Current trends in computing: hardware, software and nuclear magnetic resonance research

George C Levy

NIH Resource for Multi-Nuclei NMR and Data Processing, and CASE Center, Syracuse University, Syracuse, NY 13244–1200, USA

In almost no field of technology are current developments as dramatic as in computing hardware and software and their applications in scientific research. This article briefly summarizes the history of computing and its applications in the laboratory, with particular citation of developments supporting nuclear magnetic resonance research, including increased use of graphics techniques. The article also attempts to forecast future trends leading to automated intelligent molecular structure determination from nuclear magnetic resonance and other experiments, coming to utilize, the author believes, extensive use of molecular graphics.

Keywords: computing, hardware, software, nuclear magnetic resonance studies

received 3 February 1986, accepted 17 February 1986

HISTORICAL PERSPECTIVES

History of scientific computing

Scientific computing has undergone revolutionary change since the 1950s and batch processing on IBM 701 or Univac machines. Modern scientific research benefits from various trends in computing technology:

- Supercomputers support diverse computations impractical at earlier times.
- Mainframes, minisupercomputers, and superminicomputers have brought significant computing power to individual laboratories.
- Minicomputers, microcomputers and personal computers have become integrated into experiments and instrumentation, and have also supported standalone data processing applications.
- Various types of computer peripherals have supported scientific research. Array processors and customized processors, advanced graphics devices, and the advent of inexpensive large mass storage devices have all played important roles.

Table 1 summarizes some of the major milestones in computing, with additional commentary on the relationship between cost and ubiquity of computers. The evolution of computing in chemical information processing has recently been reviewed[1].

Relationship of NMR to molecular structure research

There are a number of scientific and medical applications for nuclear magnetic resonance (NMR) spectroscopy and, recently, magnetic resonance imaging (MRI), including: characterization of structure, dynamical studies, quantitation, diagnosis of human health, and monitoring chemical reactions. The largest impact of NMR in chemistry and biophysics has undoubtedly been in elucidation of molecular structure and conformations. In the 1980s, rapidly developing methods of 2D Fourier transform NMR have made possible determination of near X-ray quality structures of small proteins and nucleic acids, *in solution*[2,3]. Up till now, the results of NMR experiments have been fed by hand into computer programs that calculate molecular geometry. Extension to an integrated computing environment awaits only time.

History of computing in NMR

The earliest use of digital computers in NMR was for time-averaging of multiple scans, using hard-wired transient recorders in the mid-1960s and then yielding to hybrid systems performing the first Fourier transforms as early as 1969 in commercial NMR instrumentation. From the early 1970s, all research grade NMR instruments were interfaced to minicomputers which controlled data acquisition as well as performed FFT and standard post FT processing. These systems incorporated analogue oscilloscopes and plotters for I/O. By the mid-1970s, NMR instrumentation was designed around 16–20 bit word minicomputers; low resolution colour raster graphics and digital plotters began to be used. In the early 1980s NMR instrument computers began to be replaced by modern microcomputer and minicomputer systems, augmented by array processors. Furthermore, current NMR instruments incorporate multiple microcomputers in various roles ranging from sample temperature control, to data acquisition, to supervision and control of the user interface. Most current NMR instruments utilize intelligent digital plotters and separate microprocessor subsystems for fast colour raster graphics (low to medium resolution).

Table 1. An outline of the modern history of computers[a]

Date(s)	Development or milestone	Comment
	Mainframes and supercomputers	
1945–6	EDVAC and ENIAC Computers	Moore School
1951	UNIVAC	Eckert/Mauchly
1953	IBM 701	
1960s	IBM 360	
1964	Control Data 6600	
1975	Cray 1	
1980	IBM 3081	
1985	IBM 3090	
	Minicomputers, microcomputers and PCs	
1965	PDP-1	
1971	4004 Microprocessor chip	Intel Corp.
1974	PDP 11/70	
1975–6	Microcomputers	Altair, Radio Shack
1977	VAX 11/780	
1982	IBM PC	8-bit CPU
1984–5	IBM PC — AT	16-bit CPU
1985–6	68020, 32032, MicroVAX-II	32-bit CPU
	Miscellaneous	
1960s–1970s	Core memory	> > $100 000/Mbyte
1970s–1980	Semiconductor memory	
1980	16k bit/chip common	$ 20 000/Mbyte
1982–3	64k bit/chip common	$ 5 000/Mbyte
1985	256k bit/chip common	< $ 1 000/Mbyte
1986–7	1M bit/chip common	< $ 200/Mbyte
1975	10 Mbyte discs common, minicomputers	
1980	100 Mbyte discs common, minicomputers	
1985	1GByte Discs common, minicomputers	
	Laboratory computer operating systems	
1950s	Batch	
1960s	Batch; timesharing	
1970s	Multitasking; timesharing; virtual memory	
1980s	Multiprogramming; networking; virtual machine; Unix; workstation systems	
	Languages	
1956	FORTRAN	
1960	LISP	
1965	BASIC	
1971	PASCAL	
1972	PROLOG	
1976	C	

[a]Prepared largely from data contained in Reference 1.

Several current designs utilize separate processors for data acquisition and data reduction, in the latter case utilizing general purpose microcomputer systems based on the Motorola 68000 series or Digital Equipment's LSI 11/23. There is a trend currently to perform data reduction away from the spectrometer on commercial data stations or on general purpose computers, sometimes forming laboratory computer networks[4].

The development of computer graphics associated with NMR instrumentation and off-line data processing, has not generally kept pace with the level of capabilities used in molecular graphics. This is probably appropriate, since until now NMR graphics have largely consisted of vector draws and alphanumerics, with no solids graphics, 3D transformations, etc. That situation is changing with 2D FT NMR, imaging, and new spectroscopy software systems for determination of structure.

CURRENT TRENDS IN COMPUTER TECHNOLOGY

In many areas of technology, major advances occur once or twice a decade; but in the field of computers, new technology comes to light in 1–2 year cycles. This section will briefly discuss current trends in computer technology, with emphasis on areas that are likely to significantly affect methods of computer graphics.

CPU designs

In 1986, CPU designs at all levels are offering significant new capabilities. Table 2 summarizes some of these developments. Aside from the types of computers represented in Table 2 a number of other systems are now available, specializing in symbolic processing, RISC (Reduced Instruction Set Computers) machines, and computers designed for optimized database management, etc. While some of these systems incorporate powerful graphics, as yet there has not been major emphasis on chemical or biophysical graphics applications. Currently, there is a trend to increase the use of attached processors to speed theoretical computation, signal analysis, and graphics transformations. Modern array processors range from board level devices (1–10 + M floating point calculations/s) to complex standalone systems (6 to > 100 M floating point calculations/s). Array processor designs differ in more than just speed; current designs offer tools to greatly facilitate vectorization of existing programs.

In fact, some of the new minisupercomputer designs *automatically* optimize standard FORTRAN code to take advantage of parallelism *and* pipelining in the CPU.

Workstations

The early and mid-1980s will forever be known for the proliferation of advanced workstation computer systems, characterized by:

- 32-bit CPU power.
- High-resolution high-speed graphics.
- Integrated computer network design with Unix or proprietary multiprogramming operating systems.
- Size and price for the individual scientist or engineer working as part of a team.

Table 3 lists several modern workstation computers, with some special characteristics noted. At least one of these designs has evolved, in fact, *from* a graphics system rather than by adding graphics to a computer design.

The modern workstation computer speeds software development through the use of bit-mapped high-resolution displays that can concurrently operate multiple windows. In fact, on some workstations it is possible in different windows to be working in different operating system environments, or on different computers, for that matter.

Networking

Workstations from Apollo and Sun have brought networking to a level of transparency that greatly improves resource sharing and communications between systems. Even under network environments not having such transparency, modern computer networking offers much to the scientific-user. In the mid-1980s, the Ethernet medium dominated local area networking. While ubiquitous and of relatively high throughput, it is expected that Ethernet will yield by the early 1990s to a broadband network standard having 10–50 times the throughout of Ethernet.

The Ethernet transmission standard is implemented by several network protocols including XNS, TCP/IP, and DECnet. Except for homogeneous DECnet environments, the current Unix standard, TCP/IP, is currently the system of choice, although this situation may change in the next two to three years.

At Syracuse University, a project is underway to generate a new networked scientific computing environment[5]. The new network, LABnet-3 is shown in Figure 1. The main processor is a VAXcluster with a VAX 8600 which is anticipated to be upgraded to a VAX 8650. All of the computers on LABnet-3 can communicate via Ethernet; the DEC computers via DECnet and the other systems via TCP/IP. It is anticipated that a gateway will be designed for the Ultrix micro-VAX-II

Table 2. Examples of current central processing units

Class or machine	Approximate floating point speed with floating point hardware, if available (VAX 11/780 = 1)	Program address space
Personal computers		
IBM PC-XT/370	0.16	4Mbyte
IBM PC W/8087	0.05	640kbyte
APPLE MacIntosh	0.028	16Mbyte
Apple III	0.001	128kbyte
Supermicrocomputers[a]		*Implementations*
Motorola 68020	0.7–3[b]	Sun-3, Apollo, MassComp, Others[c]
MicroVAX-II chip set	0.8–0.9	MicroVAX-II
National Semiconductor 32032; Intel 386; Zilog Z80,000	≈0.5–2 (or more)	
Superminicomputers[a]		
VAX	0.7–6	VAX 8650[d]
Data General MV Systems	0.6–5	MV/20000[d]
Gould (SEL) [not virtual memory design]	3	9050 Power Node
PRIME	1.5	Prime 9950
Minisupercomputers[a]		
Convex	15 (25[e])	Convex C-1
Alliant	10 (35[e])	Alliant FX/4
Culler PSC	≈50	with host Sun-3
Supercomputer[a]		
Cray XMP (64 bit)	200 (1000[e])	

[a]32-bit designs with all supporting virtual memory operating systems. Floating point speeds estimated using Linpack (half-precision, 32 bits) Benchmark (J. Dongarra, Argonne National Laboratory); single processor implementations
[b]Use of coprocessors based on Weitek or AMD devices extends 68020 designs for floating point arithmetic
[c]Older systems have 68010 cpu's that perform at a significantly lower level
[d]Top of line in January 1986
[e]Benchmarked with a full precision (64-bit) implementation that attempts to reflect the true performance of advanced scientific computers. (See Dongarra, J J and Eisenstat, S C *ACS Trans. Math Software* Vol 10 (1984) pp 221–230

Table 3. Graphics workstation computers[a]

System	Comments
Apollo	Aegis/Domain proprietary network operating system; Unix shells; strong engineering base, excellent FORTRAN
Mass Comp	Laboratory specialization, *fast* data acquisition capabilities
Silicon Graphics	Turbo 2400 combines Unix 68020 workstation with Silicon Graphics geometry engine and fast raster graphics
Sun	Sun-3 systems add fast Weitek chips and graphics coprocessors to speed performance significantly. Open VME architecture supports array processors, etc. Network file system
VAXStation	Systems based on MicroVAX-II run under Ultrix (BSD 4.2 Unix) or VMS; DECnet. Graphics options are improving

[a]All support colour or black and white graphics at resolutions near 1K × 1K; 2 to 4 or more Mbyte of random access memory, large disc memories, Ethernet controllers

which will allow communications between these two protocols. The Apollo DN-660 contains an Ethernet interface and software for communication with the VAX systems as well as its own Domain interface for future additional Apollo nodes.

Software

Recent advances in computer software methods, while not as dramatic as the corresponding hardware developments, offer ever increasing help to scientists designing new applications. New and improved language facilities and advanced software development environments (as embodied in the workstations discussed above) presage the anticipated development of automated software engineering. One goal, in fact, of the Japanese 5th Generation Computer Project, is for those computers to have the ability to program themselves, given a description of the problem to be solved.

The most common language in scientific computing remains FORTRAN, the first high level language, originally developed in the early 1950s. The most recent FORTRAN standard, commonly called FORTRAN 77 and now supported by virtually all compilers, has gone far towards improving the language, incorporating many of the features of the newer and highly structured Pascal language. Pascal itself has become very common in scientific computing while Basic and extended Basic languages have kept a share of the action, although generally in smaller program environments. In the laboratory, there are strong advocates for FORTH[6] and in computational chemistry some large systems have used language extensions such as MAINSAIL, RATFOR, etc. Strong interest in Unix operating systems and good control of system-level operations have encouraged developments in the language, C. Apart from this, the nonprocedural languages LISP and PROLOG, used most often in artificial intelligence applications, are finding use in advanced scientific programs.

SOFTWARE FOR NMR AND ITS CHALLENGE TO COMPUTER TECHNOLOGY

In NMR there are two largely separate arenas for computing: (1) data acquisition and experiment control and (2) data reduction. The tasks of acquiring data and controlling the spectrometer or imager are fully covered by computers embedded in the instrumentation, usually through proprietary software that is not available to the user. Event timing in NMR, pulse programming, is on a rapid and sustained timescale (10^7–10^8 Hz, sustained for minutes or hours). On the other hand, the data reduction task has relatively light real-time constraints. Individual users may wish to extend or modify data reduction protocols to accomplish additional goals. A recent trend has been the incorporation of offline (ex-spectrometer) computers for NMR data reduction, in order to facilitate adaptation of new data reduction procedures or to remove lengthy processing from the instrument, and thereby increase spectroscopic throughput.

In the 1960s, FT NMR data reduction typically encompassed performing a magnitude fast Fourier transform (online phase-sensitive calculations were not implemented at that time) and plotting the results. Current NMR data reduction is far faster than was possible even a few years ago, and there have been significant advances in the degree of sophistication of certain algorithms (e.g., phasing, apodizations), but all present instrument data reduction software remains basically interactive, with the spectroscopist making processing decisions based on what he sees at a graphics terminal. Examples of noninteractive algorithms that have been used (although not yet on commercial spectrometers) include extensions of Pearson's base-line flattening,[7] DISPA analysis and automatic phasing using DISPA[8], deconvolution of overlapping lines by automatic curve fitting[9], signal-to-noise and resolution manipulation using the maximum entropy method (MEM)[10], etc.

Sophisticated software for optimal utilization of NMR and other spectroscopic data

NMR and other laboratory experiments invariably produce experimental data subject to random and systematic artifacts: noise, nonideal responses, missed or corrupted data points, etc. User judgments in interactive processing of these data inevitably bias results, often unintentionally.

Processing decisions to evaluate and quantitate spectral features should be based on statistical and other numerical methods, carefully designed to be as neutral as possible to data that is highly precise as well as to data that has been subjected to laboratory nonidealities. Such programs, in fact, can be designed to operate by default in a totally automated mode. Interactive operation can also be supported, but those capabilities should allow the user to select general actions; not specifically decide on differential processing criteria.

An approach to intelligent NMR data processing: the NMR1/NMR2 project

Design considerations can vary for a high-level software system to optimize data reduction in Fourier transform spectroscopy. In our situation, we initially set the goals

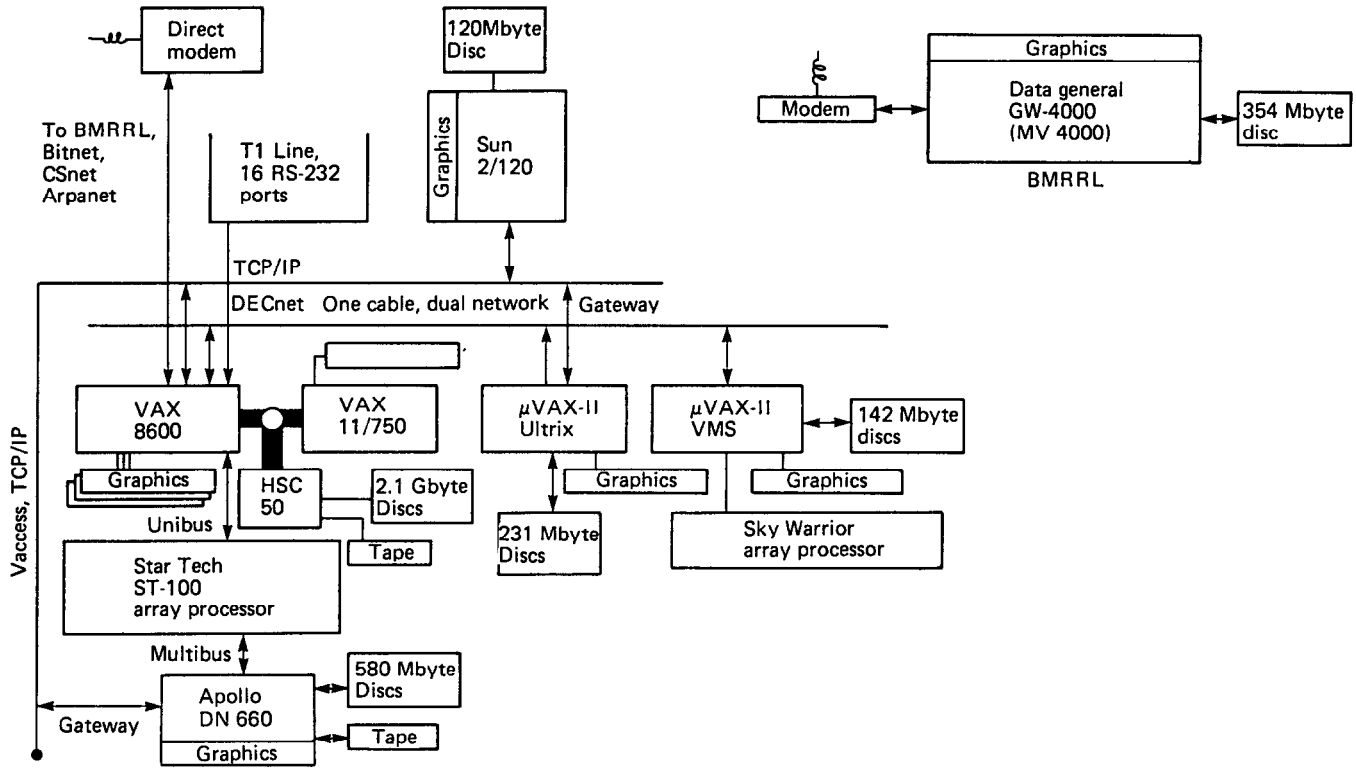

Figure 1. LABnet-3 laboratory computer network in its 1986 configuration; ⊤ *VAXCluster computer interconnect (70 MHz)*

listed in Table 4(a) (from Ref. 4), which also lists design considerations and constraints that resulted from the designation of those goals. Table 4(c) lists goals that were added over the period 1983–1985 to further develop the software and to direct the project toward development of the first family of expert systems for characterization of quantitative information resulting from NMR and other spectroscopic/imaging experiments.

The magnitude of work involved in the NMR1/NMR2 project has been significantly increased by the need to produce a portable code that supports a variety of peripherals as well as central processing unit environments. By the end of 1985, approximately 40 man years had been expended, including code development and maintenance.

A primary motivation for development of these software systems was to develop improved methods for quantitating spectral features in difficult chemical and biomedical NMR spectroscopy. An example of the former application is accurate peak area measurement for spectra having a very large dynamic range (with peaks having areas over a ratio of 10^3–10^4 in a single spectrum)[11]. An example of a biological application is the quantitation of high-energy phosphates in an *in vivo* metabolic NMR spectrum (*in vivo* spectra are usually characterized by low signal-to-noise, short time acquisitions are typical, and poorly defined base lines and lineshapes.

Figure 2 shows an example which uses one of the newest modules of NMR1, namely baseline flattening and deconvolution, designed to identify and correct poorly defined baselines to allow quantitation of spectral peaks. The algorithm used in this case is based on a finite automaton[12]. The user is allowed to parameterize the automaton for different classes of spectra. Further, the user is advised by the software of the resulting baseline identification, using colour coding in the drawing of the spectrum on the graphics terminal. Figure 2(a) shows the uncorrected ^{31}P NMR spectrum of a human brain, taken with depth pulse and topical localization. Figure 2(b) shows the result of baseline conditioning by the finite automaton and Figure 2(c) shows the result of parameterization to force baseline resolution between all peaks. (Results such as those shown in Figure 2(c) can be very valuable but in cases where there is real peak overlap, this type of processing will distort intensities.) Figure 3 describes the result of automated curve fitting, on the baseline flattened spectrum from Figure 2(b); the combination of automated baseline conditioning and peak quantification by Gaussian curve fitting, yields optimum results and the procedure uses just 2–5 minutes of CPU time on a moderate-sized VAX without array processor. Note, however, that in Figure 3(d) automated curve fitting does not work accurately for the inorganic phosphate peak. Presumably, this is a limitation of the choice of Gaussian lineshape for the dataset.

NMR2 is a program environment for processing multidimensional FT NMR spectra or datasets in magnetic resonance imaging. Efficient data processing of large 2D data arrays poses practical computational and display problems in today's laboratories, particularly when such processing is to include operations such as peak analysis, quantitation, and deconvolution or maximum entropy method computation. Further, the data processing requirements of 2D FT NMR pale next to the needs anticipated for future clinical studies where 3D imaging and 4D volume chemical shift imaging will require significant advances in computational speed, data storage size, and effective bandwidth to the CPU. It is, of course, also conceivable to add a time dimension to 2D NMR spectral data or 2D to 4D imaging data. In the case of a kinetics experiment, this would not require Fourier transforms across an additional dimension

Table 4. Design goals and design considerations for the NMR1/NMR2 data reduction software systems (see Ref. 4)

(a) Early goals
(1) Complete separation of the data acquisition and data reduction processes
(2) Allow multiple users simultaneous access on one computer
(3) Provide as much hardware independence as possible
(4) Design a software system that does not require extensive user interaction in processing data
(5) Use of algorithms for spectral conditioning and analysis that utilize statistically-based decision, avoiding unintentional introduction of operator bias
(6) Design of a program architecture that is easily extended and adapted for new capabilities
(7) A design that is highly flexible for expert users and highly prompting for novice users; extensive user help

(b) Design considerations and constraints
(1) 32-bit computers with large address space and multiuser operating system
(2) All code is written in FORTRAN 77 (largely structured)
(3) Program design is top-down and modular
(4) The programs provide support for 15 graphics devices (adding more all the time!), but this necessitates compromises in usage of graphics features, such as cursors and input devices (restricted to keyboards in most implementations of NMR1/NMR2)
(5) The sheer size of the programs mandate a virtual memory operating system
(6) Floating point implementation for highest dynamic range and for statistical treatments requires efficient floating point execution, eliminating use of some hardware

(c) Added goals (1983–1985)
(1) Extend statistical and numerical methods to independently evaluate intermediate results
(2) Automated tracking of spectral features with full statistical analysis of changes in peak intensity, positions, etc. (relaxation analysis, kinetics, etc.)
(3) Develop online parser/interpreter to insert user-selected equations, and relationships (e.g., for evaluation of complex kinetics)
(4) Design versatile program command language with ability to develop complex, totally automated analysis protocols that support conditional branching, etc.
(5) Develop custom analysis modules for automated polymer analysis (tacticity, degree of polymerization, etc.) and other specific applications; include user-specific database where appropriate
(6) Extend individual user customization capabilities, adding parameters, flags, and user-interface characteristics to personality profiles that are kept between sessions
(7) Provide initial software interfaces for future development of postprocessor Expert Systems
(8) Develop initial logic programming applications (e.g., 2D FT NMR peak picker, including logical differential of 'real' peaks from experimental artifacts and noise)
(9) Install as modules user-friendly versions of existing spectroscopic software packages (e.g., LAOCOON5, DAVINS, DNMR5)
(10) Develop maximum entropy method (MEM) algorithms for simultaneous signal-to-noise and resolution manipulation, phasing, automatic spectral conditioning, *ab initio* deconvolutions, etc.
(11) Develop and optimize capability for 3D and 4D (chemical shift-volume imaging) NMR
(12) Develop a workstation environment to optimize NMR1/NMR2 operation [obviating (b4) for example]

but other operations might require access to the full dataset. Clearly, new, much higher speed online memories will have to be used, along with data compression schemes, where possible.

FUTURE TRENDS

The sections above have already implied some of the anticipated developments in computing for NMR and related experimentation. Hardware improvements will enable scientists to extend possibilities in NMR data reduction and, perhaps more importantly, optimally couple NMR data reduction to other determinants of molecular structure. Table 5 describes the author's prejudiced view of a future colour graphics workstation computer which could be used for this work. The section below expands on the author's concept of developing automated intelligent approaches to the characterization of solution structures.

APPROACHES TO EXPERT SYSTEMS FOR CHARACTERIZATION OF MOLECULAR STRUCTURE FROM NMR SPECTROSCOPY USED IN CONJUNCTION WITH OTHER EXPERIMENTS AND COMPUTATION

Current NMR methodology supports the determination of near X-ray quality structures for small proteins. A combination of traditional and especially 2D NMR methods yields excellent information on tertiary structure, largely through determination of proton–proton distances on nonadjacent residues (but sometimes including other NMR information: ^{13}C connectivities, metal ion–nuclei distances, dihedral angles through observed spin–spin coupling, etc.).

Up till now, the use of molecular graphics in bio-

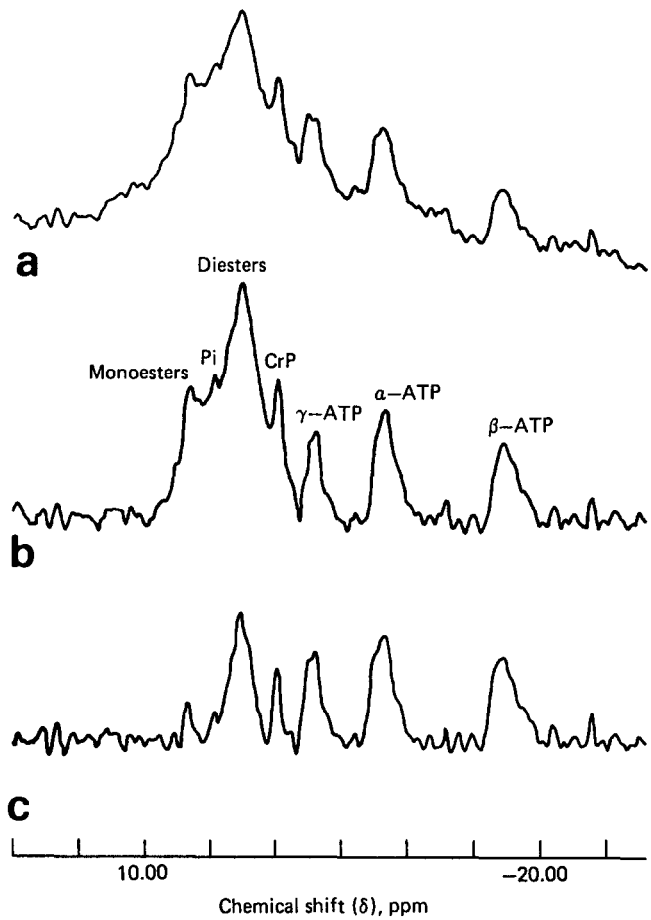

Figure 2. ^{31}P in vivo spectrum of the human brain using depth-pulses and surface coil technology (spectrum courtesy of P Styles and G Radda, Oxford University); (a) spectrum after Fourier transform, (b) spectrum after baseline conditioning to flatten baseline hump, (c) spectrum after baseline conditioning to force baseline resolution between each peak

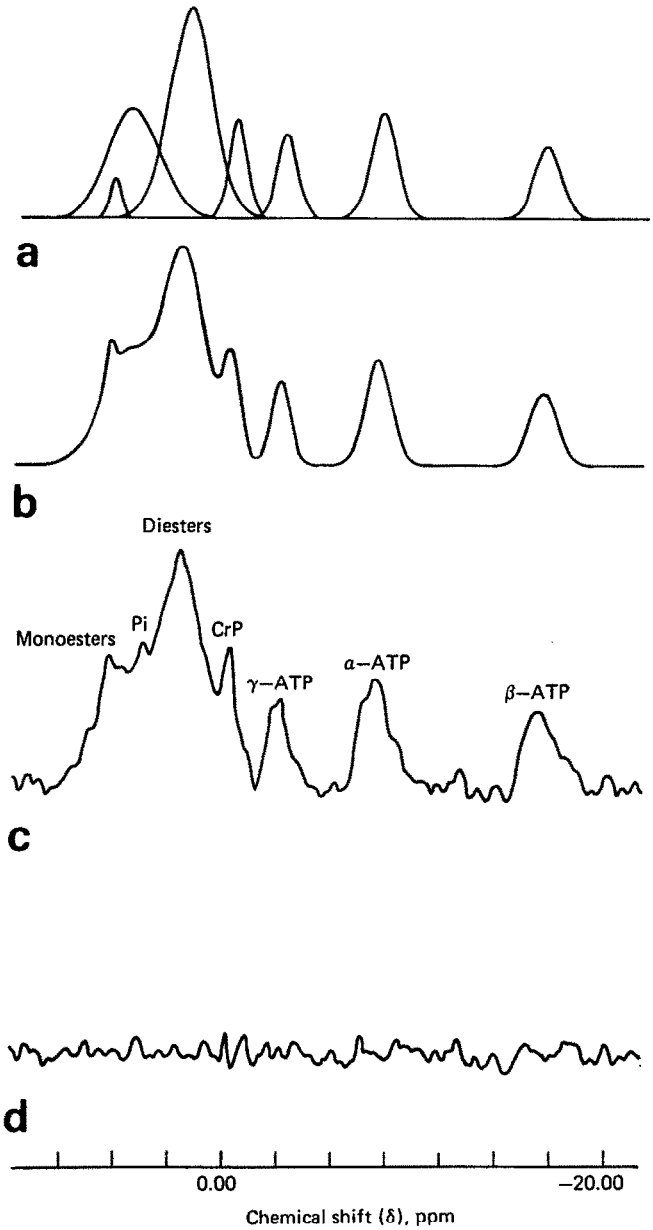

Figure 3. Result of automated curve-fitting using Gaussian lineshapes of the data from Figure 2(b), reproduced as c in this figure (expanded); (a) individual components calculated for spectrum, (b) calculated curve, (c) experimental data, (d) difference spectrum between B and C

polymer studies has been coupled with molecular mechanics and dynamics calculations; they have also in a few cases been coupled directly with information from 2D NMR experiments. Each of these potential sources of conformational/structural information has unique strengths and frustrating limitations. Thus far, there has been no effective scheme to optimally determine protein or nucleic acid structure, utilizing *and weighting* the various evidence from experiments and calculations. Such a scheme must inevitably have the expertise to sort through qualitatively similar but quantitatively different evidence, evaluating the worth of each experimental or calculated datum. An approach to this goal which we favour combines the use of multiple, confirmatory experiments with an advanced computer program combining statistical and other numerical analyses with logical analyses. Such an expert system[13] structure should have the ability to calculate geometrical details for very complex molecules/complexes *and assess the degree of accuracy* of the determined structures. Figure 4 shows such an expert system. No such expert system mechanism, as in Figure 4, exists as yet, but a number of groups are seeking the goal of solving biopolymer structures from traditional computational perspectives. The Stalog project at Syracuse University[14] aims to produce the expert systems approach of Figure 4, although the initial application is not for biopolymer structure determination.

It is obvious that any expert systems approach to molecular structure determination will greatly benefit from advanced molecular graphics methods. The use

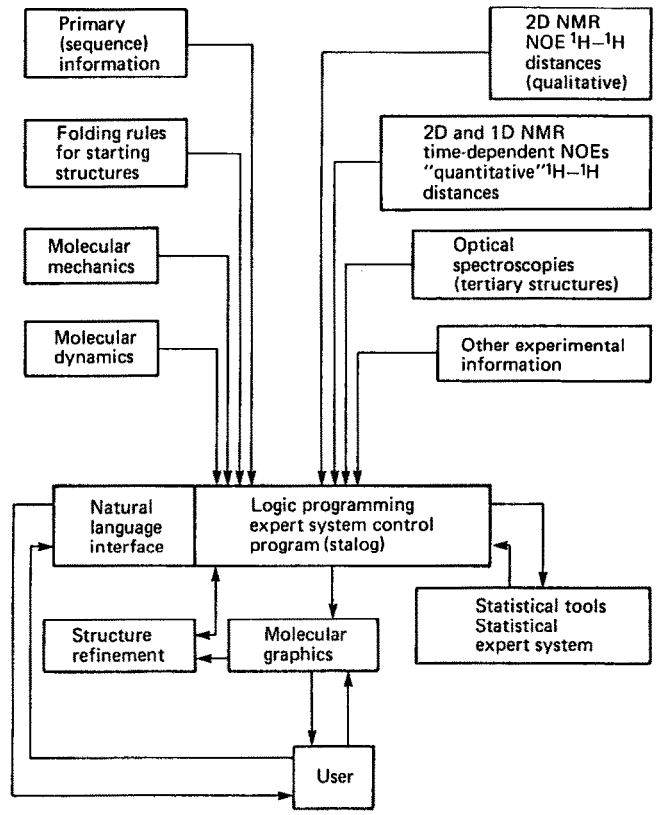

Figure 4. Future expert system for determination of solution structures of biopolymers

Table 5. A personal view of the 1990s workstation computer

Subsystem	Description
CPU	32 bit or 64 bit hybrid; 15 to 40 [VAX 11/780] units in speed; Bus bandwidth > 100 Mbyte/s
Memory	16–64 Mbyte random access memory > 1 Gbyte disc with > 10 Mbyte/s throughout
Graphics	1500 × 1200 colour graphics; 10^5 2D or 3D transformations/s
Peripherals	>> 100 Gbyte laser discs and online knowledge
Networking	Transparent, including international networks; online supercomputers (Class 8)

of such a system will receive an additional bonus: the natural language interface and expert system control program will significantly improve the user interface and subsequent control of graphical output.

ACKNOWLEDGEMENTS

The author appreciates the hospitality extended by Oxford University, where this article was prepared. Financial support from the Division of Research Resources (Grant RR-01317) and the New York state-funded Advanced Technology Center for Computer Applications and Software Engineering is gratefully acknowledged.

REFERENCES

1 **Rush, J E** *J. Chem. Inf. Comp. Sci.* Vol 25 (1985) pp 140–149
2 **Billeter, M et al.** *J. Mol. Graph.* Vol 3 (1985) pp 79–83
3 **Kaptein, R et al.** unpublished programs
4 **Levy, G C and Begemann, J H** *J. Chem. Inf. Comp. Sci.* Vol 25 (1985) pp 350–357
5 **Levy, G C** *Chem. & Comput.* submitted
6 **Dessy, R E** in **Levy, G C and Terpstra, D (eds)** *'Computer networks in the laboratory'* Wiley, USA (1981) Chap. 3
7 **Dumoulin, C L and Levy, G C** *J. Mol. Struct.* Vol 113 (1984) pp 299–310; **Dumoulin, C L and Levy, G C** *Comput. Chem.* Vol 5 (1981) pp 9–18
8 **Pearson, G A** *J. Magn. Reson.* Vol 27 (1977) pp 265–272
9 **Kumar, A et al.** *Comput. Enhanced Spectrosc.* Vol 1 (1983) pp 107–114
10 **Sibisi, S et al.** *Nature (London)* Vol 311 (1984) pp 446–447; **Sibisi, S** *Nature (London)* Vol 301 (1983) pp 134–136; **Ni, F et al.** *J. Magn. Resonance* Vol 3 (1986) pp 79–85
11 **Sotak, C H et al.** *Analyt. Chem.* Vol 55 (1983) pp 782–787
12 **Jozefson, M et al.** to be submitted
13 **Waterman, D A** *'A guide to expert systems'* Addison-Wesley, USA (1986)
14 **Levy, G C et al.** to be submitted