Constant trouble

Michael Duff

From
Physics World
October 2001

ISSN: 0953-8585

Letters to the editor can be sent by post to *Physics World*, Dirac House, Temple Back, Bristol BS1 6BE, UK, by e-mail to pwld@iop.org or by fax to +44 (0)117 925 1942. Please include your address and a daytime telephone number. We reserve the right to edit letters.

Constant trouble

You recently reported new astrophysical measurements that suggest that the fine-structure constant, α, is changing as the universe ages (September p5; see also p26 of this issue). The constant is defined as $e^2/\hbar c$, where e is the charge on the electron, $\hbar$ is the Planck constant divided by 2π, and c is the speed of light. If this new result is confirmed, it would indeed be of fundamental significance. However, I do not subscribe to your interpretation that this result might imply a change in the speed of light or the Planck constant (or the charge on the electron for that matter) over the history of the cosmos.

Since they are dimensional, the numerical values of c, $\hbar$ and e depend on the units in which you choose to measure them. The value of c in miles per hour is different to its value in metres per second, for example. In contrast, α is the same dimensionless number in any system of units (roughly 1/137). Thus the value of α does indeed represent a fundamental quantity of nature, whereas the values of c, $\hbar$ and e are merely convenient human constructs of no intrinsic physical significance. It is entirely a matter of convention, therefore, whether you ascribe a change in α to a change in c, $\hbar$ or e, or some combination of all three. Only a change in the dimensionless ratio, α, has any invariant meaning.

In fact, α is but one of 19 pure numbers that appear in the currently accepted Standard Model of particle physics. A major goal of theoretical physics is to find a "theory of everything" that would predict these numbers from first principles. In contrast, the number and values of dimensional quantities, such as c and $\hbar$, are quite arbitrary, differing from one choice of units to another. The more different units you employ, the more dimensional constants you need. So asking whether the speed of light, for example, has changed over cosmic history is simply the wrong question.

Michael Duff
Michigan Center for Theoretical Physics, University of Michigan, US
mduff@umich.edu

FORTRAN is still alive and kicking

Christopher Walker advocates the use of computing languages such as C++ and Java instead of FORTRAN for research projects (September p19). His position is one I have encountered a number of times and one with which I have never been able to agree.

FORTRAN was developed in the 1950s with the needs of scientists and engineers in mind. Many of the tasks we require computer codes for today are very similar to those that FORTRAN was originally used for – and it is now just as capable at carrying out those tasks as it ever was. Certainly, programming languages have moved on and newer programming constructs have been introduced. Some of these advancements have been incorporated into the more recent FORTRAN standards, while others require the programmer to use a language such as C++. However, where such constructs are not required, there is no reason not to use FORTRAN.

As Walker states, FORTRAN is still much liked by scientists. They are familiar with it and unwilling to learn a new language. But if FORTRAN is still perfectly adequate, why do we need the additional complexities of C++?

Walker also says he is annoyed whenever new research projects are started in FORTRAN, when C++ or Java would be more than adequate. I, however, am annoyed when the reverse is the case. It's not that I am opposed to C++ and Java, but if their additional features are not required for the task in hand, then I see no reason to use them in preference to FORTRAN. As far as I am concerned, it is all about using the right tool for the job.

Walker's suggestion that the research councils should let FORTRAN be used only if "good reasons are given" is of the greatest concern to me. It is not for the research councils to dictate how research should be carried out and prescribe which tools should be used. It is up to the researcher to decide. I have experienced such prescriptive behaviour from on high on a number of occasions in my career and it has often had negative consequences.

I agree with Walker that research should not be carried out just for the scientific results that it generates, but also for the quality of training that it gives the student. However, shouldn't we be training our students for research – and not for the software industry? In some cases, it will be appropriate for the student to be taught object-oriented languages, but a good grounding in FORTRAN will be more useful for a research career.

In my ten years in research, I have regularly used FORTRAN for both new and existing projects, and have had to use and modify a great deal of legacy FORTRAN code. I have, however, yet to encounter a situation where either C++ or Java are required. Had I followed Walker's suggestions and gone into research with a background in C++ and Java, I would have been completely unprepared. Of course, if I wanted to go into the software industry then I would have had to retrain myself for that career change, as Walker himself has successfully done.

Simon Richards
QinetiQ, Winfrith, Dorset, UK
sdrichards@QinetiQ.com

Walker's concerns are relevant and heartfelt. His letter can be viewed as yet another contribution to the debate over the welfare of FORTRAN that has run ever since its demise (in favour of Algol) was first predicted in the late 1960s. Having since survived the onslaughts of Pascal and Ada, can it now withstand C++? And should it?

A recent discussion on this topic on the newsgroup comp.lang.fortran reached the following conclusions:
1. FORTRAN is easy to learn and does not need a long initial training period.
2. FORTRAN is far from dead and will remain a major language in industry. It now exists as FORTRAN95, with important facilities for data abstraction, array handling, dynamic memory management, pointers and modular programming. The next FORTRAN standard, which will include full object-oriented features and inter-operability with C, is on target for release in 2004.
3. FORTRAN is well adapted to the way that engineers and scientists think.
4. FORTRAN95 is particularly suitable for scientific and engineering problems that have an important numerical component, especially if complex arithmetic is required.
5. Almost all platforms have at least one FORTRAN95 compiler available.
6. C++ programmers may be easier to find, but they are not necessarily needed, since any scientist or engineer will be able to programme in FORTRAN with very little training.
7. Building an application in a complex language that users do not understand well can be disastrous when those who built the application have left. C++ maintenance is proving to be tricky.
8. Applications built in FORTRAN often execute relatively quickly.
9. FORTRAN95 has features that work well with vector and parallel architectures.

Despite the popularity of C++ for general programming, FORTRAN is still alive and kicking in scientific programming. The US Department of Defense failed to impose Ada by decree. Would similar directives from UK research councils be any more effective or even desirable?

Michael Metcalf
Berlin, Germany
michaelmetcalf@compuserve.com