Storage and Retrieval of Synthetic Trees[†]

F. CHOPLIN*

Rhone Poulenc Recherches, Centre de Recherches de Saint-Fons, 69190 Saint-Fons, France

S. GOUNDIAM and G. KAUFMANN*

Laboratoire des Modêles Informatiques Appliquês à la Synthêse, ERA 641, Institut Le Bel, 67000 Strasbourg, France

Received February 28, 1982

In order to improve the access to synthesis design programs, a system has been developed to interactively store and retrieve synthetic trees. It includes three subsystems, dealing with synthesis design (PASCOP), structure and substructure search (ARGIA), and storage of synthetic trees (SERAS). A complete description of the ARGIA system is given (screen structure, file organization) along with its present possibilities.

INTRODUCTION

Output from a synthesis design program can be considered not only as a mere set of precursors and reactions but also as a logically ordered collection of structures around a given target molecule. Since several different strategies can be employed to analyze a target, updating of the corresponding synthetic tree represents a learning process from which any user can benefit. Therefore, a complete synthetic tree includes a great amount of valuable data that should be saved for latter use by the chemist. Nowadays, in industrial research, many targets submitted to synthesis design are similar, and in many cases, it can be more efficient to analyze output obtained for related structures. Thus, a system devoted to the storage and retrieval of synthetic trees can be naturally associated to a synthesis design program: its main characteristics should be an easy updating of synthetic trees and a powerful capability for structure and substructure searching. With these ideas in mind and in order to provide a complete facility for using the PASCOP system,[1a-c] we have developed a set of programs which perform storage, retrieval, and updating of synthetic trees. These programs are contained in two main modules: the first one deals with the retrieval of a target via a structure or substructure search; the second one allows storage and updating of synthetic trees. This paper describes the general organization of the system and the conditions in which it is presently run at RPR and Strasbourg University. Its potential use in the development of a reaction retriever is also discussed.

GENERAL ORGANIZATION

Synthesis design is performed by means of the PASCOP system, derived from version 2.0. of SECS.[1d] It has been fully described previously,[1a-c] and here we only give some indications about its main features: (i) it is dedicated to the treatment of organic and of organophosphorus chemistry as well and, in this purpose, accesses a library of ca. 750 transforms, 150 of them describing exclusively organophosphorus reactions; (ii) it possesses powerful strategic capabilities which allow the users either to enter their own goals (interactive strategy) or to use libraries of predefined goals (also named automata) which can automatically build synthetic trees for general classes of targets.[1c] Trees are stored on intermediate files, before being saved permanently.

Figure 1 shows the general organization of the system and the relationships between the three subsystems. Each logical module is defined by a set of routines and specific files: (i) the first one is the PASCOP system, which performs synthesis design; (ii) storage and retrieval of molecular structures are made via the ARGIA module;[2] (iii) manipulation of synthetic trees is achieved by means of the SERAS system. All of these modules can work independently, and the last two are fully described in the following paragraphs.

Table I. Main Characteristics of the Structure and Substructure Search (ARGIA System)

1	graphical input and output
2	fast storage and retrieval of structures
3	substructure search by means of atom and ring screens and atom by atom matching procedures
4	possibility of declaration of fuzzy substructures
5	molecular formula search

STRUCTURE AND SUBSTRUCTURE SEARCH

Today a powerful structure and substructure search capability is a basic requirement in any chemical research center, since the molecular structure provides the only reliable and widely accepted link between otherwise unrelated information.

Important achievements have been made recently in this field,[3] especially due to the increasing use of interactive computer graphics. The ARGIA system has been developed to fulfill this need, and details about its implementation and use are now given. Its main characteristics are summarized in Table I, and a scheme of its general organization is shown by Figure 2.

Structure Search/Input. Any structure or substructure search starts by a graphic input of the drawing. This is made via a menu which allows the user to perform many typical operations: bond drawing, atom-type specification, translation, rotation, duplication of either part of the structure or the entire structure. The chemist can also define his own library of basic fragments, which can be used later for the building of a large structural family. A check of the drawing is made by comparison of the calculated to the provided molecular formula. A structure can include any number of separated fragments or ions, the only limitation being that the total number of atoms or bonds does not exceed 64 (except hydrogen atoms). Actually, this does not represent a serious drawback, since less than 1% of the structures already entered have met this criterion. Work is under way to expand the capability to 128 atoms or bonds, but at the expense of the running speed and response time.

Once the structure has been accepted for input in the file, a biunivocal code is calculated by means of a slightly modified version of the SEMA algorithm.[4] A fast structure search is obtained by application to the code of a hash function, analogous to those described by Wipke et al.[5] In a file of 100 000 structures, comparison of 20-30 codes as a maximum

[†] Presented at the 182nd National Meeting of the American Chemical Society, New York, Aug 1981.

0095-2338/83/1623-0026$01.50/0

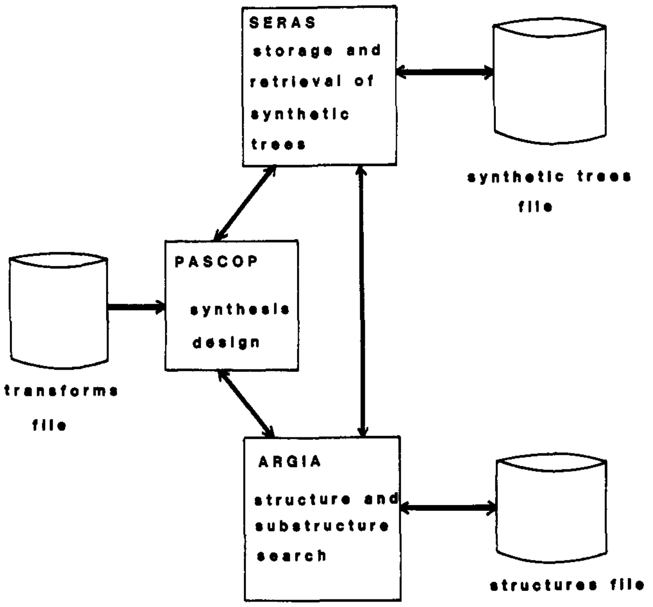


Figure 1. General architecture of the complete system.

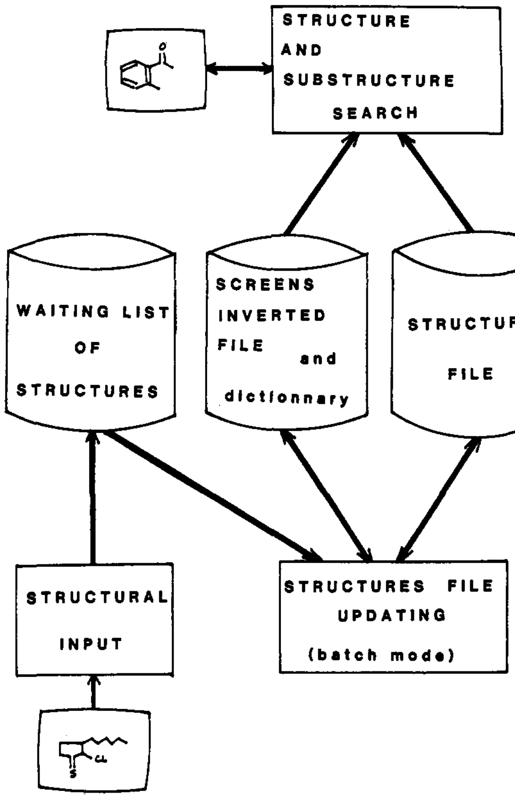


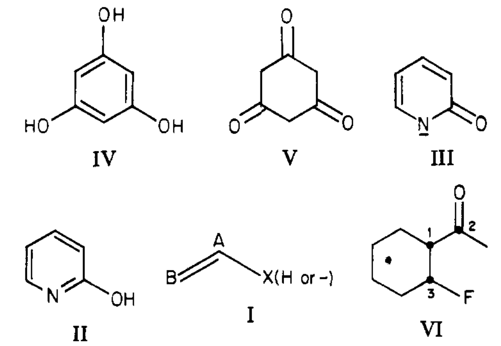
Figure 2. Structure and substructure search system (ARGIA).

is required to retrieve a given molecule. Tetrahedral carbon stereoisomers and *E*/*Z* double bond isomers are differenciated at the code level and can be stored separately, according to the user's demand.

Aromatic and tautomeric bonds are also automatically recognized. Topological aromaticity is detected by application of the Hückel's rule. Standard rules to perceive tautomeric bonds are summarized by structure I in Chart I in which A = C, N, P, or S, B, = N, S, O, or Se, and X = N, S, O, or Se, and X must bear either an hydrogen or a negative charge.

Such rules are deliberately simple and are only intended to avoid multiple entry of similar structures in the file, such as tautomeric forms. Thus, II and III will be recognized as identical and entered only once. Keto–enol tautomerism is not perceived, and structures IV and V will not be found to be identical. These rules are not sufficient to describe reactivity, and more complex rules such as those described recently[6] would be needed in this case.

Chart I



File Updating. Once a drawing has been checked by the chemist, the structure can be entered in the file. Figure 2 shows the main modules and files of the ARGIA system. A molecule is recognized as new after a scanning of the corresponding hash table in the STRUCTURE FILE. Then it is assigned a unique number by the system, and this number as well as the address where the code has been stored are kept to update the other files. Structural masks or screens, which are needed for a substructure search, are automatically perceived, and the corresponding inverted file is updated (for the description of the screen system, see Substructure Search); the user can also enter some other information associated to the new structure: internal numbers, laboratory number, stereochemistry description, etc. No direct interrogation can be performed on this additional information, which is used only for documentation purposes.

After the completion of the drawing, the storage of a new structure takes a few seconds. Usual entry speed varies from 20 to 50 structures per hour (drawing included) for molecules containing 20–30 atoms.

For an increase in this number and for reduction of the cost of interactive updating of the files, new structures are drawn, stored on a WAITING LIST (see Figure 2), and processed at night in the batch mode. This procedure, which might look obsolete, actually retains the most convenient feature of the system (interactive computer graphics) and sharply reduces the costs. Moreover, it does not represent any loss of time for the user, since batch processing is repeated every day.

At RPR, textual information asssociated with the molecules is stored on separate files and manipulated by means of the IBM STAIRS software, which allows handling of very large text files. The unique number assigned to each structure by the ARGIA system provides the link between all the information.

File Organization. The STRUCTURE FILE is organized in a direct access mode: the first eight blocks are used as a directory, where entry *i* corresponds to hash value *i* and contains the address of the first block of table *i* (see Figure 3). Structures which give the same hash value are stored sequentially (with their display coordinates) in a series of linked blocks, each one containing an integer number of SEMA codes. Stereoisomers (*E*/*Z* isomers) are given the same hash value and are therefore stored in the same table. Information providing the address of the first free block in this file and the last structure number allocated by the system are stored in a separate small file.

The space (SP) required to store a structure is given by the following formula:

$$\text{SP (bytes)} = 11 + 2 \times \text{NB} + 2 \times \text{NFRG} + (\text{NB}+ \text{NST} + \text{NEZ})/4 + 3 \times \text{NA}/4 \text{(storage of display coordinates)}$$

(+2 if the last two integer divisions by 4 are not exact)

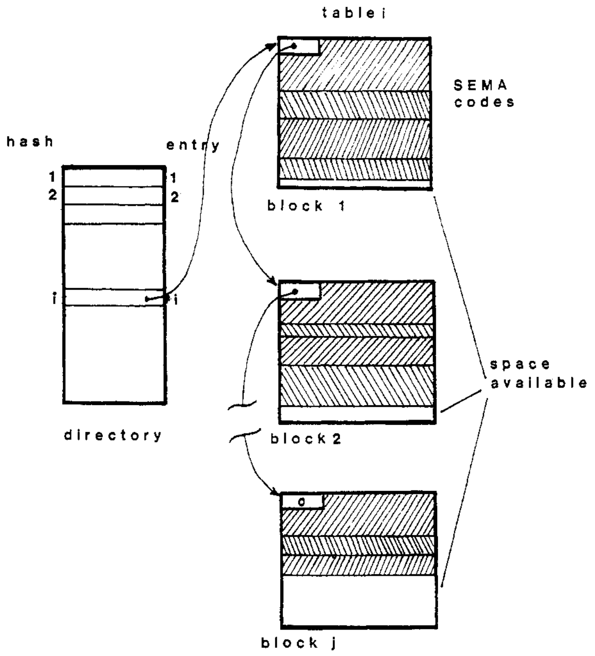


Figure 3. Structure file organization.

with NA = number of atoms, NB = number of bonds, NFRG = number of separated fragments in the molecule, NST = number of Csp3 stereocenters, and NEZ = number of C=C bonds (giving different *E*/*Z* isomers). Thus, for 100 000 molecules containing, on the average, 20 atoms, 17 bonds, and 5 stereocenters (NST + NEZ), one needs at least 6.8 megabytes. Actually, the size of the file should be larger than that, due to the hash code organization.

SUBSTRUCTURE SEARCH

Screen System. For achievement of a fast substructure search, structural masks are perceived by the program for any new molecule entered in the STRUCTURE FILE. Any ring is a screen and also any atom which obeys at least *one* of the following rules: (i) has at least three nonhydrogen attachments; (ii) is connected to a noncarbon atom; (iii) is an atom on a triple bond.

A ring screen is divided into several fields (see Figure 4) and is directly used for addressing into the corresponding dictionary. At most, 3584 such masks can be obtained (due to the fact that only three- to ten-membered rings are represented), but a more limited number is actually in use.

An atom screen contains two parts: the first one is numerical, and the second one is a binary description of the immediate bonds and atoms surrounding the screen center. The first part serves for addressing one of among 256 subtables in the dictionary, each subtable containing the possible binary parts associated to a numerical one.

The bond environment part indicates, by means of a single bit in each case, whether or not the screen atom is attached to a single, double, triple, or aromatic/tautomeric bond. Thus, if it is on a single and a double bond, bits 1 and 2 of this part will be on. In the same way, the atom environment part indicates whether the screen atom is α or not to a heteroatom, carbon, nitrogen, oxygen, sulfur, phosphorus, halogen, or an atom of another type.

Once a screen has been recognized in a new structure, it is compared to those already present in the dictionaries, where a new entry is then added if necessary. Each nonzero entry points to a list in the INVERTED FILE (see Figure 2). Such a list contains the numbers of the structures which include the associated screen. A list is represented by a bit chain, where bit number *i* is on if structure number *i* is in the list. It is physically stored as a series of linked blocks, each one containing 512 32-bit words. Empty blocks are not stored, and there is no limitation to the number of blocks in a list. Although this solution requires more space on disk, we have prefered it to the use of packed bit chains, since packing and unpacking can be rather time consuming. Moreover, all the lists do not expand at he same rate, and many of them, corresponding to rather infrequent screens, contain only one or two blocks.

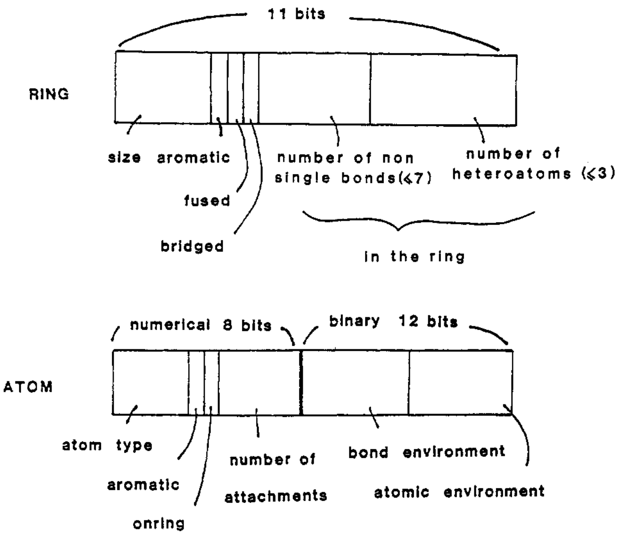


Figure 4. Description of the masks for substructure search.

In order to speed up the updating of the INVERTED FILE, each nonzero entry of the dictionaries contains a pointer to the last block of the list, so that only this one has to be recalled when new structures are added.

The total number of possible screens generated by the system is very large, but only a small fraction of it is really needed. As an example, 860 screens have been recognized in our file which presently contains over 17 000 molecules. This number is expected to increase, but at a much slower rate than previously.

Substructure Search. A substructure, including one or several fragments (and up to 64 atoms or bonds), is entered graphically. However, the drawing represents only part of the user's specifications. Other ones, and/or declaration of fuzziness, can be entered by answering questions which deal with the following topics: (a) maximum number of substituents for each atom (allows the declaration of hydrogen atoms); (b) possible types for unspecified atom; (c) possible types for unspecified bond (this is especially important for tautomeric bonds, which have to be declared by the used, since automatic perception of tautomerism is turned off on substructure search mode); (d) indication of which atoms must be on or off ring (this indication must be consistent with the drawing); (e) indication of which atoms can be aromatic; (f) specification of the rings which must be fused or not; (g) the number of molecular fragments can also be imposed (this option is useful when search is limited to salts).

Each initially perceived mask generates a set of derived screens, all being gathered on an ordered list (SCREEN LIST), the structure of which is given by Figure 5. The screens derived from an initial one are linked by the OR logic and stored on a sublist, whereas the sublists are linked according to an AND logic. An indication of the NOT logic (absence of a fragment required) can be associated to each screen by means of an extra bit, but care must be taken by the chemist when using this possibility at the screen search level. As an example, let us consider substructure VI, in which four screens are initially perceived: the cyclohexane ring and

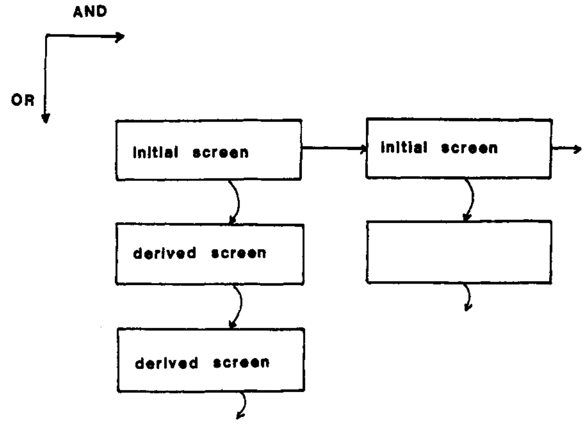


Figure 5. General organization of the SCREEN LIST for substructure search.

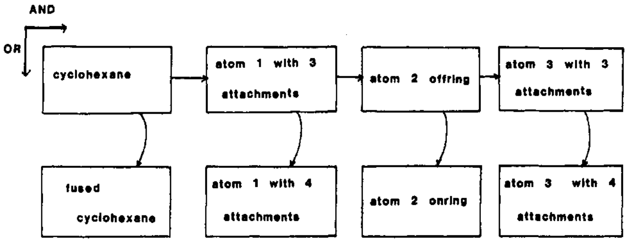

AND
OR
cyclohexane
atom 1 with 3 attachments
atom 2 offring
atom 3 with 3 attachments
fused cyclohexane
atom 1 with 4 attachments
atom 2 onring
atom 3 with 4 attachments

Figure 6. SCREEN LIST obtained for search of structures containing substructure VI.

three atom screen centred on atoms 1–3. If more substituents are allowed for atoms 1 and 3 and if no other specification is provided, new screens will be derived from the initial one and stored on the list described by Figure 6.

The SCREEN LIST is then scanned by the program, each item giving access to an entry in the dictionaries. For a ring screen, access to a nonzero entry corresponds to the presence of an already perceived screen. The situation is different for an atom mask: if the list item corresponds to an atom with the maximum number of attachments (e.g., atom 2 or atom 1 with four attachments in the previous example), there must be a complete match between the binary parts; i.e., one must have the following relationship

$$\text{binary part (SCREEN LIST item)} \equiv \text{binary part (dictionary screen)}$$

On the other hand, for screens with an incomplete number of attachments (such as initial masks for atoms 1 and 3), any dictionary screen which fulfills the following requirement is a valid candidate

$$\text{binary part (SCREEN LIST item)} \subseteq \text{binary part (dictionary screen)}$$

Each entry selected in the dictionaries points to a bit chain which is loaded into main core, and the logic described by the SCREEN LIST is applied to the combination of bit chains. Such a data organization allows fast processing of the screen search and response time, although, depending on the load of the host computer, it does not exceed a few seconds. Due to the structure of the inverted file, this time does not increase as fast as the number of molecules but is expected to grow sixfold when the STRUCTURES FILE increases from 1 to 100 000 structures.

When the screen search is ended, the user is informed as to the number of candidates structures to his query. If necessary, control can be given to an atom by atom matching procedure, based on a technique similar to the set reduction of Sussenguth.[7]

Intermediate and final lists of structure can be saved on a separate file and manipulated independently. They are assigned a number by the system and are used for cross interrogation with the text files.

FILE OF SYNTHETIC TREES

The storage of synthetic trees is an easy operation, but data structures must be adapted to a convenient updating of the file, since such an operation is likely to occur quite often. Each structure in the tree is considered as a target, and only its first level of precursors is stored as a list. Each item in this list contains the ARGIA number of the precursor and the corresponding sequence number of the transform (in the PASCOP system).

A typical interrogation of this file includes a structure or substructure search: each candidate structure is examined, and the list of its sons is visualized, the complete tree being examined in a depth-first fashion. When tree updating is performed, a precursor is considered as new if the complete pair (ARGIA number, transform number) is not in the list of sons, and it is then added to this list. At the present time we handle a file of synthetic trees in organophosphorus chemistry.

Such a data structure can be used to build a reaction retriever, as defined earlier by Gund.[8] Reactions can be registered according to the ARGIA numbers of the reactants and the products which are involved. A file contains all the target–precursor relationships between structures present in the molecular file; thus, any question regarding the existence of reaction to go from A to B, regardless of the number of steps, can be answered quickly and easily. Such a system is now being implemented at RPR and will be described later.

TECHNICAL PART

The present versions of the complete system run on UNIVAC 1110, IBM 3031 and 3033, and PDP 11/60 computers and use TEKTRONIX Models 401X and 4112 as graphic displays. It is written in FORTRAN 4, with a few specific routines in the Assembly language. Storage of a structure in the STRUCTURE FILE requires 100–150 ms (cpu time) on a UNIVAC or IBM 3031 machine (this time does not include the drawing time). The most time-consuming part of the program is the atom by atom matching procedure, for which performances vary from 400–800 structures/min (IBM 3031) to 1500–3000/mn (IMB 3033U, cpu time). As an example, a substructure search with structure II was performed on a file of 17 464 molecules with a version of the program running on an IBM 3033U:1145 candidates were selected after the ring screen search and then 70 by using the atom screens (cpu time needed 200 ms). Atom by atom matching produced 9 structures and required 1500 ms (cpu time). Response times are generally fast but obviously depend on the load of the host computer.

CONCLUSION

Structure and substructure search systems are fast becoming mandatory tools in industrial research and documentation. Although synthesis design is not so widely accepted, its use is broadening, and many chemists are now willing to undertake synthesis planning. Therefore, it appears necessary, not only to store mere structural drawings, but also to save relationships between molecules, and, for instance, those obtained by synthesis design systems. The general programs we have developed provide a mean to perform these operations.

REFERENCES AND NOTES

(1) (a) Choplin, F.; Laurenco, C.; Marc, R.; Kaufmann, G.; Wipke, W. T. *Nouv. J. Chim.* **1978**, *2* (3), 285. (b) Choplin, F.; Bonnet, P.; Zimmer, M. H.; Kaufmann, G. *Ibid.* **1979**, *3* (4), 223. (c) Zimmer, M. H.; Choplin, F.; Bonnet, P.; Kaufmann, G. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 235. (d) Wipke, W.; Braun, H.; Smith, G.; Choplin, F.;

Sieber, W. In "Computer Assisted Organic Synthesis". *ACS Symp. Ser.* **1977**, *No. 61*, 98.
(2) ARGIA stands for Acquisition Restitution Graphique Inter Active.
(3) (a) Howe, W. J.; Hagadone, T. R. "Interactive Graphics-Based Substructure Searching". Presented at the 182nd National Meeting of the American Chemical Society, New York, 1981. Howe, W. J.; Hagadone, T. R. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 9. (b) Hounshell, W. D.; Marson, S.; Peacock, S.; Wipke, W. T. "The MACSS System Search and Retrieval Using an Automated Molecular Access System". Presented at the 182nd National Meeting of the American Chemical Society, New York, 1981. (c) Yoder, D. K.; Walker, T. J. "CAS ONLINE". Presented at the 182nd National Meeting of the American Chemical Society. (d) Attias, R. "The DARC Substructure Search System: A New Approach to Chemical Information". Presented at the 182nd National Meeting of the American Chemical Society, New York, 1981. (e) Feldman, A.; Hodes, L. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 125. (f) Moreau, G. *Nouv. J. Chim.* **1980**, *4* (1), 17. (g) Bremser, W. *Anal. Chim. Acta* **1978**, *103*, 355.
(4) Wipke, W. T.; Dyott, T. M. *J. Am. Chem. Soc.* **1974**, *96*, 4834.
(5) Wipke, W. T.; Krishnan, S.; Ouchi, G. I. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 32.
(6) Roos-Kozel, B. L.; Jorgensen, W. L. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 101.
(7) Sussenguth, E. H. *J. Chem. Doc.* **1965**, *5*, 36.
(8) Gund, P.; Andose, J. D.; Rhodes, J. B. In "Computer Assisted Organic Synthesis". *ACS Symp. Ser.* **1977**, *No. 61*, 179.

MOLDYN: A Generalized Program for the Evaluation of Molecular Dynamics Models Using Nuclear Magnetic Resonance Spin-Relaxation Data

DAVID J. CRAIK,[1a] ANIL KUMAR, and GEORGE C. LEVY*[1b]

National Institutes of Health Resource for Multi-Nuclei NMR and Data Processing, Syracuse University, Syracuse, New York 13210

Received May 17, 1982

A FORTRAN 77 program is described which calculates NMR spectral relaxation parameters on the basis of user-selected models for molecular motion. The program, MOLDYN, allows the comparison of many molecular dynamics models and assists in the testing and evaluation of new models. It is designed primarily for analysis of dynamics in chemical and biological molecules, but the program structure is easily adaptable to other objectives. MOLDYN is used interactively, and extensive prompting and help options are available. Upon selection of a molecular dynamics model, the user is asked to supply or modify parameter values for that model. In one mode the program then calculates T_1, T_2 (or line width), and NOE values by assuming dipolar relaxation of a given nuclear spin pair. However, the inverse problem, i.e., prediction of correlation times or other motional parameters consistent with the NMR observables, is usually of more importance. One of the major aims of MOLDYN is to allow such computations. The program is designed to allow the user to simultaneously vary one or more selected parameters over desired ranges or to automatically optimize modeling parameters on the basis of user-supplied experimental data. Applications range in scope from the analysis of anisotropic motion in rigid molecules to the study of complex conformational and overall molecular dynamics in biological systems.

INTRODUCTION

Nuclear magnetic resonance (NMR) relaxation measurements provide a powerful means of obtaining information related to molecular dynamics.[2,3] With the advent of high-sensitivity NMR spectrometers, an abundance of NMR relaxation data for molecules ranging from simple systems such as substituted benzenes[4] to macromolecules of biological significance such as DNA have been generated.[5] The extraction of dynamics information from these data requires the formulation of mathematical models which express the relationship between expermental observables and theoretical parameters describing molecular motion. The mathematical expressions which form the basis of these models range from fairly simple functions to relatively complicated equations requiring a computer for their convenient solution. Many models are currently available to describe molecular dynamics in a variety of systems, but clear discrimination between different models is often difficult. Thus, a major task involved in the fitting or interpretation of an experimental data set in terms of molecular dynamics is the selection of a model which explains the data adequately and yet is parsimonious (i.e., not overparameterized). This paper describes a computer program, MOLDYN, which allows the comparison of many molecular dynamics models and assists in the testing and implementation of new models. Before description of this program, a short discussion of some features of molecular dynamics models is in order.

THEORY

Several mechanisms can contribute to nuclear spin relaxation.[6] However, here we will be concerned only with the dipolar mechanism in which relaxation of a given nucleus (e.g., ^{13}C) is brought about by dipolar interactions with nearby magnetic (e.g., ^{1}H) nuclei. Molecular motion brings about a modulation of this dipolar interaction and hence provides a fluctuating magnetic field at the observed nucleus which can induce nuclear spin relaxation. The mathematical expression which describes the time dependence of the local fluctuating magnetic field is termed the autocorrelation function, $G(t)$. The Fourier transform of this function, termed the spectral density, $J(\omega)$, describes the frequency components of molecular motion. This latter function enters directly into the calculation of observable NMR relaxation parameters such as spin-lattice relaxation times (T_1's), spin-spin relaxation times (T_2's), and nuclear Overhauser effects (NOE's). Expressions[3] for these parameters are shown in eq 1-3.

$$\frac{1}{T_1} = \frac{2}{15}\frac{\gamma_I^2\gamma_S^2S(S+1)\hbar^2}{r_{IS}^6}[J(\omega_I - \omega_S) + 3J(\omega_I) + 6J(\omega_I + \omega_S)] \tag{1}$$

$$\frac{1}{T_2} = \frac{1}{15}\frac{\gamma_I^2\gamma_S^2S(S+1)\hbar^2}{r_{IS}^6}[J(\omega_I - \omega_S) + 3J(\omega_I) + 6J(\omega_I + \omega_S) + 4J(0) + 6J(\omega_S)] \tag{2}$$

$$\mathrm{NOE} = 1 + \frac{\gamma_S}{\gamma_I}\left[\frac{6J(\omega_I + \omega_S) - J(\omega_I - \omega_S)}{J(\omega_I - \omega_S) + 3J(\omega_I) + 6J(\omega_I + \omega_S)}\right] \tag{3}$$