The justification of mathematical statements

BY PETER SWINNERTON-DYER

University of Cambridge, Wilberforce Road, Cambridge CB3 0WB, UK (h.p.f.swinnerton-dyer@dpmms.cam.ac.uk)

The uncompromising ethos of pure mathematics in the early post-war period was that any theorem should be provided with a proof which the reader could and should check. Two things have made this no longer realistic: (i) the appearance of increasingly long and complicated proofs and (ii) the involvement of computers. This paper discusses what compromises the mathematical community needs to make as a result.

Keywords: proof; justification; programs; conjectures

I approached this conference with a seriously split personality. I am usually regarded as a number theorist, and therefore, as a pure mathematician of the most uncompromising kind. On the other hand, I also work at the more vulgar end of the study of ordinary differential equations; indeed for years I was the only pure mathematician in Cambridge who had a visa to enter the Department of Applied Mathematics. And for a substantial part of my career I was employed not as a mathematician but as a computer scientist. In these three roles, my attitudes to what should be regarded as a proof have been quite different.

In the real world, what is regarded as an adequate proof depends very much on what one is trying to prove. It takes far less evidence to convict a person of speeding than to convict him (or her) of murder—and nowadays it appears that even less evidence is needed to justify waging war. In mathematics we need to accept (and indeed have tacitly accepted) the same diversity. We have an ideal concept of what is meant by a rigorous proof, but in many contexts we cannot afford to live up to that standard; and even before the days of computers, mathematicians had devised various ways of loosening that straightjacket. Moreover, the amount of effort which the mathematical community puts into checking a purported proof depends very much on the importance, the unexpectedness and the beauty of the result.

The most demanding standard that I have ever encountered was that impressed on me by J. E. Littlewood, who was my research supervisor. He maintained that because so many published papers contained errors or at least gaps, one should never make use of someone else’s theorem unless one had checked the proof oneself. He was of course conditioned by having lived through the traumatic process of making classical analysis rigorous; and for most of his lifetime there were important branches of pure mathematics based more on plausibility than on certainty. But if such a doctrine was ever feasible it is certainly no longer so. The final death-blow to it may well have been the
classification of the finite simple groups. But until the advent of computers, most pure mathematicians had no difficulty with the concept of a rigorous proof. To quote A. E. Houseman in a rather different context: ‘A terrier may not be able to define a rat, but a terrier knows a rat when he sees it.’

Not all mathematical statements are theorems. In most branches of pure mathematics there is a border-zone between what is rigorously established and what is totally mysterious. That zone is populated by what are variously called conjectures, hypotheses and open questions. If one were asked for criteria which justify making a particular conjecture, one might say that it must satisfy one or more of the following conditions:

(i) It sheds new light on the structure of the subject.
(ii) Its statement is simple and fascinating.
(iii) It is plausible and enables mathematicians to prove important results which they are currently unable to prove without it.
(iv) It can be shown to hold in particular (but preferably typical) cases.

These reasons are listed in what seems to me the order of decreasing merit. All but the second of them also tend to support the truth of the conjecture. In number theory the border-zone is particularly rich, and I shall take my examples from there.

Two of the Clay Institute’s Million Dollar problems fall within number theory. The original Riemann Hypothesis was that the non-trivial zeroes of the Riemann zeta function $\zeta(s)$ all lie on the line $\Re s = 1/2$. Riemann’s reasons for believing his Hypothesis (of which a good account can be found in Siegel (1932)) were sophisticated, and he probably had no computational evidence for it. Subsequently, more than a billion zeroes have been computed (mostly funded by the Pentagon), and they all lie on the critical line; but there are strong reasons for believing that even if counterexamples to the Riemann Hypothesis exist they will be rare and will have very large imaginary parts, so computers cannot provide strong evidence for it. Littlewood, indeed, believed that the Riemann Hypothesis was false, on the grounds that if it were true the combined efforts of classical analysts would have proved it long ago. But in my view this is to see it in the wrong context. The Riemann Hypothesis has been repeatedly generalized, and the more far-reaching the generalizations the more central they appear to be to the structure of modern number theory. Thus, the Riemann Hypothesis ought not to be regarded as lying within classical analysis, and one ought not to hold it against classical analysts that they have not yet provided a proof of it.

In its simplest form, the Birch/Swinnerton-Dyer conjecture for an elliptic curve relates the value of the associated $L$-series at $s=1/2$ (the mid-point of the critical strip) to the order of the Tate-Safarevic group of the curve. (The clearest detailed description of the conjecture can be found in Tate (1966).) As Tate said, ‘it relates the value of a function at a point where it is not known to exist to the order of a group which is not known to be finite’. Nevertheless, even when the conjecture was first formulated it was possible to provide numerical evidence in support of it in particular cases—primarily when the elliptic curve is defined over $\mathbb{Q}$ and admits complex multiplication, for in that case the $L$-series can be analytically continued and can be explicitly evaluated at $s=1/2$. Moreover, if the Tate-Safarevic group is finite its order is known to be a square; and even forty
years ago its $p$-component could be evaluated in principle for any prime $p$ and nearly always in practice for $p=2$. Over the last forty years a lot more numerical evidence has been obtained, and special cases of the conjecture have been proved—in contrast with the Riemann Hypothesis, which still appears absolutely impregnable. It has also been vastly generalized, though it is not clear to me that these generalizations are supported by any additional evidence.

Fermat’s Last Theorem fell into the category of conjectures until the work of Wiles. I am sure that Fermat believed he had proved it; and indeed one can with fair confidence reconstruct his argument, including one vital but illegitimate step. It satisfies the second of my four criteria, but none of the others, and it has not fascinated everybody. Gauss, when asked why he had never attempted to prove it, replied that he could set up a hundred such statements, which could be neither proved nor disproved and which served only to impede the progress of mathematics. But it was the attempt to prove Fermat’s Last Theorem which motivated Kummer to create algebraic number theory—a rich garden to grow from a single seed.

I should also mention a conjecture which turned out to be false and which satisfied none of my criteria, but from which important developments sprang. It has been known since Dirichlet that a quadratic equation defined over $\mathbb{Q}$ is soluble in $\mathbb{Q}$ if it is soluble in each completion $\mathbb{Q}_p$ and $\mathbb{R}$. (The corresponding result over a general algebraic number field, which is much harder to prove, is due to Hasse; so for any family of equations a result of this kind is known as a Hasse Principle.) Mordell conjectured that the corresponding result would hold for the equation of a non-singular cubic surface. He gave no reason for this, and I suspect that he put forward the conjecture partly for probabilistic reasons and partly because he could think of no other obstruction. The first counterexample depended on the sheer cussedness of small integers and threw no light on the nature of a possible obstruction in general. The second one was provided by Cassels & Guy (1966). It depended on a computer search, extensive by the standards of the time, which generated a list of diagonal cubic equations, which had no small integer solutions; but the proof that the simplest equation in this list was actually insoluble did not involve a computer. These counterexamples led Manin to discover the Brauer-Manin obstruction, which plays a central role in the modern theory of Diophantine equations.

I could go on. But I hope that I have done enough to demonstrate two things: first, that at least in some branches of pure mathematics the formulation of well-justified conjectures plays an important role in advancing the subject and second, that there is general agreement what ‘well-justified’ means in this context.

Both for theorems and for conjectures, one should make a distinction between structural statements such as the Riemann Hypothesis and accidental statements such as Goldbach’s conjecture. This distinction is not clear-cut; there would be disagreement, for example, about the description of the Four Colour theorem or the classification of finite simple groups. (I regard the latter as accidental, because there are so many sporadic simple groups and they are so diverse.) Most mathematicians are resigned to the likelihood that the proofs of some accidental theorems may sometimes be long, turgid and ill-motivated; but they expect that the proof of a structural theorem, even if it is long and difficult, will be in some sense straightforward.
The situation with differential equations is very different. It is true that there are theorems about differential equations which have been rigorously proved, but these tend not to answer the questions which users of differential equations actually ask. The typical situation is as follows: Consider some interesting real-world system. It is in principle possible to write down the differential equations which describe how the system varies with time, according to the laws of nature as currently understood (and ignoring the effects of the Uncertainty Principle); but these equations will be far too complicated to use as they stand. One therefore needs to make radical simplifications, hoping that the solutions of the simplified model will still describe to a good approximation the behaviour of the original system. Currently, this process seems to be a matter of pure faith; but for some systems there may be scope for a rigorous treatment. For example, in the Million Body Problem, which studies the interaction of the stars in a galaxy, the stars are treated as point masses satisfying the Newtonian laws of gravitation, though each star does gradually radiate away some of its mass. Again, Nosé managed to reduce the thermodynamics of the universe to three first order ordinary differential equations.

To this simplified system one applies whatever tools seem appropriate. For the Nosé equations (or the better known and more studied Lorenz equations, which are another system of three first order equations derived in much the same way) these are of three kinds:

(i) Genuinely rigorous arguments.
(ii) Arguments whose conclusion takes the form ‘Such-and-such happens unless there is a rather unlikely-looking coincidence.’
(iii) Information about particular trajectories, obtained numerically.

These will not be enough to determine the behaviour of the system even qualitatively; but among the possible qualitative descriptions compatible with the information obtained there will usually be a simplest one—and an appeal to Ockham’s Razor should lead us to adopt that description. This process must be regarded as a justification of the conclusion rather than a proof of it; but for differential equations there seems little prospect of ever being able to do better.

So far I have mentioned computers only peripherally. I must now turn to the issues raised by computer-based proofs; and here it is necessarily to take account of the fallibility both of computers and of programmers. Most computers probably have bugs even in their hardware. (One early fixed-point machine evaluated \((-1) \times (-1)\) as 0 owing to a very plausible design fault; fortunately that was an operation not often performed.) Probably all computers have bugs in their software—their operating systems, assemblers and compilers—though if these have not been detected and cured that implies that they very seldom cause trouble. More importantly, the process of turning an algorithm into a computer program is notoriously fallible even when the program is in a high-level language. Moreover, although it is feasible for a referee or reader to check an algorithm for errors, it is almost impossible to check that someone else’s program is correct. (Some computer scientists claim that there exist programs which will rigorously check whether an algorithm has been correctly translated into a program, and there do exist similar programs which check that the design of a microchip correctly implements its specification. But no one has yet used these methods to
check the correctness of any of the existing programs for proving the Four Colour theorem, and this can hardly be because computer scientists do not think that theorem important enough.) All this needs to be taken into account in deciding the level of credibility of a computer-based or computer-assisted proof.

That last phrase covers a considerable diversity of computer involvements, and they need to be separated. Let me start with what might be called the computer’s role as conjurer’s assistant. When you meet the word ‘Consider’ in a proof, you know that a rabbit is about to be pulled out of a hat; but you are unlikely to be told, nor logically do you need to be told, where the rabbit came from. We have already had one example of this. When Cassels thought of a method, which might prove that some pre-assigned diagonal cubic equation was a counterexample to the Hasse Principle he needed an equation for which the method might work; and the only way of finding such an equation was by a computerized search. Again, it is known that the set of rational points on an elliptic curve defined over \( \mathbb{Q} \) form a finitely generated abelian group. Its torsion part is easy to compute, so what is of interest is to find its rank. For any given curve, there is a standard method for finding an upper bound \( r \) for this rank; and empirically the upper bound thus obtained is usually the actual rank. To prove this for a particular curve, we have to find \( r \) independent rational points on the curve, and this is done by means of a search program. Once such points have been obtained, it is a relatively simple task to prove that they lie on the curve—and even if a computer is used to check this, it is most unlikely to report that a given point lies on the curve when in fact it does not.

The canonical structure of a proof, as exemplified in that unreadable tour-de-force Russell and Whitehead’s *Principia Mathematica*, is as follows. One starts with a finite collection of well-established statements. At each step one writes down one more statement, which must be an immediate logical consequence of the existing statements. Eventually one generates in this way the result which one is trying to prove. There is some resemblance here to a chess-playing program, in that there is an enormous choice at each step and it is essential to have a good evaluation function to tell one which is the most helpful next statement to adjoin. To the extent that this scheme could ever be made to work on a computer, the process would generate proofs which could be checked by a flesh-and-blood mathematician—or indeed by a proof-checking program. The latter would of course be far easier to write than a theorem-proving program because it would not need the evaluation subroutine, which is where (as with a chess-playing program) the fundamental difficulties occur. I myself do not believe that a theorem-proving program of this kind will ever prove theorems which are beyond the capacity of a human being. But the ideas which it would need could be applied, in a much simplified form, to a program to referee papers written by human beings; for such papers contain small gaps in the logic which the reader is expected to fill, and filling such gaps is theorem-proving of a very elementary kind. Any editor will bear witness to the need for such a program.

So far I have been dealing with cases where a computer is essential or at least useful in generating a proof, but is not needed in the proof itself—in other words, proofs which are computer-assisted rather than computer-based. I now turn to those which are genuinely computer-based. Let me give two examples. The first is the Four Colour theorem, for which two computer-based proofs have already been constructed, of which at least one is generally accepted as valid.
(For an account of the second proof, with many references, see Robertson et al. (1997); a summary of this paper is also available on the Web.) The second is a conjecture which has not yet been proved, largely because it belongs to a branch of mathematics which is not currently fashionable; but it is well within reach of modern desk-top computers and it illustrates the points I want to make. Recall that a lattice $L$ in Euclidean space is said to be admissible for an open set $R$ if no point of $L$ except the origin $O$ lies in $R$. Then the assertion is that every lattice admissible for the region $|X_1X_2X_3X_4|<1$ has determinant at least $\sqrt{725}$. (This is best possible if true, for the lattice $L_0$ of integers of the totally real quartic field of discriminant 725 is certainly admissible.)

As these two examples show, a large part of a computer-based proof may be devoted to vulgar numerical calculation, but this will not always be so. Such a part presents few difficulties for checking correctness. Calculation with integers is exact, though calculation with real numbers is not. In the latter case one must take account of round-off errors, and this requires working with inequalities rather than with equalities. Where serious difficulties do occur is if processes from numerical analysis are involved: it is, for example, almost impossible to generate bounds for the solution of a differential equation which are both tight and rigorous. This is a further reason for what I said earlier, that in the study of differential equations one must accept much lower standards of justification than in most of pure mathematics.

In a simplified form, the algorithm which is expected to prove the lattice assertion above is as follows. We look for all admissible lattices $L$ which satisfy say $\det L<27$. By standard theory, it is enough to consider lattices which contain the point $P_1=(1,1,1,1)$. No admissible point, and hence in particular no point of the lattice other than $O$, is a distance less than 2 from $O$; so standard theory provides an explicit constant $C$ such that there are lattice points $P_2, P_3, P_4$ within a distance $C$ of the origin which together with $P_1$ generate the lattice.

We can think of the lattice as described by the point $L=P_2\times P_3\times P_4$ in 12 dimensions, and information about the lattice is equivalent to information about the set in which $L$ lies. The admissible region $|X_1X_2X_3X_4|\geq1$ is the union of 16 disjoint convex subregions, according to the signs of the $X_i$; we initially split cases according to which region each of the $P_j$ lies in. Some of these cases can be immediately shown to be impossible: for example, if all the coordinates of $P_2$ are positive then it turns out that $P_1-P_2$ cannot be admissible. More generally, for any particular case we choose integers $n_1, \ldots, n_4$ not all zero and consider the lattice point $P=\sum n_j P_j$. (The design of an efficient algorithm for choosing the $n_j$ is the one sophisticated part of the program.) There are now three possibilities:

(i) $P$ cannot lie in any of the 16 admissible subregions; if so, this case can be deleted.

(ii) There is exactly one subregion in which $P$ can lie; if so, this is a constraint on $P$ and, therefore, reduces the set in which $L$ can lie. We can now continue the process with a new choice of the $n_j$.

(iii) There is more than one subregion in which $P$ can lie; if so, we split this case into subcases according to the subregion in which $P$ is assumed to lie.

Thus the process keeps on deleting old members from the list of cases to be studied but also putting new ones in. What we hope is that eventually the list
reduces to a single case and for that case the open region containing $L$ is small
and contains the point $A_0$ corresponding to the conjectured critical lattice $L_0$;
if so, we can complete the proof by means of a known isolation theorem. If this
does not happen, in due course we obtain a list of very small regions in one of
which $L$ must lie; and provided the process is error-free we expect each of these
regions to provide an admissible lattice which can be found by hand.

The algorithm which underlies the proof of the Four Colour theorem fits the
same pattern—and indeed this appears to be the natural pattern for a long and
complicated computer-based proof. Here too the proof starts with a finite list of
cases, and when any particular case is processed it is either deleted or replaced by
a finite number of subcases. For if the theorem is false, among the maps which
cannot be coloured with only four colours there is one which contains the
smallest number of regions. The list of cases is a list of sub-maps which might
form part of this map. A case can be split by adjoining an extra region to the sub-
map, which can be done in various ways. A case can be rejected if there is a
different sub-map having fewer regions such that if the old map cannot be
coloured with only four colours, then nor can the new map obtained by replacing
the old sub-map by the new one. (Fortunately, this is a property which can often
be established without knowing anything about the rest of the map.) The proof
succeeds if the list can be exhausted.

The principle underlying such proofs is attributed to Sherlock Holmes: ‘When
you have eliminated the impossible, whatever remains, however improbable, must
be the truth.’ The point which I wish to make about computer-based proofs of this
kind is as follows. Suppose that there are errors in the program, but the program
does in fact terminate; since that was the result which we were expecting, we have no
reason to doubt the correctness of the program—for programming errors are usually
only detected because they have led to surprising results. Moreover, in a program of
this kind an error is likely to lead either to some cases being wrongly rejected or to
some cases never being generated by the splitting process. Either of these will make
the program terminate sooner than it should have done, or even when it should not
have terminated at all. In other words, errors will usually generate false proofs
rather than merely failing to generate true proofs. It is this which makes validation
of this kind of proof so important.

More than thirty years ago I stated what I thought was needed to validate a
computer-based proof, within the limits of practicality; and I see no reason to
change my views now. (I was heartened to discover at this conference that
Annals of Mathematics has been forced to adopt a very similar attitude.)
Suppose that Alice has produced a computer-based proof and wishes Bob to
validate it; what should each of them do?

Alice should publish the algorithm which underlies the program, in so simple a
form that other people (and in particular Bob) can check it. She should be very
cautious about including in the algorithm the sort of gimmicks which make the
program more efficient, because they also make the correctness of the algorithm
harder to check. It is highly desirable, if possible, that the algorithm should also
specify some intermediate output. Alice should not at this stage provide Bob
with any other information; in particular she should not give Bob a copy of her
program or any part of it, nor a copy of her intermediate output. Ideally, Bob
should not even come from the same environment as Alice, because that would
tend to give them a common mind-set. Bob should then turn the algorithm into
a program, preferably not using the same language as the one which Alice used. If both programs yield the same results, including the same intermediate output, this is as much validation as can reasonably be provided.

Finally, a more general point. Manichaeans hold that power over the universe is equally divided between God and the Devil. At least until Gödel, mathematicians believed that their subject lay entirely within God’s share. It is my impression that most of the speakers at this conference still hold this view, even though much of what they have said points in the opposite direction. The doctrine is well illustrated by two couplets written nearly three centuries apart, the second being written as an answer to the first:

Nature, and Nature’s laws, lay hid in night;  
God said ‘Let Newton be! ’ and all was light. But not for long; the Devil, shouting ‘Ho!  
Let Einstein be! ’ restored the status quo.

Appendix A

This appendix provides some further information about some of the topics mentioned in the body of the talk.

(i) The most general version of the Riemann Hypothesis which I know is as follows. Let \( f(s) \) be a Dirichlet series satisfying the following conditions:

- (a) It occurs naturally in a number-theoretic context.
- (b) It has an Euler product.
- (c) It can be analytically continued as a meromorphic function over the whole \( s \)-plane, and satisfies a functional equation which relates \( f(s) \) and \( f(n-s) \) for some integer \( s \) and which up to sign is tantamount to a symmetry law.

Then all the non-trivial zeroes of \( f(s) \) lie on the critical line \( \Re s = (1/2)n \).

This as it stands appears to contain an escape clause, in that the first condition is metamathematical rather than mathematical. But in practice there would be little disagreement whether a purported counterexample satisfied that condition or not.

(ii) Fermat’s last theorem asserts that if \( n>2 \) there are no solutions in positive integers of \( X^n + Y^n = Z^n \). Gauss clearly regarded it as an accidental rather than a structural theorem; but the heart of Wiles’s proof is a proof of a weak form of the modularity conjecture, which is certainly a structural theorem.

(iii) The modularity conjecture (over the attribution of which controversy rages) states that each elliptic curve defined over \( \mathbb{Q} \) can be parametrized by modular functions. The first assertion of this kind is due to Tamagawa, who died young. The first substantial justification of it was given by Weil (1967), though he stated it only as an open question. The first substantial numerical evidence for it was given by Birch and his students. It has now been completely proved.

(iv) Goldbach’s conjecture is that every positive even integer other than 2 is the sum of two primes. It has been proved for all even integers up to \( 6 \times 10^{16} \). This is a case in which a purported proof of the full conjecture would deserve very careful checking, but the proof of the weaker statement in the previous sentence deserves rather little.
(v) The Nosé equations are
\[ \dot{x} = -y - xz, \quad \dot{y} = x, \quad \dot{z} = \alpha(x^2 - 1), \]
where \( \alpha \) is a positive parameter. There are certainly values of \( \alpha \) for which the behaviour of the trajectories is chaotic both in the usual and in the technical sense; whether this is so for all values of \( \alpha > 0 \) is not known. The Lorenz equations are
\[ \dot{x} = \sigma(y - x), \quad \dot{y} = rx - y - xz, \quad \dot{z} = xy - bz, \]
where \( \sigma, r \) and \( b \) are three real positive parameters. A good introduction to their study can be found in Sparrow (1982).

(vi) William of Ockham (or Occam) was a medieval theologian and philosopher. He stated the principle that ‘entities should not be multiplied without cause’, which is known as Ockham’s Razor. A reasonable paraphrase would be that one should accept the simplest explanation of any phenomenon.

(vii) The simplest base for the rational points on an elliptic curve usually consists of points with numerator and denominator comparable with the coefficients in the equation of the curve; but occasionally this fails badly. For example, the group of rational points on the curve \( y^2 = x^3 - 673 \) has rank 2, and the simplest generators are the points with
\[ x = 29 \quad \text{and} \quad x = \frac{3398323537}{61761^2}. \]
A large table of ranks and generators can be found in Cremona (1997).

References


Tate, J. 1966 On the conjectures of Birch and Swinnerton-Dyer and a geometric analog. Sém. Bourbaki 306.


Discussion

C. Jones (Computing Science Department, University of Newcastle, UK). The view that one might prefer to construct a second program (rather than study a carefully annotated one) is odd. It could be compared to a journal, which only
sends the statement of a new theorem to referees asking them to provide their own proofs. This might uncover errors but would be rather wasteful! The assertions in a program provide a rigorous argument of its correctness; or careful development using for example data abstraction is even more like the (rigorous) proof of a theorem.

R. D. ARTHAN (Lemmal Ltd, Berkshire, UK). Direct evaluation of programs within a theorem-proving environment such as HCL offers a good half way house between relying on an untrusted program and formal program verification. This has been used with some success by John Morrison and others giving validated calculations with the real numbers. Can you comment on this?

P. SWINNERTON-DYER. The question is what degree of credibility should attach to a theorem none of whose proofs conform to classical standards. This question is usually asked about proofs which depend on computer programs (as in these two questions), and this answer will deal only with those. Even if a computer program in a high-level language is itself correct, the results obtained by running it may be vitiated by undetected bugs in the compiler, the operating system or even the hardware—or indeed by viruses temporarily present in the computer being used. (Few if any compilers or operating systems are without bugs; and in this paper, I gave an example of a hardware error in an important computer, which to my knowledge went undetected for years.) To reduce these dangers it is reasonable to insist that the program should be run twice, on essentially different computers using essentially different compilers. This does not quite meet classical standards; but it is a very modest requirement, and gives rather strong assurance that the program did do what it says it does.

But does the program do what the programmer thinks it does, and how does the mathematical community obtain reasonable assurance of this? The suggestion that the reader can actually check the correctness of a complicated published program is ludicrous; indeed, I doubt if there is anyone alive who is both willing and able to do this with a high degree of reliability for the sort of programs which gave rise to this meeting. (The difficulty is not only with fundamental errors, and indeed these are usually eradicated in the course of writing and checking the program. But slips of the pen, of a kind which also occur in published classical proofs but do little damage there, can short-cut some branches of the program; and the result is apt to be that not all possibilities have been investigated.) Working within a theorem-proving environment, even when this is feasible, does add to the credibility of a program; but for a program used in the proof of an important or unexpected theorem, the mathematical community will probably not feel that the credibility which it adds is enough. Formal program verification is not at present capable of dealing with programs as complicated as those which we are discussing in this meeting, and I am not confident that it ever will be.

N. SHAH (Durham, UK). You have raised an important point, ‘theorems’ if proofs are submitted to journal, e.g. 2nd ODE where there are no closed forms. I would urge the mathematical community to make sure that mathematical software currently used has been proved correct otherwise theorems will be published but because of bugs these theorem are non repeatable.
P. SWINNERTON-DYER. Theory does provide methods for computing provable bounds for the solution of a given ordinary differential equation, but I do not know of any satisfactory implementation of any of these methods as a library subroutine. If one solves an ordinary differential equation by standard numerical methods, it is not hard to build in extra equations whose solutions are error estimates; these will not be provably correct, but in practice usually are correct. (As I implied in my talk, this is an area in which one must take a much more relaxed attitude to provable correctness than in pure mathematics.) In particular, over an interval in which the solution is stable standard subroutines are good enough.

For partial differential equations the situation is much less good. But the limitation here arises from the unsatisfactory state of the theory, and this needs to be improved before one is entitled to start complaining about any shortcomings in the software.

E. B. DAVIES (Department of Mathematics, King's College London, UK). When cold fusion was ‘discovered’ there was immediate public criticism that the effect was entirely implausible. However, a number of laboratories set up experiments to try to duplicate the findings. In experimental science an effect is not believed until it has been confirmed independently. The mathematical community should follow the same procedure with respect to computer assisted proofs by requiring independent programmes to be written.

P. SWINNERTON-DYER. I entirely agree. But there is one flaw in the analogy. Cold fusion was always implausible, and I am sure that most of the laboratories which tried to ‘duplicate’ it were actually trying to refute it. But all the computer assisted proofs which I know of are proofs of results which everyone in the area believed to be true long before any proof was announced; and not a great deal of credit is given for producing the second proof of such a result unless that second proof differs fundamentally from the first one.