the triple of rational functions $B_{11}(n,r)$, $B_{12}(n,r)$, $B_{22}(n,r)$), and then proves it, all by itself! Finally, it also verifies the compatibility conditions (Compatibility), which also consist of routine manipulations of rational functions.

Toeplitz determinants

As far as I know, Otto Toeplitz was not a Leipziger, and hence it is unlikely that he ever bumped into my grandfather Paul in the street or cafeteria. But he is still dear to me, in part because his widow was my nanny between the ages of 0 and 1, and in part because I like his determinants, that have the form $\det(h(i-j))$. All we said above about Hankel determinants carries over, with obvious modifications, to Toeplitz determinants. For details see the source-code in the Maple package `CLD`, described below.

A user's manual for the Maple package CLD

(CLD stands for Charles Lutwidge Dodgson.) First download it from my website, by going to my homepage (search Google for "Zeilberger" (or even for "Doron") or type http://www.math.rutgers.edu/~zeilberg/) then click on `programs`, then click on `CLD`. Alternatively, just download http://www.math.rutgers.edu/~zeilberg/tokhniot/CLD.

Once my Maple package `CLD` is in your own computer, stay in the same directory, go into Maple, by typing `maple`, or `xmaple`, or by clicking on the Maple icon. Once in

Maple, type: `read CLD` (if you decide to go to a different directory, you need the full path name of the file `CLD`).

Now, all you have to do is follow the *on-line* instructions. In particular, typing `ezra();` will give you a list of all the main procedures, i.e., those that you are likely to use. Typing `ezra1();` will give a list of *all* procedures, so that *you* can understand what is going on, and will be able to improve and extend this rudimentary program to more general classes of determinants and to explore other, more general, or completely different, ansatzes.

The main procedures are `EvalH` for the automatic discovery and proof of Hankel-determinant-evaluations, and the Toeplitz analog `EvalT`. These give you the output in terse style. If you want a math paper, ready for submission, use the *verbose* versions, `EvalHpaper` and `EvalTpaper`.

For example, typing the 28-character string `EvalHpaper(r!*(m-r)!,n,r4)`: would, after a couple of minutes (on my rather slow computer), output a paper that does `ALL` the following steps, previously done by humans, with only some machine help, that lead to [AE1].

1. Conjecture the expression (this was first done by the smart humans Greg Kuperberg and Jim Propp).
2. Prove it, by human-machine interaction (previously done by Human Tewodros Amdeberhan and Machine Shalosh B. Ekhad).
3. Write up the paper [AE1] for publication (formerly done by the human-partner of the A–E collaboration).

For the celebrated MacMahon determinant [M] (very important in plane-partition enumeration, first proved by the great Percy MacMahon, and then reproved by me [Z1], using Dodgson's rule (with help from Ekhad)), type: `EvalTpaper(1/(m+r)!,n,r,4):`.

Finally, for a completely automated performance of all the phases of mathematical activity: conjecture-proof-writing-it-up, for the closed-form evaluation of the Hilbert matrix, type: `EvalHpaper(1/(r+1),n,r,4):`.

Sample input and output files

The webpage of this paper (clickable from my homepage) contains sample input and output files.

How to be immortal

Dying is a stupid reason to stop publishing. If you are lucky, someone might find your unfinished work, finish it up, and publish it as joint work (like Bruce Berndt did to B.M. Wilson). But this will, at best, get you at most one or two posthumous papers. What if you want to keep on publishing papers for ever? Easy! First make your system-administrator promise not to close your account after your demise. Then in the Maple package, have a 'Unix-escape' shell-program that submits the paper to one of the many electronic journals.

It might be a good idea to sign an inclusive copy-right-transfer form for all your future submissions.

Now write an *infinite* do-loop, with increasingly more complicated determinants to be evaluated. Most of them will turn out not to fit the given ansatz (in this case the hyperhypergeometric ansatz), but whenever it does, and the computer succeeded in conjecturing, and then, automatically proving it, the computer can completely automatically, also do the *submission*.

You can do even more! Suppose that after 100 papers on this subject, the editor finally decides to reject your 101th posthumous paper, because it is 'not interesting.' Then you can automatically send an angry rebuttal. The variations are endless.

Suggestions for further work

It should be relatively painless to do the q-analog of this, and to also deal with determinants of general matrices $(a(i, j))$ $(1 \leqslant i, j \leqslant n)$, i.e., not necessarily Hankel or Toeplitz. Now we would have a 3-parameter discrete function $A(n, r, s) := \det(A(i + r, j + s))$, $0 \leqslant i, j \leqslant n - 1$, and the appropriate ansatz would be hyperhypergeometric sequences for *triple* sequences.

Recall that $B(n, r)$ is *hyperhypergeometric* means that $B(n, r)/B(n, r - 1)$ and $B(n, r)/B(n-1, r)$ are hypergeometric. This naturally leads to the more general ansatz for which the above two ratios are P-recursive. Even more generally we can consider solutions of linear recurrence whose coefficients are P-recursive (holonomic). I am sure that with these more general ansatzes, many more determinants will be computer-evaluable.

Why is this exciting?: the medium is the message!

With all due respect to the *substance* of this research, i.e. determinant-evaluation, what makes this endeavor *so* exciting is the *form* and the *research methodology*, of doing *purely theoretical* and *completely rigorous* mathematics using *experimental methods*. Of course, these are just crude and clumsy beginnings, but as we, and the computer, will get more experienced, this methodology will be applicable to proving *Goldbach, RH, Navier–Stokes*, etc. Paraphrasing Archimedes, all I need to know is the *Right Ansatz* and my computer will prove the Riemann Hypothesis.

As already mentioned before, often we are stumped because we do not have enough parameters. In the present humble case, it was impossible to evaluate the determinant of the $n \times n$ Hilbert matrix $(1/(i + j - 1))$, because it only depended on the single parameter n, but evaluating the more general $\det(1/(r + i + j - 1))$ was possible, since this enabled induction on n and r.

So the reason that my computer is unable, at this time, to prove the Riemann Hypothesis, is that $\zeta(s)$ only depends on one variable. With an appropriate generalization, belonging to an appropriate ansatz, it would be doable.

Building up experiments is even more important than performing them

Physical science consists of an interplay of *theory* (hypothesis-forming) and *experiments* (hypothesis testing). But in order to perform experiments, one has to *design* them, and *build instruments*. So ultimately, the most important members of the scientific community are neither theorists nor experimentalists, but *engineers* and *technicians*, who build the instruments needed to carry out the experiments.

The analog of scientific instruments are *software*. My role in this project was neither *hypothesizing*, i.e., conjecturing, nor, *experimenting*, it was all done by my computer. All I did was 'build the equipment,' i.e., design an algorithm and implement it, that is, write the program. But this would not have been possible without the computer algebra system, and its associated *programming language*, (Maple in my case), developed through many years of dedicated labor by such pioneers as Keith Geddes and Gaston Gonnet. And of course, even more importantly, the meta-equipment, i.e., the computer itself, that is the *hardware*.

Eventually, such computer programs will also be written by computers, thanks to meta-programs and future meta-Maple. But hopefully there will still be a role for humans, in thinking up new ansatzes and *meta-ansatzes*, and in finding the *trivializing generalization*. But then again, eventually computers will learn how to do these things all by themselves, and we might not be able to even follow the general *drift* of what they are doing, because of its immense complexity.

I am even more like my grandfather Paul than I thought before

My grandfather Paul Alexander was a chemist, hence an experimental scientist. However he was not a 'pure' scientist, but rather an 'applied' and 'industrial' one, who designed *verfahrens* to do specific tasks, in his case, recycling rubber. In my case too, I am not interested in probing the 'nature of mathematics' per se, only in designing *algorithms* do to specific tasks. That specific task happens to be discovering and proving mathematical facts, but the 'recycled rubber' itself (i.e., mathematical theorems) are much less exciting than the process (i.e., computer program) that generated them.

Acknowledgments

I am very grateful to Wolfgang Hackbusch for kindly accepting my offer to give a seminar talk here. I am also very thankful to all of you for coming!

References

[AE1] T. Amdeberhan, S.B. Ekhad, A condensed condensation proof of a determinant evaluation conjectured by Greg Kuperberg and Jim Propp, J. Combin. Theory Ser. A 78 (1997) 169–179.

[AE2] T. Amdeberhan, S.B. Ekhad, Determinants through the looking glass, Adv. Appl. Math. 27 (2001) 225–230.

[D] C.L. Dodgson, Condensation of determinants, Proc. Roy. Soc. London 15 (1866) 150–155.

[H] W. Hackbusch, From classical numerical mathematics to scientific computing, Documenta Mathematica, extra volume, ICM, Berlin, Vol. 1, 1998, pp. 235-254, based on a plenary talk.

[M] P.A. MacMahon, Combinatory Analysis, Cambridge Univ. Press, 1918; Vol. 2, reprinted by Chelsea, 1984.

[Z1] D. Zeilberger, Reverend Charles to the aid of Major Percy and Fields Medalist Enrico, Amer. Math. Monthly 103 (1996) 501–502.

[Z2] D. Zeilberger, Dodgson's determinant evaluation rule proved by TWO-TIMING MEN and WOMEN, Electron. J. Combin. 4 (2) (1997) R22.