NAG Fortran Library

Foreword

The following Foreword was contributed by the late Professor Fox and the late Dr Wilkinson to the NAG Fortran Library Manual which was released in 1975.

Those who have organised computing services are well aware of the two main problems which face the users of computing machines in scientific computation. First, considerable experience is needed before the user can transform a given algorithm into a very efficient program, and there are many examples in which relatively small amendments to a few instructions can transform a modest program into one considerably more economical in time and storage space. Second, our user needs knowledge of the principles and techniques of numerical analysis, however efficient he might be at program construction, before he can reasonably guarantee to have an efficient algorithm which is as free as possible from numerical instability and which gives good results in economic time. Both the cost of computation and the ever-present desire for quick results make obligatory at least a partial solution to these two problems.

Many computing laboratories and computing services have made some attempts at solution by constructing libraries of computer programs, but only in the last few years has it been possible to develop really comprehensive schemes based on two or more decades of research into methods and their error analysis by numerical mathematicians, and on the development of a new breed of expert in 'numerical software'. This NAG Fortran Library was in fact initiated by a small mixed university band of numerical analysts and their software counterparts, but has increasingly received encouragement, support and material from many 'extramural' organisations.

The compilers of this library have used, as main criteria for the selection of their programs, the concepts of (i) usefulness, (ii) robustness, (iii) numerical stability, (iv) accuracy and (v) speed. But within these criteria several rather difficult decisions have to be made. First, how many different routines are needed in each particular subject area, such as linear equations, optimization, ordinary differential equations, partial differential equations and so on? What is relevant here is the number of 'parameters' of the particular subject area. With linear equations, for example, the matrix might be 'dense' or have some particular 'sparse' structure, it might be symmetric and, if so, possibly positive definite, it might be too large for the high-speed store of some particular computer, it might be one for which an iterative method is known to converge, or the problem might involve the same matrix but have many different right-hand sides, and so on. Each of these sub-groups may require quite different routines for best efficiency, but within each sub-group there may also be several computing techniques requiring a further selection decision.

A second question which has to be answered is the nature and amount of material to be provided for the 'answer' to problems. If the data of the problem are exact, and if the problem has a unique solution, then it is meaningful to ask for results accurate to a specified number of figures. Whether one can get them easily, say with single-precision arithmetic, will depend on the sensitivity of the answers to small changes in the data. For even the storage of exact numbers cannot usually be performed exactly, so that from the outset our problem differs slightly from the one we hoped to solve. Moreover inevitable computer rounding errors will produce solutions which are the exact solutions of a perturbation of the original problem, the amount of the perturbation depending on the degree of stability of the numerical method. With so-called 'ill-conditioned' problems small perturbations from any of those sources produce large changes in the answers, so that 'exact' or very accurate solutions can be difficult to obtain even if they are meaningful.

But the data may not be known exactly. Some of them may be measured by physical apparatus or involve physical constants known with certainty only to a few figures. In that case the answers are meaningful only to a few figures and perhaps even to no figures, and whether the precision of the answers is larger or smaller than that of the data again depends on the degree of ill-conditioning of the problem. How much of this sort of information should the routines provide?

A third decision is the amount of explanation to be included with the programs. It is clearly desirable to include elements of 'why' something is done as well as 'what' is done, but the desirable amount of such information is rather delicate. If there is too much the expert may be too bored to read all of it and may therefore miss something important, while the amateur may find the discussion rather involved, appearing to him rather like an introductory text in numerical analysis, and again may skip most of it but now on the

grounds of indigestibility. Too little, on the other hand, may detract from the value of the routines by giving the amateur too little guidance in the choice which he also always has to make.

This NAG Fortran Library deals with these problems about as well as could be expected in the present state of knowledge of numerical analysts, software and library compilers, and the majority of the users. With regard to the number of routines to be provided it usually gives just the best available within each sub-group, and selects the particular sub-groups which at present seem to be the most needed and for which good techniques are available.

With regard to sensitivity and accuracy it achieves rather less, but this is a problem so far not well treated even by numerical analysts. Information is provided in a fairly economical way for the solution of linear equations, in which the so-called 'iterative refinement' involving a little double precision arithmetic gives valuable information on the sensitivity and a more accurate answer when this is meaningful. For many other problems the user can only obtain this sort of information by his own efforts, for example by deliberately introducing small perturbations and observing their effects on his solutions. This whole area is one in which one hopes for continual improvements in the library routines when better ways to implement them are discovered.

With regard to annotation, the routines do include a fair but not prohibitive amount of 'why' as well as 'what', and there is no doubt that a mastery of this material will enable the user not only to increase the value he gets from this library but also to improve his performance in the inevitable writing of his own routines for problems not directly treated here.

Two other topics are worth mentioning. First, the routines which appear in this library are the result of years of detailed study by numerical analysts and software experts, and it is dangerous in varying degrees to tamper with them and to try to modify them for 'local needs'. In the solution of linear equations, for example, one could without great peril omit the iterative refinement and still get useful results. One loses here just the extra but often extremely valuable knowledge about the 'condition' of the problem which iterative refinement gives comparatively economically. A far greater danger would arise from an attempt to 'speed-up' the routine by, for example, omitting the row interchanges. which are essentially unnecessary with exact arithmetic. Computer arithmetic is not exact, and this fact could cause complete rubbish in the solutions obtained by neglecting interchanges, which in this context ruins the stability of the numerical method.

Second, the library cannot help the user in the proper formulation of his problem. Given, for example, the problem of computing

$$I_r = e^{-1} \int_0^1 e^x x^r dx, \quad \text{for } r = 0, 1, 2, \ldots, 20$$

the library will have routines for evaluating this integral by numerical quadrature, to whatever accuracy is required, for each value of r. But nothing in the library can tell the user that a very much faster method would use the recurrence relation (in the 'backwards direction')

$$I_{r-1} = \frac{1 - I_r}{r}, \qquad \text{with } I_N = 0,$$

where N (> 20) depends on the accuracy required but is determinable by simple and very rapid numerical experiment (and even, in this simple case, by elementary analysis). Nor could the library tell him that the perhaps more obvious use of the forward recurrence

$$I_r = 1 - rI_{r-1}, \qquad \text{with } I_0 = 1 - e^{-1},$$

would fail to produce accurate results beyond the first few values of r with only single-precision arithmetic: that this formulation, in fact, gives a very ill-conditioned problem.

In summary, then, this NAG Fortran Library represents a timely and very important aid to the computer user in scientific computation. Here, and in future extensions, it provides the best available routines for a wide variety of numerical subject areas, backed by a non-prohibitive amount of sensible explanation of both what is being done and why it is being done. But the user must realise that the library can provide no more than it claims in its annotation, that it cannot except where explicitly stated determine for him the degree of ill-conditioning of his problem, nor help him in general to cast his problem into a better form. For such information he should study some numerical analysis or ask the advice of a colleague reasonably experienced in this field. It may happen that in future editions of the library it will be possible to give

more assistance of this kind to the general user, and it is our hope, in welcoming warmly this edition, that future productions will have some useful expansions of this kind, in addition to the obvious need for new routines in the subject areas which in this first venture are not touched upon or treated only sparsely. The research involved will be both exciting and fruitful!

Professor L Fox (Oxford University)
Dr J H Wilkinson, FRS (National Physical Laboratory, England)