

8. REFINEMENT OF STRUCTURAL PARAMETERS

ordinary least-squares problems. For a discussion of some of the statistical properties of the resulting estimates, including a procedure for the computation of the variance-covariance matrix, see Boggs & Rogers (1990).

8.1.7. Software for least-squares calculations

Giving even general recommendations on software is a difficult task for several reasons. Clearly, the selection of methods discussed in earlier sections contains implicitly some recommendations for approaches. Among the reasons for avoiding specifics are the following:

- (1) Assessing differences in performance among various codes requires a detailed knowledge of the criteria the developer of a particular code used in creating it. A program written to emphasize speed on a certain class of problems on a certain machine is impossible to compare directly with a program written to be very reliable on a wide class of problems and portable over a wide range of machines. Other measures, including ease of maintenance and modification and ease of use, and other design criteria, such as interactive *versus* batch, stand alone *versus* user-callable, automatic computation of related statistics *versus* no statistics, and so forth, make the selection of software analogous to the selection of a car.
- (2) Choosing software requires detailed knowledge of the needs of the user and the resources available to the user. Considerations such as problem size, machine size, machine architecture and financial resources all enter into the decision of which software to obtain.
- (3) A software recommendation made on the basis of today's knowledge ignores the fact that algorithms continue to be invented, and old algorithms continue to be rethought in the light of new developments and new machine architectures. For example, when vector processors first appeared, algorithms for sparse-matrix calculations were very poor at exploiting this capability, and it was thought that these

new machines were simply not appropriate for such calculations. Now, however, recent methods for sparse matrices have achieved a high degree of vectorization. For another example, early programs for crystallographic, full-matrix, least-squares refinement spent a large fraction of the time building the normal-equations matrix. The matrix was then inverted using a procedure called Gaussian elimination, which does not exploit the fact that the matrix is positive definite. Some programs were later converted to use Cholesky decomposition, which is at least twice as fast, but many were not because the inversion process took a small fraction of the total time. Linear algebra, however, is readily adaptable to vector and parallel machines, and procedures such as QR factorization are extremely fast, while the calculation of structure factors, with its repeated evaluations of trigonometric functions, becomes the time-controlling step.

The general recommendation is to analyse carefully the needs and resources in terms of these considerations, and to seek expert assistance whenever possible. *As much as possible, avoid the temptation to write your own codes.* Despite the fact that the quality of existing software is far from uniformly high, the benefits of utilizing high-quality software generally far outweigh the costs of finding, obtaining, and installing it.

Sources of information on software have improved significantly in the past several years. Nevertheless, the task of identifying software in terms of problems that can be solved; organizing, maintaining and updating such a list; and informing the user community still remains formidable.

A current, problem-oriented system that includes both a problem classification scheme and a network tool for obtaining documentation and source code (for software in the public domain) is the *Guide to Available Mathematical Software* (GAMS). This system is maintained by the National Institute of Standards and Technology (NIST) and is continually being updated as new material is received. It gives references to software in several software repositories; the URL is <http://math.nist.gov/gams>.