3.3. MOLECULAR MODELLING AND GRAPHICS

3.3.2. Molecular modelling, problems and approaches

This section is concerned with software techniques which permit a set of atomic coordinates for a molecule to be generated *ab initio*, or to be modified, by reference to some chosen criterion, usually the electron density. Software that can change the shape of a molecule must be cognizant of the connectivity of the molecule and the bonding characteristics of atom types. It must also have means of regaining good stereochemistry if current coordinates are poor in this respect, or of performing its manipulations in ways which conserve essential stereochemical features. Approaches to some of these problems are outlined below. Many of the issues involved, including the topics outlined in Section 3.3.1.4 above, have been excellently reviewed by Hermans (1985), though with little reference to graphical aspects, and a comprehensive treatment of modelling methods based on energies is given by Burkert & Allinger (1982).

3.3.2.1. *Connectivity*

It is necessary to distinguish three different kinds of connectivity, namely structural, logical and drawing connectivities. Structural connectivity consists of the specification of the chemical bonding of the molecule and, as such, is an absolute property of the molecule. Logical connectivity consists of the specification of what part or parts of a molecule are moved, and in what way, if some stereochemical feature is altered. Logical and structural connectivity are closely related and in simple cases coincide, but the distinction is apparent, for example, if the puckering of a five-membered ring is being modelled by permitting folding of the pentagon about a line connecting non-adjacent corners. This line is then a logical connection between two moving parts, but it is not a feature of the structural connectivity.

Drawing connectivity consists of a specification of the lines to be drawn to represent the molecule and often coincides with the structural connectivity. However, stylized drawings, such as those showing the α-carbon atoms of a protein, require to be drawn with lines which are features neither of the logical nor of the structural connectivity.

3.3.2.1.1. *Connectivity tables*

The simplest means of storing connectivity information is by means of tables in which, for each atom, a list of indices of other atoms to which it is connected is stored. This approach is quite general; it may serve any type of molecular structure and permits structures to be traversed in a variety of ways. In this form, however, it is extravagant on storage because every connection is stored twice, once at each of the nodes it connects. It may, however, provide the starting material for the algorithm of Section 3.3.1.5.2 and its generality may justify its expense.

From such a list, lists of bonds, bond angles and dihedral angles may readily be derived in which each entry points to two, three or four atoms in the atom list. Lists of these three types form the basis of procedures which adjust the shape of a molecule to reduce its estimated potential energy (Levitt & Lifson, 1969; Levitt, 1974), and of search-and-retrieval techniques (Allen *et al.,* 1979).

Katz & Levinthal (1972) discuss the explicit specification of structural connectivity in terms of a tree structure in which, for each atom, is stored a single pointer to the connected atom nearer to the root, virtual atoms being used to allow ring structures to be treated as trees. An algorithm is also presented which allows such a tree specification to be redetermined if an atom in the tree is newly chosen as the root atom or if the tree itself is modified.

Cohen *et al.* (1981) have developed methods of handling connectivity in complicated fused- and bridged-ring systems.

3.3.2.1.2. *Implied connectivity*

In cases where software is required to deal only with a certain class of molecule, it may be possible to exploit the characteristics of that class to define an ordering for lists of atoms such that connectivity is implied by the ordering of items in the list. Such an ordering may successfully define one of the three types of connectivity defined in Section 3.3.2.1 but it is unlikely to be able to meet the needs of all three simultaneously. It may also be at a disadvantage when required to deal with structures not part of the class for which it is designed. Within these limitations, however, it may be exceedingly efficient. Both proteins and nucleic acids are of a class which permits their logical connectivity to be specified entirely by list ordering, and the software described in Section 3.3.3.2.6 uses no connectivity tables for this purpose. The ordering rules concerned are given by Diamond (1976*b*).

Drawing connectivity needs explicit specification in such a case; this may be done using only one 16-bit integer per atom, which may be stored as part of the atom list without the need of a separate table. This integer consists of two signed bytes which act as relative pointers in the list, positive pointers implying draw-to, negative pointers implying move-to. As each atom is encountered during drawing the right byte is read and utilized, and the two bytes are swapped before proceeding. This allows up to two bonds drawn to an atom and two bonds drawn from it, four in all, with a minimum of storage (Diamond, 1984*a*).

Brandenburg *et al.* (1981) handle drawing connectivity by enlarging the molecular list with duplicate atoms such that each is connected to the next in the list, but moves and draws still need to be distinguished.

Levitt (1971) has developed a syntax for specifying structural connectivity implicitly from a list structure which is very general, though designed with biopolymers in mind, and the work of Katz & Levinthal (1972) includes something similar.

3.3.2.2. *Modelling methods*

Fundamental to the design of any software for molecular modelling are the choices of modelling criteria, and of parameterization. Criteria which may be adopted might include the fitting of electron density, the minimization of an energy estimate or the matching of complementary surfaces between a pair of molecules. Parameterizations which may be adopted include the use of Cartesian coordinates of atoms as independent variables, or of internal coordinates, such as dihedral angles, as independent variables with atomic positions being dependent on these. Systems designed to suit energetic criteria usually use Cartesian coordinates since all aspects of the structure, including bond lengths, must be treated as variables and be allowed to contribute to the energy estimate. Systems designed to fit a model to observed electron density, however, may adequately meet the stereochemical requirements of modelling on either parameterization, and examples of both types appear below.

Inputs to modelling systems vary widely. Systems intended for use mainly with proteins or other polymeric structures usually work with a library of monomers which the software may develop into a polymer. Systems intended for smaller molecules usually develop the molecular structure atom-by-atom rather than a residue at a time, and systems of this kind require a very general form of input. They may accept a list of atom types and coordinates if measurement and display of a known molecule is the objective, or they may accept 'sketch-pad' input in the form of a hand-drawn two-dimensional sketch of the type conventional in chemistry, if the objective is the design of a molecule. Sketch-pad input is a feature of some systems with quantum-mechanical capabilities.

references