MacroModel—An Integrated Software System for Modeling Organic and Bioorganic Molecules Using Molecular Mechanics

Fariborz Mohamadi,[a] Nigel G. J. Richards,[b] Wayne C. Guida,[c] Rob Liskamp,[d] Mark Lipton, Craig Caufield,[e] George Chang,[f] Thomas Hendrickson,[g] and W. Clark Still[h]

Department of Chemistry, Havemeyer Hall, Columbia University, New York, NY 10027

Received 15 March 1989; accepted 24 August 1989

An integrated molecular modeling system for designing and studying organic and bioorganic molecules and their molecular complexes using molecular mechanics is described. The graphically controlled, atom-based system allows the construction, display and manipulation of molecules and complexes having as many as 10,000 atoms and provides interactive, state-of-the-art molecular mechanics on any subset of up to 1,000 atoms. The system semiautomates the graphical construction and analysis of complex structures ranging from polycyclic organic molecules to biopolymers to mixed molecular complexes. We have placed emphasis on providing effective searches of conformational space by a number of different methods and on highly optimized molecular mechanics energy calculations using widely used force fields which are supplied as external files. Little experience is required to operate the system effectively and even novices can use it to carry out sophisticated modeling operations. The software has been designed to run on Digital Equipment Corporation VAX computers interfaced to a variety of graphics devices ranging from inexpensive monochrome terminals to the sophisticated graphics displays of the Evans & Sutherland PS300 series.

INTRODUCTION

With recent advances in the techniques used to construct and analyze chemically complex molecules, the computational modeling of molecules has become an important tool for designing new compounds with novel properties. Developments in computer graphics have aided this process by combining visualization of structures in three-dimensional space with display and coloring techniques that represent nonstructural features and properties. Empirical force field methods (molecular mechanics) have developed to a level where accurate geometries can be obtained for most organic structures including those of biological interest. These calculations also provide good estimates of the relative stabilities of specific conformational states in the gas phase for both small organic and complex biological molecules, and substantial progress is being made toward computing the free energies of molecules in solution. In recent years, many efforts in computational chemistry have focused on the integration of molecular mechanics with graphics. Such graphics interfaces facilitate the rapid assembly and qualitative study of molecular structures; however, we feel that the primary utility of graphics in molecular modeling is to facilitate control of the energy-based analyses of structure and properties. We provide here an overview and a description of novel features of a graphics-driven molecular modeling system developed at Columbia University as an aid for our experimental work in molecular design and recognition. The system is named MacroModel and it provides computational techniques for structure assembly, conformational searching, molecular mechanics calculations and molecular dynamics simulations for application to problems in organic and bioorganic chemistry.

Owing to the large number of different molecular modeling operations that can be carried out with MacroModel, we will limit this discussion to an overview of the main functions of the program and concern ourselves primarily with its novel features. Like any tool which reflects

[a] Present address: Lilly Research Laboratories, Indianapolis, Indiana.

[b] Present address: Department of Chemistry, The University, Southampton, U.K.

[c] Present address: Pharmaceuticals Division, CIBA-GEIGY Corp., Summit, New Jersey.

[d] Present address: Department of Chemistry, Gorleaus Laboratories, University of Leiden, Leiden, The Netherlands.

[e] Present address: Wyeth-Ayerst Research, Inc., Princeton, New Jersey.

[f] Present address: Medicinal Chemistry Division, Pfizer, Groton, Connecticut.

[g] Present address: Department of Chemistry, Drexel University, Philadelphia, Pennsylvania.

[h] To whom all correspondence should be addressed.

Journal of Computational Chemistry, Vol. 11, No. 4, 440–467 (1990)

CCC 0192-8651/90/040440-28$04.00

current scientific thought, MacroModel is under continuous revision and development. The initial version of the program was completed in February 1986 and distributed to several dozen laboratories outside Columbia. Since then, there have been three further publicly available versions of MacroModel. Version 1.1 was distributed in August 1986 and Version 1.5 in April 1987. The discussion that follows refers specifically to the recent Version 2.0 release, although many of the features described were also part of earlier versions.

Given the ever-increasing number of graphics packages available for molecular modeling, some comments on the historical development of MacroModel seem appropriate. Several years ago, research programs on the fundamental processes of molecular recognition in both organic and biological systems were initiated in our laboratory with work aimed specifically at the design of binding agents (hosts) for certain peptidic and functionalized organic guests. A method for prediction of the binding properties of potential ligands was deemed essential, given the lengthy effort required to prepare complex molecules by organic synthesis. In particular, we needed computational methods that would allow us to rapidly assemble new molecular structures and provide the relative energies of populated, low energy conformations of both hosts and complexes. Physical molecular models were inadequate for such modeling tasks because of the complexity of the potential energy surface of polar molecules. It was also evident that the preorganization requirement of effective hosts would require us to carry out effective conformational searches of flexible molecules. Finally, realistic evaluations of configurational entropy and solvation free energies would be necessary for modeling quantitatively complexation phenomena.

In contrast to modeling with physical models, computational methods seemed to offer plausible ways to not only search conformational space and establish the relative stabilities of the various conformers found but also to predict ultimately the binding properties of new molecules in solution prior to the commitment to synthesis. Given the size of the molecules to be analyzed and the number of structures which would have to be processed during conformational searching, we chose the fastest quantitative energy calculation available, i.e., molecular mechanics. We knew that we would need to develop a variety of computational techniques in order to construct relatively large, complex molecules including biopolymers, to study their structures and energetics with a number of different molecular mechanics force fields, to parameterize new molecular substructures, to search out their low energy conformations and so forth. Furthermore, if the computations were to be used effectively for designing molecules, they would have to be interactive and fast and, for the more computationally intensive parts of a problem, methods would have to be devised that would initiate effortlessly noninteractive computations. For subjective endeavors like the design of new molecules where ideas and strategies continuously evolve based on intermediate results, we viewed a speedy and supple dialogue between the user and the computational modeling system as essential.

In previous work involving modeling macrocyclic compounds for synthetic purposes, one of us (WCS) made use of graphically controlled molecular mechanics and automated conformational searching. For that work, we developed a modeling program called MODEL which integrated molecular mechanics subroutines from MM2[1] and interactive graphics from our reaction-searching system SynLib. While MODEL had been used extensively for studying small organics, it was not really appropriate for our new problems in molecular design since it was limited to only one force field, because it could not be modified easily to carry out energy calculations on large, biopolymer-like structures and their complexes, and because its rudimentary Tektronix 4010-style graphics were not appropriate for displaying and manipulating structures having large numbers of atoms. It was also clear that simple molecular mechanics alone would not provide realistic predictions of binding phenomena and that problems of free energy, solvation, conformational averaging and multiple minima would have to be addressed if useful modeling results were to be obtained. Thus, we began development of the molecular modeling system described here.

METHODS AND RESULTS

Overall Organization

We designed MacroModel for users with little experience in scientific computing. Users interact with the program via drawings of molecular structures and via graphically activated buttons. A series of permanent menus comprise the graphics interface from which the user selects options by "picking" items (buttons, atoms, or bonds) with a standard graphics input device such as a mouse, lightpen, or data tablet. The screen is divided into a number of viewports that are updated selectively during program operation. These viewports include an area for structure display, an area for textual dialogue between the user and the program, and the menu buttons themselves. The screen organization is illustrated in Figure 1, which shows the INPUT screen template used for the construction and manipulation of small organic molecules. Each word along the right and lower edges of the template is a button that controls program operation. The buttons are grouped by their function. In Figure 1, the button groups marked A, B, and C are used to select molecular structures, to control the view and orientation of the molecule in the structure viewport and to switch between the main program modes. These buttons are permanently displayed. The buttons in the remaining groups change depending upon the program options in effect.

The program itself is divided into three main modes, INPUT, ENERGY, and ANALYZ and each of these modes is divided further into nine submodes. Button groups C and D (Fig. 1) control access to the modes and submodes respectively. There is also a supermode called USER which allows the user to program his own special functions and add them to MacroModel. As described in a later section, we have programmed one of the USER submodes to construct a graphical interface to the semiempirical molecular orbital program AMPAC.[2]

INPUT mode is used for constructing new molecules, modifying preexisting ones and changing three-dimensional structure. The available submodes reflect the different types of molecules that can be constructed in a semiautomated way and include small organics, peptides, nucleotides, and carbohydrates. In its handling of biopolymers, MacroModel allows but does not require the distinction of particular atom sets as residues but processes biopolymers as atom-based entities just as it does small molecules. INPUT mode also includes submodes for controlling molecular geometries. These submodes provide capability for carrying out conformational searches and substructure translation/rotations, including docking of the components of multimolecular systems.

ENERGY mode is the heart of the program. It has submodes for energy minimizations, force field modifications, geometrical constraint introduction, molecular dynamics and gas phase free energy calculations. All of these calculations are done interactively with MacroModel although many of the more time consuming computations are done with a related but noninteractive program called BATCHMIN. Over the years, we have investigated many methods for speeding

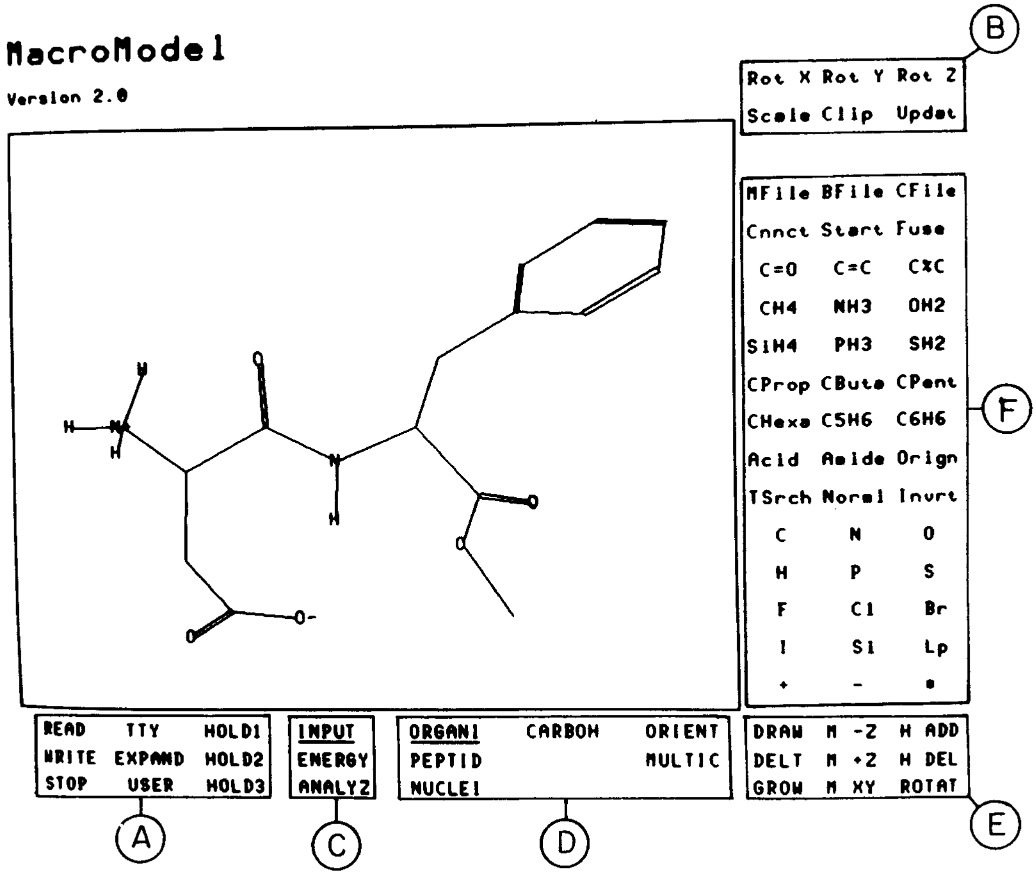


Figure 1. Screen organization showing buttons (menu items) along the right and lower edge of the screen. The program is shown in INPUT mode, ORGANI submode. Current structure viewport shows a structure of Aspartame.

and generalizing molecular mechanics calculations and we will describe some of our more useful findings in a later section. All features of ENERGY mode are available with any of the available force fields (Allinger's MM2,[3] Kollman's AMBER,[4] and Jorgensen's OPLS[5]). Unlike many other molecular modeling systems, MacroModel is not an interface to other energetic programs but incorporates self-contained molecular mechanics routines that select the appropriate energy equations and parameters of established force fields by reading external text files.

ANALYZ mode is the least developed of the three main modes but provides a variety of simple display options including substructural recoloring with various color coding options, van der Waals/solvent-accessible dot surfaces and CPK-like space filling renderings of molecules or substructures. Substructures are chosen and manipulated as atom sets, which are constructed in SETS submode. SETS provides options for selecting various basic atom sets and applying Boolean operations to specify desired atoms uniquely. Other operations conducted from ANALYZ mode include rigid and flexible least squares atomic superimpositions, internal coordinate measurements, NMR coupling constant calculations and included/excluded volume mapping.

The actual implementation of MacroModel was done in FORTRAN on a Digital Equipment Corporation VAX 11/780 computer. We made an effort to speed structural display but this was accomplished at the expense of software portability. Hence MacroModel has generally been coded using a number of the VMS extensions to FORTRAN-77 to allow fast execution of graphics operations via standard methods including VMS-QIO calls and output buffering. To facilitate use of the system, MacroModel does nearly all of its graphics operations on the host computer and therefore may be operated using a variety of graphics devices ranging from Tektronix 4107 terminals or emulators to Evans and Sutherland PS300 graphics stations providing real-time translation and rotation of molecular structures. The graphics routines have been designed so that essentially any operation carried out on the powerful Evans and Sutherland displays can also be accomplished using inexpensive graphics terminals, though not necessarily with equal convenience.

Associated with the main MacroModel graphics program are a number of auxiliary programs that provide capability for lengthy energy calculations, for force field parameter development and for reformatting structual data derived from external sources such as the crystallographic data bases. The BATCHMIN program, which performs molecular mechanics/dynamics calculations and conformational searching in noninteractive batch mode, has been coded in ANSI standard FORTRAN-77.

THE USER INTERFACE

The MacroModel user interface consists of a number of graphics viewports allowing the display of molecular structures, permanent menu options and textual input/output. To minimize the amount of information which must be provided by the user in modeling, we have incorporated a number of semiautomated features and have made extensive use of program defaults for adjustable parameters. Program options and buttons change as the various modes and submodes of the program are accessed and a visual indication of current state of user-selected options as well as program defaults is provided by button highlighting or underlining. Online HELP for each menu item is available.

Actual interaction with MacroModel is carried out by a series of graphical menu button and atom selections. Thus the program operates much like other graphically based programs in chemistry. Since MacroModel has many options, it necessarily uses an extensive set of controls. To limit the potentially bewildering array of control buttons, many relevant suboptions are presented only when a button is activated. Thus many buttons initiate one or more prompts for additional information, which are answered either textually or graphically (by indicating atoms or bonds of the displayed structure using a data tablet or mouse). Other buttons operate differently depending upon the number of times they are activated consecutively. The user is informed of the current state of multiple hit buttons by messages that appear in an interactive area at the top of the screen. Program output is generally summarized on the display screen. More extensive summaries of energetic results are listed to disk files for subsequent examination.

Although many modeling operations may be done interactively, lengthy computational tasks are best accomplished as noninteractive batch jobs. Most of the programs that carry out such calculations are rather unfriendly and have extensive command languages which users must master before effective modeling can be done. Such processing is often problematic for new users because noninteractive computing provides feedback only after the full modeling job has been completed. To minimize such difficulties, we have provided MacroModel with ways of operating noninteractively.

One approach to noninteractive modeling is to use the nongraphical batch mode modeling program BATCHMIN. Although BATCHMIN has the usual extensive command language, command files for common tasks may be set up graphically and jobs initiated from MacroModel by selection of the BATCH button in several of the relevant program submodes. Such tasks include energy minimizations and conformational searching.

Alternatively, MacroModel itself can be run noninteractively. During standard interactive operation, MacroModel creates a log file containing all the information (button and atom selections, textual input, etc) supplied by the user. After the interactive session is complete, the log file may be designated as a "script file," which is accepted by MacroModel as input and which will cause the program to run automatically. By modifying the script file (e.g., to a select a different starting molecule or modify parameters) created during a particular modeling session, related operations may be carried out automatically. We call such jobs 'script runs' and they may be carried out as fully automatic graphics jobs or, more usefully, as standard batch jobs.

CONSTRUCTION OF MOLECULAR STRUCTURES—INPUT MODE AND ASSOCIATED SUBMODES

INPUT mode is used for the construction and modification of three-dimensional molecular structures. Applicable structures range from small organic molecules to macromolecular biological compounds. Since all molecules are handled as atom-based and not residue-based entities, hybrid structures such as glycopeptides may also be assembled and manipulated using standard program options. The modeling of simple complexes of metal atoms and organic or bioorganic ligands is also possible although coordination compounds in general are not handled at this time. Retrieval of structures from crystallographic databases can be performed and the structures can then be subjected to interactive modification using standard input options or carried over into the energetic or display modes of the program.

New chemical structures are supplied to the system by manual sketching with the graphics input device or by semiautomated fragment linking. We will not detail our structure sketching input method since it is standard in chemical graphics systems. We note only that reasonable, hand-drawn X,Y-projections serve as starting points for subsequent modification along the Z-axis (see Fig. 1, button group E). These modifications are made by activating the $M + Z$ or $M - Z$ button and then pointing to those atoms which are to be moved respectively out of or into the plane of the display screen (defined as the X,Y plane). Such starting geometries need not be particularly accurate but they should pyramidalize tetrahedral centers to define stereochemistry properly. Addition of hydrogens and lone pairs is carried out automatically. In addition to structural input by manual sketching, MacroModel provides a number of features for semiautomated construction or modification of molecular structures.

Biopolymer Construction

Fragment-based construction of biopolymers is implemented for the usual biological residues (amino acids, nucleotides, and monosaccharides). These residues can be linked automatically to "grow" the corresponding oligopeptide, nucleic acid or saccharide with idealized three-dimensional structure. During structure growing, user-defined secondary structural information is combined with a user-supplied primary sequence to generate the desired structure with the desired conformation as described below.

There are three submodes for constructing biopolymers (PEPTID—for assembly of peptides or small proteins, NUCLEI—for double- or single-stranded nucleic acids, and CARBOH—for straight-chain or branched oligosaccharides). Chemical structures formed in one of these submodes can be linked to those formed in any other submode. Novel functionalization of these structures using fragments from the ORGANI submode (used for the construction of "organic" structures) is possible as well as by manual sketching of the desired structural modification. For example, ORGANI submode can be used for the incorporation of N-methylated peptide linkages by selecting the CH4 (methane) fragment and then pointing to the amide hydrogens to be replaced by methyls. All fragments are read into the program from standard ASCII data files and can be modified to permit the user to incorporate novel fragments. Each fragment is loaded into the program as an atom connection table and a set of Cartesian atomic coordinates.

The operations and algorithms used in the above submodes are similar and we shall therefore use the construction of an oligopeptide to illustrate the general procedures involved in building a biopolymer. Figure 2 shows the PEPTID submode menu. Buttons specific to this submode provide control of conformation (PHI, PSI, S CIS, HELIX, SHEET, R TURN) and sequence (ALA, ARG,...). Activation of the GROW button turns on automatic oligomer growing. Thus as amino acid residues are selected from the menu, they

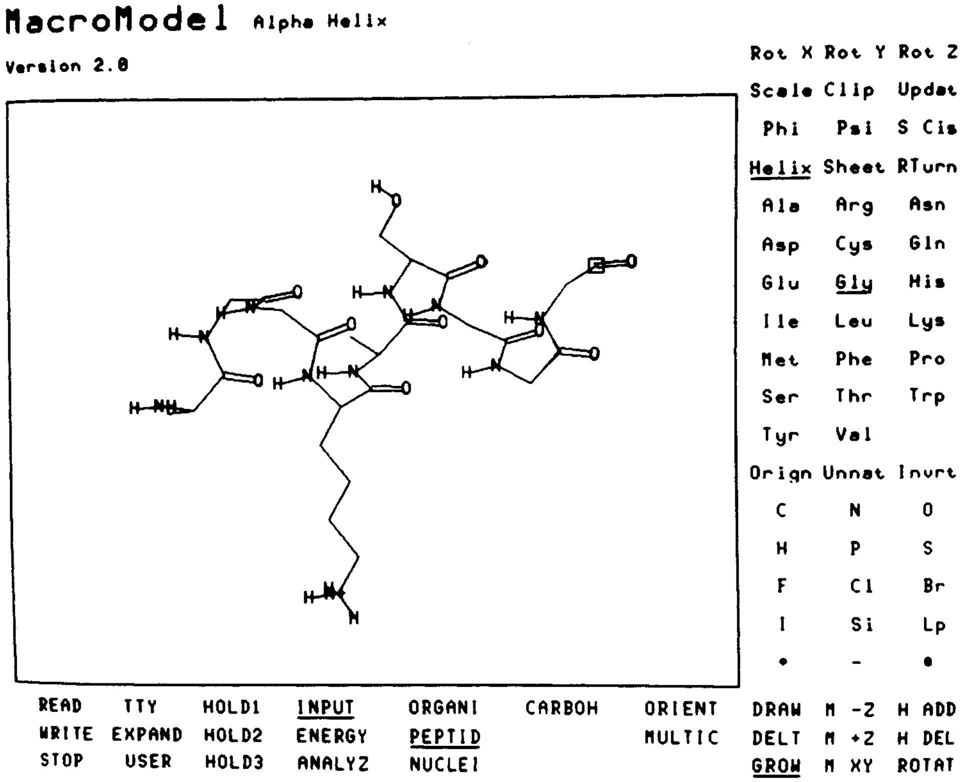


Figure 2. PEPTID submode showing the GROW option. The polypeptide is "grown" from the N to C terminus in an α-helix conformation.

are automatically added to the C-terminus or some other user-defined origin of the growing chain. For peptides, the default growing direction runs from N to C. A number of common secondary structure comformations are available, but in the example shown in Figure 2, the HELIX option has been activated, causing the peptide to be grown to adopt an idealized α-helical conformation. HELIX is a multiple hit button. Depending on the number of times the button is activated consecutively, other helical conformations including the 3-10, π, and collagen helices may be selected. At any point during the growing process, the torsional settings used to define the backbone conformation can be reset to yield other secondary structural segments. Although the default amino acid configuration is (L), activation of the UNNAT option results in the addition of (D) residues to the growing end of the chain. *S-trans* amide linkages are normally produced but may be switched to *s-cis* using the S CIS button.

Using the options summarized above, construction idealized three-dimensional oligopeptide structures is straightforward. Thus after the GROW option and a secondary structure have been set, selecting to a series of residues from the menu generates the corresponding three-dimensional structure interactively. The point at which the next amino acid will be added is indicated by a small box as shown at the right hand, C-terminal end of the peptide shown in Figure 2. If desired, that growing point may be reset using the ORIGN button. After the complete sequence is generated, the appropriate end groups (ammonium, carboxylate, etc.) are added usually via modifications made in ORGANI submode. For building proteins or large oligopeptides, noninteractive script runs are advantageous because both the sequence and conformational information contained in the script file may be easily entered and modified as necessary to build desired structures.

Changes in geometry or chemical structure may be made at any time. Geometrical changes may be made in either external or internal coordinates. Cartesian movement of atoms or molecules is accomplished by selecting the *MXY* button, pointing to a current atomic position and then to a desired atomic position to give movement in the plane of the display screen. Movement perpendicular to the plane of the screen is controlled by the $M + Z$ and $M - Z$ buttons and moves atoms or molecules incrementally by a user-determined distance (default = 0.5 Å). Movement along the Z-axis can be visually monitored by moving atoms or molecules with the display operating in stereo mode. Structural modifications made in stereo result in immediate updating of both images of the stereo pair and facilitate constructing reasonable three-dimensional starting structures even with simple displays. Like many buttons in MacroModel, *MXY*, *M* + *Z*, *M* − *Z* are multiple function buttons. If any of these buttons is activated twice, then the translations described above are applied to whole molecules and may be used to control relative molecular orientations. Other geometrical transformations, in either external or internal coordinates, can be made from ORIENT submode (see below).

Internal coordinate modifications, including relative molecular rotations and bond and torsion angle variations, are controlled by the ROTAT button. Global or local torsional searches for low energy geometries of user-selected bonds may be initiated using the TSRCH button. When this button is activated, a user-specified torsional angle is varied in user-defined increments and the approximate strain energy (see EPROX in later section) for each rotamer is computed. After all specified geometries have thus been generated and evaluated, the structure is set to the lowest energy geometry found. More automated geometrical modifications are also possible and will be described below as part of the discussion of ORIENT and MULTIC submodes.

Once a molecular structure has been assembled with the desired geometry or has been read in from an external file (e.g., from the Brookhaven Protein Data Bank[6] or Cambridge Crystallographic Database[7]), it can be structurally modified by either using the substructure fragments or manual sketching. With biopolymers for example, residues can be either replaced or modified to incorporate unnatural features. When the program is used to replace existing residues with new ones, the new residue is selected from the menu and then substituted by picking an atom of the existing residue.

A completely analogous set of procedures are available for constructing and manipulating oligonucleotides and oligosaccharides. Within the former set of compounds, the user can build either single- or double-stranded DNA or RNA. In the case of double-stranded growth, the user specifies the residue sequence for one chain, working in the 5′ to 3′ direction and the program constructs the complementary chain along with the one specified explicitly. Some modeling operations such as oligonucleotide helix unwinding require the simultaneous adjustment of multiple dihedral angles. Such operations are facilitated in MacroModel by the use of a "gang" feature, which allows the user to place any number of dihedral angle variations on a single PS300 dial which will rotate them in concert.

MacroModel also possesses a submode, CARBOH, which allows the semiautomated assembly of linear or branched oligosaccharides. Although there are separate modes for organics and the various classes of biopolymers, all structures are treated as simple atom-based entities, and molecules of any class may be brought into any submode to be modified or manipulated.

Organic Small Molecule Construction

The implementation of the substructural fragment assembly method of building biopolymers is straightforward because the rules for their construction are relatively simple. In contrast, molecules traditionally classed as organic may possess a variety of complex ring assemblies having various substitution patterns. To aid in construction of such molecules, ORGANI submode contains more generalized construction strategies that minimize the number of substructural fragments required to build organic molecules. The buttons necessary to carry out such constructions are shown in Figure 1 as button group F. The fragments we use include common organic functional groups (e.g., C=O [carbonyl], C=C [alkene], Acid [carboxylic acid], OH2 [hydroxyl]) plus a number of saturated and unsaturated carbocyclic rings (e.g., CPROP [cyclopropane], CBUT [cyclobutane], C6H6 [benzene]). Such fragments can be connected directly by formation of new bonds (CNNCT), fused together to superimpose atoms and bonds (FUSE) or used to substitute hydrogens or other substituents of existing structures. The atom symbol buttons allow changes of atoms types. With simple fragments and unit operations such as these, simple substructures may be used to construct almost any complex organic molecule in a semiautomated fashion. The construction procedures are best illustrated by an example: the bridged polycyclic adamantanone (Fig. 3).

To assemble adamantanone (Fig. 3), the user first selects the cyclohexane fragment from the ORGANI template (Fig. 1, CHEX button). By indicating a blank area on the display screen, the user places the cyclohexane fragment in the drawing area. Selection of a methane fragment (CH4) followed by picking of one of the axial hydrogen atoms on the cyclohexane results in an automatic replacement of the indicated hydrogen by the methyl from the fragment to produce axial methylcyclohexane. The algorithm used to accomplish this operation as well as other substructural fragment connections operates by geometrical construction of valence coordinates appropriate for the substitution (in this example, tetrahedral *sp*^{3}) on the indicated atom (here, the axial hydrogen) followed by a least squares superimposition[8] of fragment atoms and valence coordinates. Two repetitions of the procdure af-

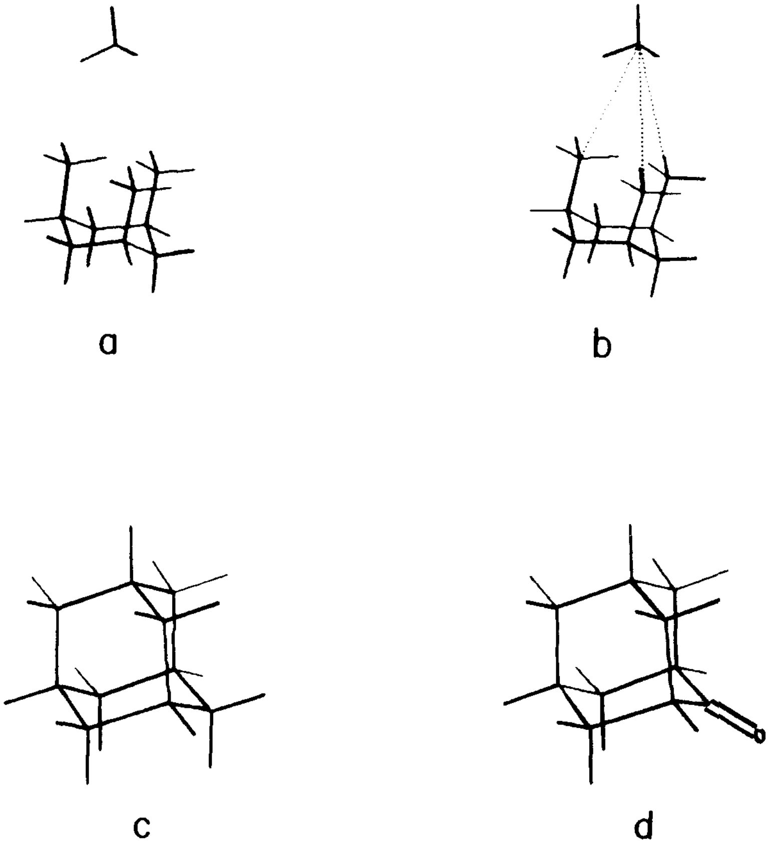


Figure 3. Consecutive structure viewport showing the construction of adamantone using the Cnnct option.

ford triaxail 1,3,5-trimethylcyclohexane, which can be then converted to adamantane by "connecting" a methane fragment simultaneously to the three axial methyl carbons (Fig. 3(a)). This is performed by selecting the CNNCT button and specifying the atom pairs to be joined. The function of CNNCT is to join indicated atom pairs with a single bond using standard bond lengths, angles and torsions. After picking the three pairs of atoms to be connected (Fig. 3(b)), activating the START button generates the adamantane ring system (Fig. 3(c)). The final transformation of a methylene group into a carbonyl requires selecting the carbonyl (C=O) fragment followed by picking the carbon atom of the methylene group to be replaced. Atoms of the carbonyl fragment are superimposed upon those of the indicated methylene group and hydrogen atoms are removed together with the methylene carbon. The net result is replacement of methylene by carbonyl to afford adamantanone (Fig. 3(d)). In this manner, functionality can easily be incorporated and substituents may readily be added to existing molecular structures of any kind with idealized three-dimensional geometries.

Functionalized acyclic molecules and substituents may also be conveniently 'grown' with idealized three-dimensional geometries. For example, one may grow methyl *E*-pentenoate by sequentially selecting the menu button sequence: CH4, CH4, C=C, Acid, CH4. Branched molecules may be constructed by using the Orign button to reset the growing site to the desired branch point and then growing the desired structure as described above. Stereochemistry is generally defined by selecting the appropriate hydrogen substituent for replacement or as a new origin atom for further structure growing. The stereochemistry of existing chiral centers and olefins may be changes using the INVRT button.

Small molecules are also available from the Cambridge Crystallographic Database (CCD). Such structures are not read directly into MacroModel, but are translated by an associated program called CFILER, which creates a standard MacroModel structure file containing atomic coordinates, atoms and bond types and atomic connectivity.

There are two additional submodes of INPUT mode. They are not used for building structures per se but instead allow the manipulation of the inter- and intramolecular geometry of existing chemical structures. These submodes are called ORIENT(ation) and MUTLIC(onformer). ORIENT is used to modify the Cartesian coordinates of substructures or whole molecules by way of internal coordinate transformations. MUTLIC is used for conformational searching.

Internal Coordinate Modifications and Molecular Docking

Intermolecular reorientations (e.g., docking) and intramolecular internal coordinate adjustments (e.g., rotating bond or torsion angles) are common operations in molecular modeling. While some such transformations can be accomplished with the *M XY*, *M* − *Z*, *M* + *Z*, and ROTAT buttons found in all the INPUT submodes, more complex geometrical transformations are carried out via the ORIENT submode. Operation of ORIENT procedures typically involves: (1) selection of a set of atoms for translations and/or rotation, (2) specification of a convenient set of local Cartesian axes in which the desired geometrical tranformations are to take place, and (3) incremental movement of the selected atoms by translation or rotation with respect to the local axes. Thus to dock molecule A into a binding site in molecule B, the user first selects all of the atoms in molecule A as the set of moving atoms (see the discussion of SETS submode later). The DefAx button (Fig. 4) is then used to define a set of local axes (*X*′, *Y*′, and *Z*′). For docking a convenient local axis set might be one whose *X*′-axis runs through molecule A and the binding site of B and whose *Y*′-axis is aligned with a major cleft of the binding site. Using the Tra (translation), ROT (rotation), or T/R buttons (a reduced instruction translation and rotation option), the user can specify incremental translations along these local axes and/or rotations around the axes to conveniently move A into the desired orientation within the binding site of B. Each transformation is accomplished by immediate updating of the display. On the Evans and Sutherland PS300 series displays, such operations may be carried by simply adjusting the dials which control molecular translations and rotation. However, the incremental methods available in ORIENT are more general and provide more precise control of geometrical manipulations. They also allow such transformations including docking to be carried out on inexpensive terminals.

ORIENT maneuvers may also be monitored energetically and accompanied by a graphical report of the relative energies of the various geometries explored. The graphical report is in the form of a bar graph with signed ln(energy) appearing on the vertical axis and internal coordinate changes along the horizontal axis. Figure 4 shows such a display for the docking of imidazole into the binding site of a larger host structure. The figure shows that the energy falls as imidazole becomes encapsulated within the host. In addition to providing a convenient display of en-

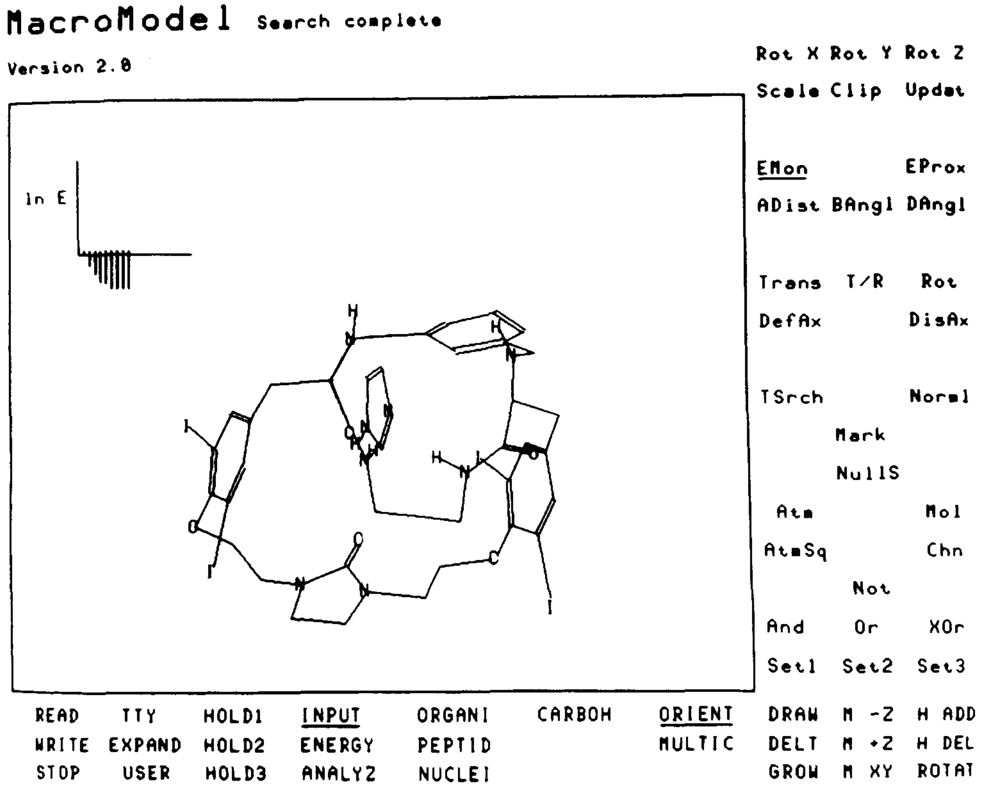

MacroModel Search complete
Version 2.8
ln E
Rot X Rot Y Rot Z
Scale Clip Updat
EMon EProx
ADist BAngl DAngl
Trans T/R Rot
DefAx DisAx
TSrch Norml
Mark
NullS
Atm Mol
AtmSq Chn
Not
And Or XOr
Set1 Set2 Set3
READ TTY HOLD1 INPUT ORGANI CARBOH ORIENT DRAW M -Z H ADD
WRITE EXPAND HOLD2 ENERGY PEPTID MULTIC DELT M +Z H DEL
STOP USER HOLD3 ANALYZ NUCLEI GROW M XY ROTAT

Figure 4. Orient submode; rigid body docking of imidazole to a synthetic host with energy monitoring.

ergies, the graphical bars may be selected via graphics input device to recover the corresponding structure. These energy-guided translation/rotation reorientations may be accomplished either manually or by using a Simplex algorithm[9] to locate the nearest minimum energy geometry automatically. In Figure 4, the docking shown was performed in translation and rotation using the Simplex search.

The energies used to monitor internal coordinate changes come from a speedy molecular mechanics calculation we call EPROX. It differs from standard molecular mechanics force fields in that it uses a small set of generalized parameters and equations which have been chosen primarily for speed. Thus bond stretching and bending energies are computed from the squares of 1,2 (bond stretch) and 1,3 (bond angle) distance to avoid time-consuming square root extractions. The program computes torsion angles only once per rotatable bond and a nonbonded pairlist having precomputed constants is used for rapid evaluation of distance-dependent dielectric electrostatics and Lennard–Jones van der Waals energies. For intermolecular, rigid body translations/rotations, only the nonbonded component of the energy is evaluated. The cpu time for computing the total EPROX energy of a 500 atom structure is ca. 2 cpu seconds on a VAX 11/780. For rapidly computing the energy of complexes of small molecules with molecules the size of biopolymers, we use algorithm described by Levitt et al.[10] Thus prior to docking or other internal coordinate manipulations, the factored nonbonded (van der Waals and electrostatic) interaction potentials are computed at points within a 25 × 25 × 25 Å box which surrounds the atoms to be moved. These potentials are then used to estimate interaction energies in real time. The energy functions used in EPROX are summarized in Appendix A.

Conformational Space Searching

Perhaps the most difficult aspect of molecular modeling is the extreme complexity of the conformational space available to most molecules and the attendant problem of finding the various populated, low energy minima. There are many approaches to this problem. The zero-order solution, finding the global minimum, is less useful than one might expect because most molecules have many populated conformers which contribute to their properties. Additionally, the specific conformer having the global minimum energy varies depending on the nature of the potential functions and the parameters, as well as upon any treatments of solvation and entropy. Searching for the set of all minimum energy conformers (or at least representative subset) within some given energetic bounds is a better approach.

The MULTIC submode is used for searching the conformational space available to small molecules with the objective of finding the low energy conformers. It operates by allowing the user to indicate relevant internal coordinates for variation (typically low barrier torsion angles) and then to use one structure to generate a large number of new starting geometries which are distributed throughout conformational space subject to any user-defined constraints. These constraints typically direct the search away from structures having high energy nonbonded contacts but may also select geometries consistent with NMR coupling constants and/or nuclear Overhauser effects. The generation of such starting geometries is fast. Depending on the nature of the structure and constraints, MULTIC generates 1-100 starting geometries/CPU second on a VAX 11/780. After the starting geometries are generated and saved as a disk file, the batch mode modeling program BATCHMIN is used to minimize the energy of each structure in the file and store the final set of unique, low energy minima. A conformer is defined here as unique when a least squares superimposition of its atoms with those of each previously found structure gives at least one atomic superimposition with a deviation of >0.25 Å. A detailed account of algorithms used in MULTIC and their performance characteristics has been presented elsewhere.[11]

Whereas the MULTIC method is effective in searching conformational space of molecules having as many as six or seven rotatable bonds, searches of more flexible molecules results generating too many initial structures for energy minimization in a reasonable length of time. For such structures, major reductions in the scope of the search are necessary. One way to limit the search is to provide geometrical constraints taken from experimental measurements (e.g., NMR spectra). These constraints are typically supplied as internuclear distances or angles and can result in reductions in the number of geometries generated amounting to many orders of magnitude. Care is required, however, since the NMR data is usually averaged over many conformers and inferred geometrical data may not reflect the structure of any single conformer. Using coarse dihedral angle resolution (e.g., 120°) is another way to limit the search. This alternative, while typically producing many different minima, usually misses some of the low energy conformers and thus at best leads to a representative set of conformers. The best alternative currently appears to be the internal coordinate Monte Carlo conformational search feature of our batch modeling program BATCHMIN.

That method appears capable of finding all populated minima for structures having as many as 10–12 rotatable torsion angles.[12]

In concluding our comments on conformational space searching with MacroModel, we wish to reiterate that the multiple minimum problem is one that needs serious attention when modeling flexible molecules. There are numerous instances in the literature of modeling studies having conclusions drawn from structures which turn out to be high in energy relative to an undiscovered global minimum. Even analysis of a compound based on the true global minimum energy conformer can be inappropriate if the molecule is highly flexible. Other conformers may make major contributions to its properties. While a general solution to the conformational multiple minimum problem remains elusive, the methods outlined above provide solutions to conformational searching with small-to-medium sized molecules.

Summary of INPUT mode

To summarize this section on the input and structural/geometrical modification of molecules, we have assembled in MacroModel a comprehensive, graphically-based scheme for the construction of molecules and for modifying existing molecular structures and geometries in a convenient and semiautomated way. Some graphical molecular modeling programs stop here and address primarily those problems associated with manipulating and examining large molecules such as biopolymers. Such programs thus function as qualitative tools for molecular modeling, and both their means and their ends are primarily graphical. In MacroModel, the reason for dealing with molecules graphically is to facilitate analysis and prediction of their properties quantitatively using energy calculations. Our means are also graphical but our ends are energetic and MacroModel energetics form the subject of the next section.

COMPUTATIONAL ALGORITHMS FOR MOLECULAR MECHANICS AND DYNAMICS—ENERGY MODE AND SUBMODES

With MacroModel, we compute structures and energies using molecular mechanics. Rather than detail the model here, we refer the reader to the appropriate monograph[13] and describe only the novel features of the MacroModel molecular mechanics implementation. These features are controlled by the second main mode of the program which is called ENERGY. In the paragraphs below, we will discuss the MacroModel force fields, force field parameterizations, energy minimizations and molecular dynamics.

MacroModel is used primarily for interactive energy calculations, minimizations and dynamics while the associated batch mode modeling program BATCHMIN accomplishes these functions via noninteractive batch jobs. Since quantitative molecular modeling requires much computer time, interactive energy computations might appear inappropriate. There are, however, a number of advantages to interactive energetic capabilities. First, small molecules can be modeled quantitatively in a few seconds or minutes to provide rapid feedback of molecular geometries and energies to users engaged in analysis/design activities. Second, development and refinement of molecular mechanics parameters for new molecular substructures and functionalities are handled effectively and conveniently as an interactive process. Third, preparation of computationally intensive jobs is facilitated by conducting scaled-down test calculations, evaluating parameters and options, and performing various other control experiments prior to the full-scale job. These operations, among others, are intrinsically interactive operations in which a variety of computational questions must be asked and the answers obtained direct what one does next.

To facilitate interactive energetic calculations, we have made common modeling operations easy to perform. Energy minimization, for example, requires only that the user supply a crude starting geometry as outlined above, select the MINIMIZ submode of ENERGY (Fig. 5), choose the appropriate force field (MM2, AMBER, etc.) and activate the Start button. Other options (minimization algorithm = block diagonal Newton Raphson, convergence criterion = gradient of 0.02 kJ/Å, etc.) are set by default. The minimization will then run interactively to converge the structure to a final rms gradient of 10^{-2} kJ/Å. The energy (kJ/mol) and extent of atomic movement (in units of 10^{-5} Å) is displayed every five iterations during the minimization and the user is allowed to halt the minimization every 50 (or some other number) iterations. The total steric energy and its components are listed at the top of the screen. The minimization terminates automatically when the operative convergence criterion is achieved. Other interactive operations within the ENERGY mode include, inter alia, molecular mechanics force field modifications, normal mode analyses, harmonic free energy calculations, molecular dynamics simultations, and manual substructural conformational searches.

MacroModel is not an interface to external programs that handle energetics but rather is a self-contained processor of molecular mechanics force fields. Thus it is not wedded to any particu-

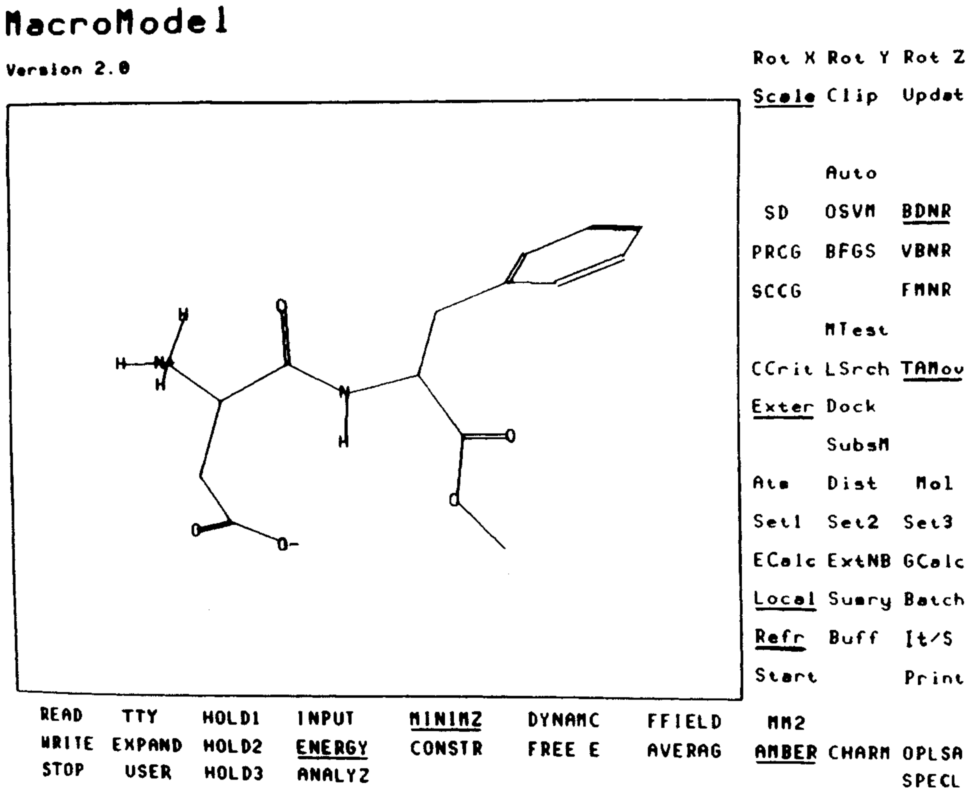


Figure 5. Minimization of Aspartame using the united atom AMBER force field. Minimization algorithm is SOR BDNR with terminal atom movement.

lar molecular mechanics equation/parameter set, to any particular type of molecule or to any of the optimization algorithms associated with some standard field. Instead, MacroModel is equipped with a set of standard molecular mechanics equations from which the appropriate ones are automatically selected when the program reads in an external force field file. The file contains the parameters found in standard molecular mechanics force fields and allows MacroModel to function as if its potential surface were, for example, MM2,[3] AMBER,[4] or OPLS/AMBER.[5] The program contains its own energy minimization and molecular dynamics modules which are used for molecular modeling with whatever force field the user selects.

Molecular Mechanics Force Field Implementation

The force field files are a most important part of the MacroModel energetic implementation because they contain all the data necessary to compute a molecular mechanics energy from a three-dimensional molecular structure. To handle the general case of an arbitrary structure to which must be assigned a specific array of force constants, changes, etc., most molecular mechanics systems use special atom types to distinguish one structural environment from another. The problem with such a system is that either the user or the software must assign highly structure-dependent atom types whenever new molecules are to be studied. MacroModel uses a different approach in which atom types describe only local structure (i.e., the chemical element and its hybridization) while an external force field file provides both the parameters and the associated structural environments. As described below, the structural descriptors in the MacroModel force field files have been designed for generality. They allow the automatic assignment of specific or general molecular mechanics parameters to virtually any part of any structure.

Our force field files are organized into three sections as illustrated in Appendix B, which shows the portions of the MacroModel AMBER force field file. The first section contains data that directs the program to select the correct potential functions from the available molecular mechanics equations for the stretching, bending, nonbonded, etc. contributions to the total energy. It also contains conversion factors that change the force field author's choice of units to kJ/mol. The second section is typically the largest part of the file. It contains subsections with standard force field parameters for stretching, bending, torsion, improper torsion (or out-of-plane bending), nonbonded and hydrogen bonding interactions. The last section of the file provides the field with structure-specific parameters for specialized substructures. A special substructure may include any molecular substructure up to (currently) 25 atoms and is defined by a simple linear notation. Whenever a special substructure is found within a structure, the associated special parameters replace the standard force field parameters in energy calculations. We use such substructures to provide parameters for the various biopolymer residues (e.g., amino acids and nucleic acids), aromatic ring systems and small rings. MacroModel electrostatics employ the atomic partial charge model and charges are normally computed from atomic formal charges and from bond dipoles stored in the stretch section of the force field. Charges may also be taken from explicit partial charge sets associated with the special substructures.

The force field file itself is a simple ASCII disk file which may be readily edited to modify existing parameters or add new ones. As noted above, the first section of the force field file defines the form of the force field by selecting the appropriate equations and energetic conversion factors. In Appendix B (Section 1) for example, the line reading '0 STR 1 8.36800' activates stretching equation 1 (harmonic stretch) and converts the AMBER stretching force constant ($k_{str}/2$ in kcal/mol) to k_{str} in kJ/mol. The lines beginning '0 SEL' and '0 ALT' control parameter line reading from subsequent sections of the field and allow the user to select subsets of parameters based on their quality and/or origin. The last part of section 1 is labeled "Equivalenced Atom Types" and defines certain symbols (e.g., CS) as being equivalent to any of the following standard atom types (e.g., CA (united-atom methine), CB (united-atom methylene), CC (united-atom methyl)). These equivalenced symbols reduce the number of force field entries necessary to describe all possible arrangements of related closely atoms. The lines beginning with "C" are comments and are not used by the program.

Each line in the second section of the force field file holds parameters for a particular molecular mechanics interaction and a description of the atomic environment to which the parameters should apply. Atomic environments are defined by the arrays of atom and bond types given in the entries of the force field file. Generally like MM2 and unlike AMBER, an atom type in a MacroModel force field represents only the chemical element, formal charge and hybridization. A partial listing of the codes used for atoms and bonds is given in Table I. The undefined atom and bond types (00 and * respectively) are used as wild card descriptors to which any atom or bond will match. Wild card atom and bond descriptors are usually associated with generalized parameters which are accessed when no match for a specific molecular interaction is found in the force field file. For example, the bond angle descriptor 00-C3-00 defines a generalized bend for any sp^3 carbon (all-atom model). We have included an extensive set of generalized parameters in our force field files so that most molecular structures may be modeled qualitatively without adding new parameters.

Table I. Partial listing of atom types and bond types used in MacroModel.

C3	Carbon (sp^3)
C2	Carbon (sp^2)
C1	Carbon (sp)
O3	Oxygen (sp^3, —O—)
O2	Oxygen (sp^2, =O)
OM	Oxygen anion (—O—)
N1	Nitrogen (sp)
N2	Nitrogen (sp^2)
N3	Nitrogen (sp^3)
N4	Nitrogen cation (sp^2)
N5	Nitrogen cation (sp^3)
H1	Hydrogen (of —C—H)
H2	Hydrogen (of —O—H)
H3	Hydrogen (of —N—H)
H4	Hydrogen (of —N$^+$—H)
CA	Methine (sp^3, united atom CH)
CB	Methylene (sp^3, united atm CH$_2$)
CC	Methyl (sp^3, united atom CH$_3$)
CD	Methine (sp^2, united atom =CH—)
CE	Methylene (sp^2, united atom =CH$_2$)
00	Undefined atom type

—	Single bond
=	Double bond
%	Triple bond
*	Undefined bond type

Shown below are typical environmental descriptions from a force field file which define particular two-atom stretches (1, 2), a three-atom bend (3) and a four-atom torsion (4) respectively:

Atom/bond descriptors		Optional atom descriptors	
C3—C2	···	0000 O200	(1)
C3—C2	···	0000 0000	(2)
C3—C3—C3	···	0000 H1H1 0000	(3)
C2=C2—C2=C2	···	0000 0000 0000 0000	(4)

For the stretches (1) and (2) to match a particular bond in the molecule, that bond must be a single bond joining sp^3 and sp^2 (all-atom) carbons. Bond orders in the force field are explicitly matched unless they are indicated by a wild card bond symbol (*). For the torsion (4) to match a part of a molecule, it must map exactly into the molecule's atoms and bonds as shown. This example illustrates the substructure required to set torsional constants for the central bond of conjugated butadiene-like fragments.

The field labeled "Optional atom descriptors" provides additional specificity in the assignment of force constants to arrays of atoms and bonds. Each four-character string defines up to two additional atoms which must be attached to the corresponding (first, second, etc) atom in the atom/bond descriptor field for the interaction to match. In (1) , the 0000 (00 is the wild card label for any atom type) means that no specific additional atoms are required to be bound to the first atom (C3) in the stretch descriptor. The second optional atom descriptor, O200, indicates that an O2 (carbonyl oxygen) must be attached to the second atom, the C2. Thus optional atom descriptors allow in example (1), the carbon-carbon stretch of a C(sp^3)—C=O, to be distinguished from all other C(sp^3)—C(sp^2) stretches. The force constants used in a calculation come from the first valid match to be found in the force field file. In the bend example (3), the optional H1H1 requirement for the central carbon (—C3—) matches only when the central carbon is an explicit methylene. Such distinctions are required by force fields like MM2. In the stretch section of the AMBER field shown in Appendix B, the carbon-carbon bonds of amides and carboxylates are distinguished in this way. Thus, by requiring explicit matching of bond types and optional atom types, it is possible to assign force constants properly to many structural components of complex molecules.

While specific atoms, bonds and optional atom descriptors allow many small structural entities to be distinguished, larger entities such as amino acids and aromatic ring systems require more information if specific parameters are to be assigned to them. For these systems, we use structural descriptors we call 'special substructures.'

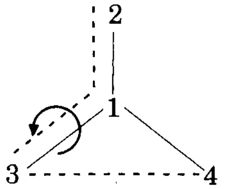
Special substructures are defined by a simple linear notation which is exemplified by notations below:

Linear notation	Corresponding molecule or substructure
O3—C3—C3—1	Ethylene oxide
N2=C2—C2=C2—C2=C2—1	Pyridine
C2*C2*C2*C2*C2*C2(*1)*C2*C2*C2*C2*5	Naphthalene
O2=C2—CA(—CB—O3—H2)—N2	Serine (united-atom)

In this notation, each two-character symbol corresponds to an atom type (Table I). Bond orders are matched explicitly and the numerals denote bonds back to previous atoms in the sequence numbered left to right. In the naphthalene notation, the wild card bonds (*) are used so that the substructure given will match either of the two naphthalene resonance forms. The ordered form of the substructure notation facilitates use a simple tree-search algorithm for matching substructures which is rapid even in large, complex molecules. Once the atoms in the structure have been matched to the corresponding substructural notation, any specific parameters associated with the substructure in the force field can be assigned directly to the appropriate parts of the molecule.

The special substructure capability of MacroModel is a major advantage to many users of molecular mechanics in that it allows the ready incorporation of specialized parameters for molecular structures of special interest. Thus, essentially any substructure may be permanently provided with a geometry, rotational barriers, charges, etc. which reproduce whatever data is available. It has been used in our AMBER implementation of the amino and nucleic acid parameter sets and in our MM2 field for various aromatic ring systems. In this context, we have avoided an MM2PI-like approach[14] to conjugated systems and instead provided the special substructural scheme as a method for incorporating specific structural and energetic data directly into the force field. Examples of substructure use in AMBER are shown in Appendix B, Section 3. In the imidazole and valine substructure examples, the entries give specialized bond dipoles for any optional C—H bonds and nonhydrogen atom partial charges to generate the published partial charge sets. These entries will match both united-atom and all-atom representations of imidazole (e.g., in histidine) and valine (e.g., in peptides or proteins).

In our implementations of published force fields, we have reproduced the authentic force field energetics as faithfully as possible with the only exceptions being the form of electrostatic and improper torsion (out-of-plane bending) equations used, and the handling of nonbonded cutoffs. Like most molecular mechanics force fields (MM2 being a notable exception), the MacroModel fields employ the partial charge model for electrostatic interactions. Depending upon the user's choice, either constant dielectric or distance-dependent dielectric treatments may be used. In our MM2 implementation, we have retained the Coulomb's law, partial charge model in place of the MM2 dipolar electrostatics treatment. We made this substitution because we know of no convincing evidence of the superiority of the dipolar model in reproducing molecular energies, and because partial charge electrostatics are simpler and significantly faster to evaluate. Partial charges for our implementation of the MM2 field are computed from the MM2 bond dipole moments. A second and relatively minor substitution for our MM2 field is the replacement of the out-of-plane bending term with an improper torsion term which has been scaled to reproduce MM2 energy differences for small out-of-plane deformations. The improper torsion model has the advantage of simplifying energy and derivative evaluations since it is handled computationally like a standard proper torsion. In the diagram shown below, the MacroModel improper torsion is given by the dihedral angle defined by atoms 2-1-3-4. It may be used to favor either planar or nonplanar geometries of atom 1.

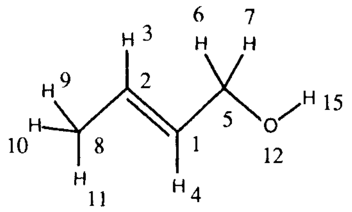


Finally, MacroModel employs cutoffs for nonbonded distances that limit the evaluation of van der Waals and electrostatic interactions to those which make significant contributions to the energy. By default, our cutoffs are centered at 7 Å for van der Waals and 12 Å for electrostatics with a linear reduction (i.e., a soft cutoff) over a range of ±1.0 Å from these distances. Alternatively the user may specify ExtNB (extended nonbondeds: van der Waals to 8 Å, electrostatics to 20 Å) or any other explicit nonbonded cutoff distances. By setting the cutoffs to distances larger than the maximum dimension of the molecule, the authentic MM2 protocol is followed.

Since both molecular mechanics parameters and the associated substructures come from an external file, adding or modifying parameters is straightforward. To aid in parameter development, a submode called FFIELD is incorporated into MacroModel. The main function of this submode is to allow the user to interactively modify the contents of the program-stored image of the force field. Like most of the other submodes, FFIELD is operated graphically. To modify a bend parameter for example, the user activates the Bnd button, and selects the three atoms in the molecule defining the bend of interest. The program then lists the current values of the natural angle and force constant and asks for modifications. This feature allows users to adjust force field parameters, energy minimize appropriate test structures with the modified parameters, analyze their energies or geometries and then iteratively readjust the parameters until the calculation fits the data to the desired degree of precision.

A perpetual problem in molecular mechanics is that parameters for many common substructures are not available or, if they are, they have not been tested or optimized adequately. To allow users to evaluate the parameters being used in a specific calculation, the force field contains information describing the origin and quality of each parameter. To specify the origin of parameters, we use the symbols "O" for original, indicating parameters published by the author of the field, "M" for modified from the original values, and "A" for new parameters added by other researchers. To define the reliability of the parameters, we use the symbols "1" for the most reliable, fully tested parameters, "2" for tentative parameters which have been tested against experimental data on a few compounds, and "3" for parameters which may be reasonable but are generally untested. Generalized parameters as described above are usually given the low quality designation "3." In addition to these one-character descriptors, each parameter line has an associated text string which describes the substructure and, if the parameter is not an original one, gives a reference to the literature or other source. Appendix B, section 2 shows such descriptors in our AMBER force field as the "O1," "O2," and "A2" labels found toward the right end of the parameter lines.

The origin, quality and textual descriptors of the force field parameters may be accessed by users in several ways. The simplest way for a user to find out about the parameters being used with a given molecule is to compute its energy and then select the Sumry button in FFIELD submode. This action will cause the program to tabulate and list the number of original (labeled O), modified (M) and added (A) parameters, and the number of quality 1, 2, and 3 parameters used in the energy calculation. This summary breaks down the parameter descriptors into stretch, bend, torsion, and improper torsion contributions. Since stretches, bends, and improper torsions do not contribute much to the relative energies of conformers, it is the quality of torsional parameters which is most directly related to the accuracy of the total energy. A summary noting use of low quality (3) torsions indicates an energy calculation (and minimized geometry) which is qualitative at best. The user may find details concerning the quality of parameters used in the energy listing which is controlled by the Print button in MINIMIZ submode. In addition to tabulating all interactions, internal coordinates, parameters, etc., the listing contains notations of the quality of parameters, their origin and a brief textual description, often including relevant literature references. These descriptors may be seen at the ends of the inter-



```
Dihedral Angles and Torsional Energies     18 Torsional Interactions Present

Atom Numbers      Force Consts            Angle      Energy
               V1      V2      V3                  kJ/mol  Select  Alts  Comments

1  2  8  9    0.000   0.000  -0.240      0.0839    -1.004   O2     0    C=C-C-H
1  2  8 10    0.000   0.000  -0.240    120.3869    -1.004   O2     0    C=C-C-H
1  2  8 11    0.000   0.000  -0.240   -120.2222    -1.004   O2     0    C=C-C-H
1  5 12 15    0.000  -1.000   0.090   -179.3581     0.000   M2     0    C(sp2)-C-O-H, WCS (CU)
2  1  5  6    0.000   0.000  -0.240      5.2303    -0.985   O2     0    C=C-C-H
2  1  5  7    0.000   0.000  -0.240    125.4547    -0.984   O2     0    C=C-C-H
2  1  5 12   -1.000   0.000   0.000   -114.9473    -1.210   M2     0    C=C-C-O, Allylic ether, WCS (CU)
5  1  2  3    0.000  12.500   0.000     -0.4112     0.003   O1     0    C-C=C-H, JCC, 1051 (87)
3  2  1  4    0.000  15.000   0.000   -179.5822     0.003   O1     0    H-C=C-H
3  2  8  9    0.000   0.000   0.520   -180.0000     0.000   O1     0    H-C(sp2)-C-H
3  2  8 10    0.000   0.000   0.520    -59.6937     0.000   O1     0    H-C(sp2)-C-H
3  2  8 11    0.000   0.000   0.520     59.6972     0.000   O1     0    H-C(sp2)-C-H
8  2  1  4    0.000  12.500   0.000      0.3351     0.002   O1     0    C-C=C-H, JCC, 1051 (87)
4  1  5  6    0.000   0.000   0.520   -175.5764     0.029   O1     0    H-C(sp2)-C-H
4  1  5  7    0.000   0.000   0.520    -55.3520     0.032   O1     0    H-C(sp2)-C-H
5  1  2  8   -0.100  10.000   0.000    179.5062     0.003   O1     0    C-C=C-C, JCC, 1051 (87)
6  5 12 15    0.000   0.000   0.200     58.0602     0.002   O1     0    H-C-O-H
7  5 12 15    0.000   0.000   0.200    -58.8950     0.001   O1     0    H-C-O-H
```

Figure 6. Output listing from a MM2 calculation of crotyl alcohol. Only a listing of the torsional section is shown.

action lines in Figure 6, which shows part of the torsion section of an output listing from a MM2 calculation on crotyl alcohol (a simple allylic alcohol). Under the column heading "Select," several entries labeled "M2" may be found. These are interactions employing new torsional force constant parameters developed specifically for allylic alcohols. These new parameters were added to the force field by one of us (WCS) at Columbia University (CU). They are labeled as M2 (modified, moderate quality) parameters since they replace unoptimized allylic alcohol parameters which are part of the original MM2 force field.

In concluding this section on the MacroModel force field implementation, we wish to add that force field files also contain switches controlling which parameters may be read into MacroModel. These switches allow, for example, the user to omit low quality (type 3) parameters, or to select only an original (type O) parameter set, or to switch between alternate sets of parameters (e.g., MM2 vs MM2′[15] or united-atom AMBER[4a] vs. all-atom AMBER[4b]).

Energy Calculations, Minimizations, Data Structures and Derivatives

The calculation of molecular mechanics energies and derivatives is central to the molecular modeling with MacroModel. Since such calculations are typically the most time-consuming operations in molecular modeling, it is important that they be carried out efficiently. For iterative energy minimizations, speed is attained by using effective geometry optimization algorithms (i.e., by iterating as little as possible) and by computing derivatives efficiently (i.e., by iterating rapidly). During the development of MacroModel, we have had occasion to test most of the commonly used optimization algorithms and to evaluate a number of schemes for speeding the calculation of derivatives. Taken together, the various parts of the optimized system we have developed function together to accelerate the convergence of energy minimizations by 4–10 times over previous implementations of the same force fields.

Before discussing the various minimization algorithms, we should commeht on the criteria used to test for convergence, i.e., the termination condition for energy minimization. In MacroModel, the user has a choice of convergence criteria including achievement of a low gradient, low atomic movement or energy stabilization. Of these, the latter two are generally inadequate since such conditions may occur simply through the use of an inefficient minimization algorithm. In contrast, the rms gradient, a measure of the residual force on each atom, is a simple property of the force field which indicates the slope of the potential surface at a particular molecular geometry. At an actual minimum, the value of the gradient is zero by definition. In practice, achievement of a sufficiently low value for the gradient is an indication of convergence to an energy minimized structure. With molecules having several hundred atoms using single precision coordinates and derivatives, rms gradients $<10^{-3}$ kJ/Å may be achieved and are indistinguishable from zero gradients given the intrinsic errors of single precision variables and arithmetic. If conformational energies are to be compared, gradients should be reduced to at most 10^{-2} kJ/Å to assure that energies represent minimum energy values.

A key part of our optimization system is an assortment of useful algorithms for energy minimization, including successive overrelaxation (SOR) block diagonal Newton–Raphson,[16] Polak–Ribiere conjugate gradient[17] Oren–Spedicato variable metric,[18] and full matrix Newton–Raphson[16] optimizers. Each of these optimizers has its own virtues. For small molecules, the SOR block diagonal Newton Raphson (BDNR) procedure utilizing terminal atom movement is particularly effective and is closely related to the method by which the MM2 program optimizes structures. The MacroModel implementation differs primarily by the use of an SOR multiplier of 1.2 and an additional memory increment of 0.2. In solving for the new position of an atom, the SOR multiplier moves atoms 20% farther than the Newton–Raphson solution dictates, and the memory increment augments further the atomic movement by 20% of the previous iteration's solution vector. Both of these movement enhancements operate on the assumption that atoms moving in a certain direction during minimization will continue moving in that direction upon subsequent iterations. This is a reasonable assumption for single atom minimization methods like BDNR which move atoms individually rather than en masse. Before any Newton–Raphson solution is applied, it is tested for upgradient movement (e.g., toward a saddlepoint) by direct evaluation of the quadratic fit to the potential surface. If such movement toward higher energy is detected, the program reverts to either steepest descent or linesearching along the steepest descent vector. On the occasion that the augmented atomic movements described overshoot the minimum and result in an energy increase, the SOR multiplier is reduced by 25% and the iteration is repeated. These simple modifications approximately double the rate at which structures converge to final geometries having low gradients and should be useful in other programs using BDNR minimization algorithms.

For molecules having 200 or fewer atoms, the most effective way to energy minimize a structure to full convergence is first to refine the geometry to a gradient of ca. 0.5 kJ/Å and then switch the minimization algorithm to the full matrix Newton–Raphson (FMNR) method. FMNR is the most effective method we have investigated for rapidly converging to the best possible local minimum energy geometry. Farther from the minimum, we find that FMNR with double precision energetic line searching[19] down the FMNR solution vector helps avoid optimization to energetic maxima. In any event, FMNR typically gives complete convergence in <10 iterations. We solve the $3N + 6$ linear equations (the six extra equations fix translation and rotation[20]) for the FMNR solution using simple Gaussian elimination with partial pivoting[21] using single precision arithmetic. Except with very small molecules, the linear equation solution is the most time-consuming part of the FMNR calculation, but it may be accelerated significantly through the use of special machine-optimized linear equation-solving subroutines supplied by makers of computing hardware. FMNR capability has another advantage. The second derivative (force constant) matrix used by the algorithm can also be used to compute vibrational frequencies by standard normal mode analysis. These frequencies allow verification of structures as true minima as distinct from saddle point structures and allow the computation of harmonic vibrational enthalpy and entropy.

Other minimization schemes available include Polak-Ribiere conjugate gradient (PRCG),[17] self-correcting conjugate gradient (SCCG),[22] Oren–Spedicato variable metric (OSVM),[18] and Broyden–Fletcher–Goldfarb–Shanno variable metric (BFGS)[23] methods. Steepest descent is also available but it is invariably inferior to SOR BDNR[16] when minimizing energies in Cartesian coordinates. Of the two variable metric methods studied, OSVM generally converges structures faster but the speed advantage is only ca. 10%. The variable metric algorithms are limited in MacroModel to 250 atoms or fewer owing to the necessity of storing the large approximation to the inverse Hessian; however, the simple PRCG method works well for reducing the gradient of molecules having up to 500 atoms to 0.1–1.0 kJ/Å. Further reduction of the gradient is difficult using conjugate gradient methods. We have investigated more complex conjugate gradient methods but none perform significantly better than PRCG when using a quadratic, double precision energy-based linesearch.[19]

Table II. Comparison of various minimization algorithms available in MacroModel.

Speed of Convergence. Approximate Number of Iterations Necessary to Converge Structures to Gradients of $<10^{-2}$ kJ/Å.

Method	25 Atoms	50 Atoms	100 Atoms	200 Atoms
SOR BDNR	40	100	500	2000
FMNR	3	4	5	6
PRCG	100	175	300	1000
OSVM	75	150	300	600

Speed of Iterations. Approximate Speed of Iterations on a VAX 11/780 in CPU Seconds/Iteration.

Method	25 Atoms	50 Atoms	100 Atoms	200 Atoms
SOR BDNR	0.5	1.5	4	10
FMNR	3	10	90	300
PRCG	0.8	2.5	5	12
OSVM	1	3	10	38

Table II provides an estimate of the rate at which the various MacroModel energy minimization algorithms converge various substituted polycyclic molecules to rms gradients of 10^{-2} kJ/Å using the MM2 force field. Most of the minimization time is spent in reducing the gradient from ca. 1 kJ/Å to the final value. Thus, less stringent convergence criteria allow the optimizations to finish more quickly than is suggested by Table II. The number of iterations shown are typical but can show wide variations depending upon the size and flexibility of the structure. Acyclic molecules in particular require more iterations for convergence than the estimates below suggest. For such flexible systems that require substantial cooperative atom movement, FMNR and OSVM are generally more effective than PRCG (which is still more effective than SOR BDNR).

While MacroModel handles molecular systems larger than 200 atoms, the time required for energy minimization convergence of such molecules is not only lengthy but highly structure-dependent. For such molecules, we use our batch mode modeling BATCHMIN using PRCG, OSVM, BDNR, or TNCG methods. The most effective method we have found for large, flexible molecules is the truncated Newton method (TNCG) recently described by Ponder and Richards.[24]

The standard method for minimizing the energy of a very large structure is to conduct a substructure minimization. In MacroModel, such energy calculations are facilitated by the use of two structural representations in the form of connection tables that we call the main and the substructure connection tables. The main connection table is used for graphics and structural manipulations. The substructure connection table is used for energy calculations. The substructure table is loaded from the main table using a set of atoms which define the substructure for energetic treatment. For normal, full structure calculations, the set of all atoms is used for filling the substructure connection table. For substructure calculations, the assignments of peripheral atoms and positional restraints are generated automatically and stored in a disk file for subsequent substructure runs. We save and use a constant set of peripheral atoms restraints if energies from different runs are to be compared.

The heart of the MacroModel energetics module is a set of subroutines that generate and fill the interaction array and a set of subroutines that calculate the derivatives of the molecular mechanics energy with respect to various atomic coordinates. These routines are the primary determinants of the speed with which MacroModel conducts minimizations and dynamic simulations. Operation of these modules is as follows. When the user initiates an energy calculation, MacroModel generates a large data structure known as the interaction array. The interaction array is an explicit list of all interactions contributing to the molecular mechanics energy, with associated parameters taken from the force field. These interactions include stretch, bend and torsional interactions, and a complete nonbonded pairlist which is periodically regenerated (when any atom moves >0.5 Å or every 100 iterations) during energy minimization or molecular dynamics. A key advantage of an interaction array is that it allows the program to avoid certain recomputations and thus speeds energy and derivative evaluation. The pairlist, while lengthy, is particularly effective because it allows evaluation of only the energetically contributing proximate atom pairs and allows the storage of precomputed components of the nonbonded energies. Such precomputed components include, for example, a single value for bulk of the Coulombs law expression kQ_iQ_j/ε, which is a constant and needs only be divided by the distance between atoms i and j to give the corresponding electrostatic energy. For structures having >100 atoms, such a pairlist approximately doubles energetic calculation speed and the advantage increases with increasing numbers of atoms.

Generation of the interaction array is fast, but initially filling it with the correct parameters from the force field is somewhat time-consuming. To speed the matching of the interactions with the appropriate entries from the force field, MacroModel makes use of *set* processing. The scheme is best illustrated with an example. Assume that a particular bend interaction consists of the atom types C(*sp*3)-C(*sp*2)-O(*sp*3), which corresponds to MacroModel atom types C3-C2-O3. To find all entries in the force field which might match this array of atoms, the program generates the *set* of force field bend entries using rapid bitwise Boolean logic:

(Bend entries with C3 as the first atom .AND.
Bend entries with C2 as the second atom .AND.
Bend entries with O3 as the third atom)
.OR.
(Bend entries with C3 as the third atom .AND.
Bend entries with C2 as the second atom .AND.
Bend entries with O3 as the first atom)

The sets used take the form of bit strings in which a 1 in the N-th position in a string signifies that entry N is a member of that set. Such prescreening of the stretches, bend and torsional entries in the field reduces by 1–2 orders of magnitude the number of entries which must be tested for a complete match.

After the main section parameters are loaded, the program examines the structure for matches from the special substructure section of the force field. The matching of special substructures from the force field with the corresponding fragments of the molecule whose energy is to be calculated is driven by the special substructures. Thus the program considers each special substructure in turn and attempts to map it into any corresponding parts of the molecule. Each occurrence of a match with a part of the molecule is saved as a list of corresponding atoms. These correspondence lists allow replacement of parameters that were previously loaded from the main section of the force field with the special ones associated with special substructures. Partial charges from ab initio calculations are commonly loaded in this way. Conjugated systems such as aromatic heterocycles require modification of stretch, bend, and torsional parameters and are also handled via special substructures. Benzene, for example, is perceived as being composed of butadiene fragments by the main section of the force field which thus loads parameters corresponding to cyclohexatriene. The substructure section of the field, however, subsequently identifies the same set of atoms as being benzenoid and resets the bond length and force constant parameters appropriately. The MacroModel implementation of AMBER and OPLS makes extensive use of special substructures to provide parameters and atomic partial charges for the various amino and nucleic acids.

In energy minimizations and molecular dynamics, most of the time is spent in computing derivatives of the energy with respect to Cartesian coordinates. For large molecules with many interacting pairs of nonbonded atoms, the time can be localized further into the subroutines which compute nonbonded derivatives. To speed this calculation, we have adopted the method of Streett and co-workers,[25] which was devised originally for derivative calculations in molecular dynamics. The method operates on the assumption that the components of the nonbonded derivatives (forces) resulting from the interaction of distant nonbonded atoms change little from iteration to iteration and may thus be treated as constants which are periodically updated. This assumption turns out to be better for energy minimizations than for molecular dynamics because atomic positions change relatively slowly during the bulk of the time spent in an energy minimization. Since the remote atom pairs in large molecules significantly outnumber the proximate ones, the Streett constant derivative scheme is quite effective at speeding derivative calculations. Thus nonbonded derivatives (including the contributions from remote atom pairs) are evaluated as if a very short range nonbonded cutoff distance was being employed. In energy minimizations, we define atom pairs >4.5Å as distant and update the constant, remote components of the nonbonded derivatives only every 25 iterations. The advantage of the Streett nonbonded derivative scheme over continuous evaluation of the complete pairlist increases with molecular size and approximately doubles the minimization speed by the time the size of molecules being minimized reaches 200 atoms.

To conclude this subsection on molecular mechanics, we will briefly summarize several other features of our program which aid in quantitative modeling studies. As noted above, the ability to compute a full second derivative (force constant) matrix allows one to compute vibrational frequencies by the classical normal mode analysis. An energetic saddlepoint (or other maximum) will have at least one imaginary (negative) frequency. To test for such metastable species, MINIMIZ submode provides a button labeled MTest. This button directs the program to evaluate the second derivative matrix, calculate the vibrational frequencies and test the frequencies for negative values. We also use the vibrational frequencies to compute vibrational entropy as part of the FREE E(nergy) submode, which also provides translational and rotational entropies (and partition functions, enthalpy, and heat capacity). While translational and rotational free energies of conformers are similar, the difference in the vibrational entropy of conformers depends on the size and flexibility of the molecule being analyzed. For conformers of very small molecules, vibrational entropies are generally comparable.[26] However, as molecules become larger, entropy differences between conformers can become significant. For example, the rigid rotor/harmonic oscillator entropy differences between the conformers of cycloheptadecane range up to 7 eu (2.1 kcal/mol at 300°K).[27] Thus if even semiquantitative modeling of such molecules is the goal, then free energies rather than molecular mechanics steric energies should be compared. For larger molecules, the anharmonicity of the potential surface also makes a significant contribution to the free energy differences of conformers. For this reason, in MacroModel V2.5 we have added the capability of computing average entropy with inclusion of anharmonic contributions via a quasiharmonic analysis of molecular dynamics simulations.[41]

Molecular Dynamics

MacroModel includes an energetic submode called DYNAMC for conducting molecular dynamics simulations using the velocity Verlet integration algorithm.[28] In MacroModel V2.0, we use these simulations primarily for accumulating average energies, distances, angles, etc., and for conducting local conformational searches. For comparisons of calculated and experimental data, molecular properties averaged over a molecular dynamics simulation are more appropriate than properties taken from energy minimized structures corresponding to 0 K. To provide such information, MacroModel accumulates both averages and histograms of the values of user-selected internal coordinates over the course of a molecular dynamics simulation. The program also provides averaged energies and allows for periodic structure sampling. A dynamic run may also be viewed in real time on the PS300 displays by filling the display memory with sampled structures and replaying the simulation as an animated movie. Such movies are useful for examining simulations for visual evidence of incomplete convergence, viewing low frequency vibrational modes, etc.

We also use molecular dynamics for conformational searches. While contemporary dynamics explores conformational space too slowly to be useful as a global search method, it can be quite effective for searching for new, local minima, which are geometrically similar to the starting structure. Such local conformational searches are accomplished by a molecular dynamics simulation which periodically samples structures for subsequent energy minimization. Files of sampled structures are processed by BATCHMIN in the same way as are the files of structures generated from the MULTIC conformational search described above. MacroModel samples structures in two different ways. One option is to sample a structure at fixed periods in simulation time. Alternatively, the program can save structures only when they are sufficiently different from all previously saved structures so that they might represent a new minima. Structural differences are defined by the residuals from a least squares superimposition[8] of the new structure with ones saved previously. We generally sample a structure when its minimum rms residuals exceed 0.25 Å/atom. The difference method is more effective for conformational searching than is sampling by time only because it saves fewer structures destined to become duplicates upon subsequent energy minimization. In more recent versions of MacroModel, the molecular dynamics module has been extended to provide complete free energy simulations in which average enthalpy, entropy, and free energy is evaluated.

ANALYSIS OF MOLECULAR STRUCTURES—ANALYZ MODE AND SUBMODES

The third main MacroModel mode is ANALYZ. It is used to control display options, examine structures, and measure or compute simple molecular properties. Within this mode are specific submodes for choosing and displaying substructures (SETS, DISPLA), for generating surfaces or other space filling representations (SURFAC, MODEL), for producing molecular volume displays and comparisons (VOLUME) and for computing NMR coupling constants (NMR).

Many of the simple tools for rudimentary geometrical analysis are found in GEOMETry submode. Here one can evaluate internuclear distances and angles by selecting relevant atoms. The submode also contains buttons which allow the user to find atoms or residues by number, or to number and name atoms or residues by graphically selecting them. These features are useful for locating substructures of interest in complex molecules. Other options display potential hydrogen bonds (actual hydrogen bonds depend on the nature of the force field) or distance-defined neighbors of user-selected atoms.

Molecular superimpositions are accomplished in GEOMET submode by overlaying atoms pairs. Algorithms for comparison of the similarity between two or more structures has obvious application to problems in computer-assisted drug design. MacroModel provides two methods for superimposing structures. In both methods, the user selects pairs of atoms to be superimposed to give the best fit by least squares.[8] In the simple method, the molecules are superimposed as rigid bodies. Alternatively, user-defined torsions are allowed to vary during the superimposition process. In this flexible superimposition procedure, a torsion angle driver varies the user-specified dihedral angles and a Simplex procedure[9] is used to optimize the superimposition residuals as a function of the selected torsion angles. This flexible superimposition procedure is rapid when five bonds or fewer are selected for internal rotation.

Atom Set Selection, Manipulation, and Display

Methods for selecting collections of atoms from within complex molecules are found in SETS submode. Sets of atoms are defined either by picking atoms, residues or molecules graphically, or by button selections followed by textual input in response to program prompts. Common atom sets (heteroatoms, charged atoms, unsaturated atoms, terminal atoms, etc.) are available as menu options as are sets of atoms that are defined geometrically (atoms within a certain distance or number of bonds from an initial atom set). Sets of atoms are stored as bits in integer arrays such that a 1 in the *N*th bit of an array describing an atom set indicates that atom *N* is a member of that set.[29] Such sets store atom numbers economically and allow the user to define new sets of atoms using Boolean logic. Given a protein structure, for example, the user can use simple sets and Boolean logic to easily define a new set of atoms which are in lysines and within 10 Å of other cationic residues or atoms. The Boolean operations .AND., .OR., .XOR., and .NOT. rapidly perform bitwise operations on entire computer words. These Boolean operators provide remarkable power in allowing the user to define and store complex sets of atoms. Set processing is used throughout MacroModel for a variety of purposes ranging from force field parameter loading to the selective display of molecular substructures. Sets are also used for certain structural manipulations including the deletion of atoms, substructural reorientations and the setting of atomic positional constraints.

MacroModel uses a range of standard display features to facilitate the manipulation and study of complex molecular structures. Thus relevant substructures of molecules may be readily recolored or displayed using atom selections from SETS submode. For structures having up to several hundred atoms on simple graphics terminals, 3-D geometry is handled acceptably using simple stereopairs which may be displayed and manipulated throughout virtually all MacroModel Modes and submodes. Larger molecules are more easily viewed and manipulated on the Evans & Sutherland displays having local rotation, zooming and hardware depth cuing capabilities.

Molecular Surface Creation and Display

Representation of atomic surfaces and volumes by dots or grids is also performed in ANALYZ mode. In MacroModel Version 2.0, only the fastest and simplest methods for computing van der Waals and solvent accessible surfaces[30] have been implemented. Thus we produce simple molecular surfaces using the method of Langridge[31] to scale and position dot-surfaced spheres followed by elimination of internal dots. Approximations to the solvent accessible surface are made by summing the solvent probe radius and atomic van der Waals radii.[32] These features provide a rudimentary graphical view of the molecular surface as defined by the exposed dot surfaces of intersecting spheres. The dot surface calculations are rapid. We generate the complete surface with removal of internal points and a maximum density of 400 dots/atom at the rate of ca. 0.05 cpu s/atom on a VAX 11/780 computer. The dot surface for each atom requires minimal memory since it is stored internally as a bit set. Other appealing molecular images are provided by the MODEL submode, which produces transparent or colored space-filling models[33] on simple displays or realistic, high quality rendering of Corey–Pauling–Koltun models using the CPK firmware[34] on an Evans & Sutherland PS390 display.

For hard copy output, ANALYZ mode contains a PLOT button which causes the program to create a generic ASCII plot file of pen up/down commands, pen positions, colors and text. This file contains data for plotting the standard line diagrams used to draw structures on the terminal screen. A small external program reads the plot file to produce the plot and is easily adapted to most plotters. Alternatively, the user may produce ouput files for use by the commercial chemical structure plotting program Chem3D.[35]

Volume Mapping

A method[36] for the rapid computation of approximate molecular volumes and their subsequent display is available in the ANALYZ submode.

The method relies on bit representation of individual volume elements and differs from previously described algorithms[37] in the use of bit-encoded templates of van der Waals atomic volumes. A bit array is used to define the volume elements of a box into which the molecules for the volume calculations are embedded. For each atom in the molecules, an image of the appropriate template is offset to a location within the box-defining bit array that corresponds to the atomic position. Since no distance tests need to be performed to determine whether or not a given volume element is interior or exterior to the molecule, the computation is fast. Location of the surface volume elements is straightforward and allows a mesh surface to be constructed for display. Bitwise Boolean operations are utilized for volume comparisons (common volume, excluded volume, etc.). The speed of the method allows interactive computation and comparison of molecular volumes on molecules containing several hundred atoms.

Calculation of NMR Properties

NMR submode is used to evaluate established equations for computing NMR properties of molecules from energy minimized structures. While we envision providing options of prediction of a variety of NMR observables, MacroModel currently allows only the calculation of certain proton-proton coupling constants for which reliable equations have been described in the literature. The selection of the appropriate coupling constant equation and parameters is made from data contained in an external text file we call the "NMR file." This file contains the molecular substructures and parameters which are associated with the various coupling constant equations. It is thus organized like the special substructure section of the force field files and may be edited to add new coupling constant calculations or modify existing ones.

To calculate a coupling constant with MacroModel, the user selectes the Coupl button in NMR submode and then the two atoms involved in the coupling interaction for computation. The program then finds the shortest bonding path linking the atoms and compares it to the substructures found in the NMR data file. Should a match be found, the associated equation and constants are used to calculate and display the coupling constant. Literature references are also contained in the NMR file and are listed in the interactive area of the terminal screen. Our NMR file presently contains various three- to five-bond proton-proton equations for allylic coupling, homoallylic coupling, peptidic NH—CH backbone coupling and the Altona formulation[38] of the generalized Karplus equation.[39] If the user-selected atoms do not match a substructure in the NMR file, no coupling constant is computed and the user is notified accordingly.

USER SUPERMODE

The USER option allows users to incorporate their own special purpose programs and subroutines into MacroModel. It consists of a set of FORTRAN routines which handles graphical menu buttons analogous to the various mode and submode templates used for the modeling operations described above. With these routines, the user may define his own buttons, call existing MacroModel subroutines for structure drawing, atom picking, etc and add his own subroutines to carry out special purpose tasks. The current implementation of USER supermode allows users to create a total of three main modes and 27 submodes. Structures from the main modes of MacroModel are carried transparently into USER modes and submodes by selection of the USER button.

As an example of USER mode programming, we have defined one mode as QUANTM and one of its submodes as AMPAC. From this submode, we can initiate batch mode AMPAC calculations[2] on MacroModel structures and extract coordinate, charge, and energy from the AMPAC output files without leaving MacroModel.

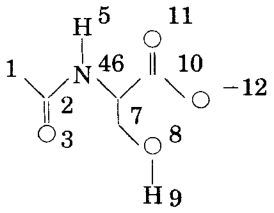
COMPUTATIONAL IMPLEMENTATION OF MACROMODEL AND BATCHMIN

MacroModel consists of approximately 200,000 lines of FORTRAN using many of the VMS extensions. Most of the program is heavily documented with comment statements amounting to approximately 20% of the FORTRAN source. While we have written MacroModel assuming that memory will not be limiting, we have attempted to minimize storage requirements by liberal use of INTEGER*2 and BYTE variables and by declaration of several large arrays into which many temporary arrays are equivalenced. Nevertheless, MacroModel uses approximately 50 Mb of virtual memory. Consequently, the program makes substantial demands on the resources of the host computer and its use on a dedicated work station is probably most appropriate if energy calculations are to be carried out.

With so many complex, interacting features, MacroModel has been organized to facilitate maintenance and trouble shooting. It consists of a large library of subroutines that carry out simple unit operations and a set of higher level modules containing extensive debugging options. Primary among the debugging features is a labeled common block named /DEBUG/, which contains an array of 1000 logical variables which serve as debugging switches. These switches can be turned on individually from the nongraphical TTY mode to activate the debugging options within the selected subroutines. These options generally direct detailed subroutine operation data to output disk files for examination. Other debugging features are available by direct TTY commands. Such features allow the comparison of analytical and numerical derivatives of molecular mechanics energies, listing of the interaction array and testing of structural connection tables for internal consistency.

The batch mode modeling program BATCHMIN is coded differently from MacroModel because it is designed to be used without graphics on a variety of different machines. It is programmed in ANSI FORTRAN 77 with special machine-specific sections included by a preprocessing program which generates the actual FORTRAN source. It presently uses vectorized nonbonded derivative routines and a vectorized double precision energy calculator to speed molecular mechanics and dynamics calculations on vector processing machines. It can accomplish many of the energy-related operations described for MacroModel and carry out Monte Carlo conformational searching as well. Both MacroModel and BATCHMIN share structure and force field files.

BATCHMIN is controlled by a simple command file which provides the input and output structure file names followed by a series of fixed format lines which direct the modeling task. As an example, we show a command file below which reads a file (IN.DAT) of starting geometries for the various conformers of *N*-acetyl serine (numbered as shown in the structural diagram below), minimizes each conformer using the AMBER force field, removes duplicate structures and outputs the unique conformers, ordered by energy, to a file called OUT.DAT.



```
IN.DAT
OUT.DAT
FFLD      3
MULT
COMP      1     2     3     4
COMP      5     6     7     8
COMP      9    10    11    12
ATEQ     11    12
BGIN
READ
MINI      2     0   250
END
```

Line 3 selects the molecular mechanics Force FieLD (3 = AMBER) and line 4 activates the MULTIConformer processing option which checks structures for duplication conformers and sequences output structures by energy. Lines 5–7 give the numbers of the atoms which are to be COMPared by a least squares superimposition check for duplication. In the example, all atoms are being superimposed. The default value for the rms residuals in the superimposition test is 0.25Å. Line 8 indicates that atoms 11 and 12, the two carboxylate oxygens are EQuivalent ATоms. Lines 9–12 then sequentially read all structures in the file, MINImize with block diagonal Newton Raphson (2), without linesearching (0) and allow a maximum of 250 iterations/structure.

CONCLUSION

MacroModel was created to provide graphically controlled molecular modeling with emphasis on quantitative techniques based on molecular mechanics and dynamics. We have made it simple to carry out important modeling tasks, provided rapid graphical feedback whenever possible and also constructed what we believe is the most effective and reliable implementation of molecular mechanics yet devised. Many of the features of MacroModel were created in response to long-standing problems in molecular modeling. In particular, MacroModel provides general and effective solutions to problems of small molecule/substructural conformational searching, of converged energy minimization of large molecules, and of force field parameter assignment and origin/quality documentation.

At the time of this writing, Version 3.0 of MacroModel has been completed and includes new methodology for carrying out semiquanitative modeling of molecules in solution. To retain the ability to carry out energy calculations interactively and conduct extensive conformational searches, we have concentrated on the development of continuum solvation models which are speedy to evaluate and do not have the convergence problems associated with explicit solvent models.[40] For evaluating the anharmonic entropy of solutes at any given temperature, we have implemented the quasiharmonic entropy calculation originally developed by Karplus and Kushick[41] and which we use with constant temperature molecular dynamics. Using these and other methods, we expect to be able to evaluate average free energies in solution almost as rapidly as we can in the gas phase.[42]

APPENDIX A

The EPROX routine calculates the approximate energy of a molecular structure or set of molecules. The force constants used generally approximate the MM2 or AMBER force field energies. Sums of covalent atomic radii are used to compute ideal bond lengths. Bond angles are based on the central atom of the bend interaction and are evaluated as 1,3-distances. For dihedral angles, only one torsion angle per bond is calculated with the torsional constants reflecting the entire barrier. Van der Waals interactions are computed using the Lennard-Jones potential, and a partial charge, distance-dependent dielectric Coulombic potential is used for the electrostatic energy.

For the hypothetical molecule, A—B—C—D—E, the components of the approximate molecular mechanics energy are computed as follows:

Table III. Partial listing of Covalent Radii used in EPROX.

Atom type	Covalent radius (Å)
C1 (Csp)	0.606
C2 (Csp^2)	0.668
C3 (Csp^3)	0.762
CA (united CH)	0.762
CB (united CH_2)	0.762
CC (united CH_3)	0.762
02 (=O)	0.530
03 (—O—)	0.646
H1 (H)	0.352
N2 (Nsp^2)	0.662
N3 (Nsp^3)	0.689

Bond Stretching

The stretching energy is approximated by using a single stretching force constant for all bonds. Comparison of squares of distances avoid computing square roots thus enhancing speed.

$$E_{\text{stretch}} = K_{str} S[l_{\text{A,B}}^2 - (C_{\text{radA}} + C_{\text{radB}})^2]^2$$

where

E_{stretch} = stretching (or compression) energy.
K_{str} = approximated by using ~1/2 the MM2 C(sp^3) − C(sp^3) stretching constant for all bonds (100.0 kJ/Å-mol)
$l_{\text{A,B}}$ = the A-B distance
$C_{\text{rad}N}$ = the covalent radius for atom N (Table III)

Bond Angle Bending

The bending energy is approximated by using a single 1,3-stretching term. The ideal 1,3-distance is computed from the optimal bond angle (Table IV) and two 1,2-distances for the atoms involved in the bending interaction by application of the cosine rule. The 1,2-distances are, in turn, computed from the covalent radii of the atoms involved (Table III). The ideal 1,3-distance thus computed is a constant which is independent of geometry. As with stretching interactions, squares of distances are used. For the A-B-C bending energy:

$$E_{\text{bend}} = K_{\text{bnd}} S[l_{\text{A,C}}^2 - (C_{\text{radA}} + C_{\text{radB}})^2 + (C_{\text{radB}} + C_{\text{radC}})^2 - 2(C_{\text{radA}} + C_{\text{radB}}) \times (C_{\text{radB}} + C_{\text{radC}}) \cos Q_{\text{ABC}}]^2$$

where

E_{bend} = bending energy
K_{bnd} = 1,3-stretching constant for all bond angles which is set to approximate MM2 C(sp^3)-CH_2C(sp^3) bending energies (5.0 kJ/Å-mol for single bond angles).
$l_{\text{A,C}}$ = the A-C distance
$C_{\text{rad}N}$ = the covalent radius of atom N (Table III)
Q_{ABC} = the optimal bond angle for ABC (Table IV)

Torsion

The torsional energy is computed by using a standard expression; however, only $V2$ and $V3$ terms are employed (V2 and V3 terms are shown in Table V). Only a single torsion angle is evaluated for each rotatable bond. For the ABCD torsion:

$$E_{\text{torsion}} = \sum 0.5\, V_2(1 - \cos(2\,\omega)) + 0.5\, V_3(1 + \cos(3\,\omega))$$

where

E_{torsion} = torsional energy
ω = ABCD torsion angle
V_N = N-fold torsional potential barrier (Table V)

Table IV. Partial listing of Atom Type Based Optimal Bond Angles used by EPROX.

Atom type	Optimal bond angle (degrees)
X^a—C2 ^{b} X	120.0
X —C3—X	111.0
X —CA—X	111.0
X —CB—X	111.0
X —O3—X	107.0
X —N2 ^{b} X	120.0
X —N3—X	108.0

[a] X any atom, — single bond, = double bond.
[b] Any bond.

Table V. Partial listing of Torsional Constants used by EPROX.

Torsion	V2	V3 (kJ/mol)
X[a,b] C2—C2 [b] X	20.0	0.0
X—C2═C2—X	250.0	0.0
X—C3—C3—X	0.0	10.0
X [b] C2—O3—X	20.0	0.0
X—C3—O3—X	0.0	3.0
X [b] C2—N2—X	20.0	0.0
X—C3—N2—X	0.0	6.0
X—C2—C3—X	0.0	5.0

[a] *X* any atom, — single bond, ═ double bond.
[b] Any bond.

Van der Waals

The van der Waals energy is computed by using the standard Lennard-Jones equation. Variable van der Waals radii are used (see Table VI) but with a uniform ε. A 5Å cutoff distance is employed. For the A-E nonbonded interaction:

$$E_{\text{VDW}} = \sum \varepsilon[R_o{}^{12}/R^{12} - 2(R_o^6/R^6)]$$

Table VI. Partial listing of VDW Radii used in EPROX.

Atom type	VDW radius (Å)
C2	1.85
C3	1.80
CA	1.85
CB	1.93
CC	2.00
02	1.60
03	1.65
H1	1.30
N2	1.75
N3	1.85

where

E_{VDW} = van der Waals energy
ε = 0.10 kJ/mol
R_o = sum of the VDW radii of atoms A,E (Table VI)
R = the A,E distance

Electrostatics

For the electrostatic energy, approximate charges (see Table A5) are computed for each atom and the Coulomb potential with a distance-dependent dielectric constant is employed to calculate the electrostatic energy. For the i-j nonbonded interaction:

$$E_{\text{elec}} = \sum_{i<j} \frac{Q_i Q_j}{R_{ij}{}^2}$$

where

E_{elec} = electrostatic energy
Q_N = charge on atom N (Table A5)
R_{ij} = i-j distance

Table VII. Partial listing of Atomic Charges used in EPROX.[a]

Atom type	Atomic Charges
C2	0.0
C3	0.0
CA	0.0
CB	0.0
CC	0.0
02	−0.45
03	−0.25
H1	0.0
N2	−0.30

[a] Charge opposite to that given is distributed equally over all attached atoms.

APPENDIX B

```
    Amber Force Field
     P. Kollman, JACS, 106, 765 (1984); J. Comp. Chem., 7, 230 (1986)
C
C                              Includes All Atom and United Atom
C                                   Force Field Parameters
C
C  ENERGY FUNCTIONS AND CONVERSION FACTORS IN USE
-1
0   STR       1          8.36800
0   BND       1          8.36800
0   S-B       0          0.00001
0   TOR       1          4.18400
0   IMP       2          4.18400
0   VDW       2          4.18400
0   ELE       2       1389.50000
0   HBD       1       -418.40000
0   V14       3          0.50000
0   HYD       0          0.00000
0   FIX       1          4.18400
0   SEL       1                       O     Original AMBER params
0   SEL       1                       M     Modified params
```

SECTION 1

```
0  SEL        1                         A      Added params
0  SEL        2                         1      Specific, high quality params
0  SEL        2                         2      Tentative values for params
0  SEL        2                         3      Generalized, low quality params
0  ALT        1                         b      United atom field charges
0  ALT        2                         N      Z0 is Sodium
C
C  Field 1 (Origin):  O = Original AMBER, M = Modified Parameter, A = Additional Parmeter
C  Field 2 (Quality): 1 = Final Values, 2 = Tentative Values, 3 = Generalized Value
C   Alternative 001:     a - Atomic charges from the all atom paper (JCC, 230 (86))
C                        b - Atomic charges from the united atom paper (JACS, 765 (84))
C   Alternative 002:     L - Lithium (JACS, 2212 (85))
C                        N - Sodium (JACS, 3258 (82))
C                        K - Potassium (JACS, 3258 (82))
C
C EQUIVALENCED ATOM TYPES
-5
 1 CS        CA   CB   CC
 2 CT        CA   CB   CC   C3
 3 CU        C2   CD
 4 CH        CA   CB   CC   C3   H1
 5 AA        C2   CD   N2   N4   O3   S1

C        Stretching Interactions (STR)                Opt. Descriptor      Parameter Referencing
C
C         Bond Length    Constant    Bond Moment      Atm1    Atm2         Select    Altm
C            (ang)     (mdyn/ang)     (debye)                              1 2 3 4 5 S A   Comment
-2
1  CT - H1   1.0900     331.0000    -0.2000          0000    0000         O 1             C(sp3)-H
1  CU - H1   1.0800     340.0000    -0.7777          0000    0000         O 1             C(sp2)-H
1  CS - CS   1.5260     260.0000                     0000    0000         O 1             C(sp3)-C(sp3), UA
1  C3 - CS   1.5260     285.0000                     0000    0000         O 2             C(sp3)-C(UA,sp3)
1  C2 - CS   1.5220     317.0000     0.8187          O2N2    0000         O 1       1 a   C(sp3)-C(amide), AA charges
1  C2 - CS   1.5220     317.0000     0.1900          O2N2    0000         O 1       1 b   C(sp3)-C(amide), UA charges
1  C2 - CS   1.5220     317.0000     0.8188          O2OM    N000         O 1       1 a   C(sp3)(N)-C(carboxylate) AA chg
1  C2 - CS   1.5220     317.0000     0.2340          O2OM    N000         O 1       1 b   C(sp3)(N)-C(carboxylate) UA chg
1  C2 - CS   1.5220     317.0000     1.8250          O2OM    0000         O 1             C(sp3)-C(carboxylate)
1  C2 - CS   1.5100     317.0000    -0.2500          AAAA    0000         O 1             C(sp3)-C(aromatic)

C      Bending Interactions (BND)                  Opt. Descriptor
C
C                            Angle       Constant      Atm1 Atm2 Atm3
C                            (deg)      (KCal/mol)
-2
2  H1 - CT - H1           109.5000     36.0000        0000 0000 0000        O 1              H-C-H
2  N2 - CT - H1           109.5000     38.0000        0000 0000 0000        O 1              N(sp2)-C-H
2  O3 - CT - H1           109.5000     35.0000        0000 0000 0000        O 1              O-C-H
2  CB - CB - N2           111.2000     80.0000        0000 0000 0000        O 1              CH2-CH2-N(sp2)
2  CC - CA - N2           109.5000     80.0000        0000 0000 0000        O 1              CH3-CH-N(sp2)

C      Torsional Interaction (TOR)                    Opt. Descriptor
C
C                          V1/2     V2/2     V3/2    Atm1 Atm2 Atm3 Atm4
C                          (Constants in Kcal/mol)
-2
4  O2 = C2 - N2 - 00    0.0000  2.2625  0.0000  0000 O300 0000 0000    O 2          O=C(-O)-N-, JACS, 2058 (86)
4  O3 -  C2 - N2 - 00   0.0000  2.2625  0.0000  0000 O200 0000 0000    O 2          O-C(=O)-N-, JACS, 2058 (86)
4  O2 = C2 - N2 - 00    0.0000  2.0000  0.0000  0000 N200 0000 0000    A 2          O=C(-N)-N-, Urea, WCS (CU)
4  N2 -  C2 - N2 - 00   0.0000  2.0000  0.0000  0000 O200 0000 0000    A 2          N-C(=O)-N-, Urea, WCS (CU)
4  O2 = C2 - N2 - H3    0,6500  2.5000  0.0000  0000 0000 0000 0000    O 1          O=C-N-H
4  O2 = C2 - N2 - 00    0.0000  2.5000  0.0000  0000 0000 0000 0000    O 1          O=C-N-X (X.NE.H)
4  O2 = C2 - N2 - CU    0.0000  0.7500  0.0000  0000 0000 CU00 0000    A 3          O=C-N(*Csp2)-C(sp2), WCS (CU)
C
C - NOTE:  The torsional respresentation above differs superficially from that described in AMBER but is
C             in fact mathematically equivalent.

C      Improper Torsional Interaction (IMP)            Opt. Descriptor
C
C                            V1/2     V2/2     V3/2    Atm1 Atm2 Atm3 Atm4
C                            (Constants in Kcal/mol)
-2
5  CA * C2 * CB * N5    0.0000  0.0000  7.0000  0000 O200 0000 0000   O 1          CH(C=O)(CH2)(N+sp2)
5  CA * C2 * CA • N5    0.0000  0.0000  7.0000  0000 O200 0000 0000   O 1          CH(C=O)(CH)(N+sp2
5  CA * C2 * CC • N3    0.0000  0.0000 14.0000  0000 O200 0000 0000   O 1          CH(C=O)(CH3)(Nsp3)
5  N2 • C2 •  00 * 00   0.0000  0.7500  0.0000  0000 N2O2 0000 0000   A 2          Nsp2(C=O,Nsp2), Urea, WCS (CU)
5  N2 * 00 *  00 • 00   0.0000  1.0000  0.0000  0000 0000 0000 0000   O 1          N(sp2)

C      Van der Waals Interactions (VDW)                  Opt. Descriptor
C
C          Radius     Epsilon      Offset      Charge          Atm1 Lp
C          (Ang)     (Kcal/mol)    (Ang)     (Electrons)
-6
C2        1.8500      0.1200                                   0000          O 1
C3        1.8000      0.0600                                   0000          O 1
CA        1.8500      0.0900                                   0000          O 1
CB        1.9250      0.1200                                   0000          O 1
CC        2.0000      0.1500                                   0000          O 1
```

```
                          ...
Z0        1.0000     0.2200                  1.0000     0000        O 2    2 L  Lithium
Z0        1.6000     0.0310                  1.0000     0000        O 2    2 N  Sodium
Z0        2.0000     0.1300                  1.0000     0000        O 2    2 K  Potassium
END OF NONBONDED INTERACTIONS

C     Hydrogen Bond Interactions (HBND) Opt. Descriptor
C
C Accpt/Donor            "C/100"     "D/100"         Atm1 Atm2
C                      (Constants in Kcal/mol)
-2
7  H1   OM            141.8400     30.8200         S100 0000
7  H1   O2            141.8400     30.8200         S100 0000
7  H1   O3            141.8400     30.8200         S100 0000
                         ...

C     Amber Substructure File:  Selected Amino Acids and Bases
-3
C  Imidazole (AMBER histidine)
9  CU-N2-CU=N2-CU=1
-2
1   1    2                         1.3830   424.5000
1   2    3                         1.3430   477.0000
1   3    4                         1.3350   488.0000
1   4    5                         1.3925   412.0000
1   5    1                         1.3730   515.0000
1   1    H1                        1.0800   340.0000    -0.5814
1   2    H3                        1.0100   434.0000    -1.1061
1   3    H1                        1.0800   340.0000    -0.1867
1   5    H1                        1.0800   340.0000    -0.0934
2   1    2    3                  107.3000    70.0000
2   1    2    H3                 126.3000    35.0000
2   3    2    H3                 126.3000    35.0000
2   2    3    4                  111.6000    70.0000
2   3    4    5                  110.5000    70.0000
2   4    5    1                  110.9000    70.0000
2   5    1    2                  105.9000    70.0000
4  00    1    2    00              0.0000     5.6000     0.0000
4  00    2    3    00              0.0000     9.3000     0.0000
4  00    3    4    00              0.0000    10.0000     0.0000
4  00    4    5    00              0.0000     4.8000     0.0000
4  00    5    1    00              0.0000    14.3000     0.0000
-4
8  -0.0700   0.0830   0.2770  -0.5020   0.2130                                O 1    1 a  JCC, 230 (86)
8   0.1220  -0.1240   0.3840  -0.5270   0.1450                                O 1    1 b  JACS, 765 (84)
-3
S  Valine
9  N0-CA(-CA(-CC)-CC)-C2=O2
-2
1   2    H1                        1.0900   331.0000    -0.2512
1   3    H1                        1.0900   331.0000    -0.1256
1   4    H1                        1.0900   331.0000    -0.1623
1   5    H1                        1.0900   331.0000    -0.1623
-4
8   0.0000  -0.0160   0.0120   0.0020   0.0020   0.0000   0.0000              O 1    1 a  JCC, 230 (86)
8   0.0000  -0.0450   0.0330   0.0060   0.0060   0.0000   0.0000              O 1    1 b  JACS, 765 (84)
```

SECTION 3

References

1. MM2 program number 395 available from the Quantum Chemistry Program Exchange at Indiana University.
2. AMPAC program number 506 available from the Quantum Chemistry Program Exchange at Indiana University.
3. N. L. Allinger, *J. Am. Chem. Soc.*, **99,** 8127 (1977). The most recent parameter set for MM2 was kindly provided by Professor N. L. Allinger at the University of Georgia.
4. a. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, *J. Am. Chem. Soc.*, **106,** 765 (1984). b. S. J. Weiner, P. A. Kollman. D. T. Nguyen, and D. A. Case, *J. Comp. Chem.*, **7,** 230 (1986).
5. W. L. Jorgensen and J. Tirado-Rives, *J. Am. Chem. Soc.*, **110,** 1657 (1988).
6. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. J. Tasumi, *J. Mol. Biol.*, **112,** 535 (1977).
7. F. H. Allen, S. Bellard, M. D. Brice, B. A. Cartwright, A. Doubleday, H. Higgs, T. Hummelink, B. G. Hummelink-Peters, O. Kennand, W. D. S. Motherwell, J. R. Rodgers, and D. G. Watson, *Acta Crystallogr., Sect. B,* **B35,** 2331 (1979).
8. W. Kabsch, *Acta Crystallogr., Sect A,* **A32,** 922 (1976); ibid., **A34,** 827 (1978).
9. J. A. Nelder and R. Mead, *Computer J.,* **7,** 308 (1965).
10. N. Pattabiraman, M. Levitt, T. E. Ferrin, and R. Langridge, *J. Comp. Chem.*, **6,** 432 (1985).
11. M. Lipton and W. C. Still, *J. Comp. Chem.*, **9,** 343 (1988).
12. G. Chang, W. C. Guida, and W. C. Still, *J. Am. Chem. Soc.*, **111,** 4379 (1989).
13. U. Burkert and N. L. Allinger, *Molecular Mechanics,* ACS Monograph 177, American Chemical Society, Washington, D.C., 1982.

14. N. L. Allinger and J. C. Tai, *J. Am. Chem. Soc.*, **87,** 2081 (1965); N. L. Allinger, J. C. Tai and T. W. Stuart, *Theor. Chim. Acta,* **8,** 101 (1967).
15. C. Jaime and E. Osawa, *Tetrahedron,* **39,** 2769 (1983).
16. Reference 13, pp 67–72.
17. E. Polak and G. Ribiere, *Revue Francaise Informat. Recherche Operationelle,* **16,** 35 (1969).
18. S. S. Oren and E. Spedicato, *Math. Prog.,* **10,** 70 (1976).
19. D. G. Luenberger, *Linear and Nonlinear Programming,* 2nd ed., Addison-Wesley, Reading MA, 1984.
20. D. H. Faber and C. Altona, *Comput. Chem.,* **1,** 203 (1977); M. W. Thomas, D. Emerson, *J. Mol. Struct.,* **16,** 473 (1973).
21. S. D. Conte and C. de Boor, *Elementary Numerical Analysis,* McGraw-Hill, New York, 1980, pp. 147–168.
22. A. Perry, *Intern. J. Computer Math., Section B,* **6,** 327 (1978).
23. Reference 19, pp. 268–292.
24. J. W. Ponder and F. M. Richards, *J. Comp. Chem.,* **8,** 1016 (1987).
25. W. B. Streett, D. J. Tildesley, and G. Saville, *Molec. Phys.,* **35,** 639 (1978).
26. Reference 13, p. 172.
27. D. Lee, G. Chang, and W. C. Still, unpublished results.
28. L. Verlet, *Phys. Rev.,* **159,** 98 (1967).
29. For an early use of bitwise set manipulations see: E. J. Corey, W. T. Wipke, R. D. Cramer, and J. W. Howe, *J. Am. Chem. Soc.,* **94,** 431 (1972).
30. B. K. Lee and F. M. Richards, *J. Mol. Biol.,* **55,** 379 (1971); F. M. Richards, *Annu. Rev. Biophys. Bioeng.,* **6,** 151 (1977); M. L. Connolly, *Science,* **221,** 709 (1983).
31. P. A. Bash, N. Pattabiraman, C. Huang, T. E. Ferrin, and R. Langridge, *Science,* **222,** 1325 (1983).
32. A. Bondi, *J. Phys. Chem.,* **68,** 441 (1964).
33. G. M. Smith and P. Gund, *J. Chem. Inf. Computer Sci.,* **18,** 207 (1978).
34. Evans and Sutherland CPK firmware Version #A2.V03.
35. Cambridge Scientific Computing, Inc., Cambridge, MA 02139 USA.
36. R. S. Bohacek and W. C. Guida, *J. Mol. Graph.,* in press.
37. M. Marsili, P. Floersheim and A. S. Drieding, *Computers & Chemistry,* **7,** 175 (1983); T. R. Stouch and P. C. J. Jurs, *Chem. Inf. Comput. Sci.,* **26,** 4 (1986).
38. C. A. G. Haasnoot, F. A. A. M. de Leeuw, and C. Altona, *Tetrahedron,* **36,** 2783 (1980).
39. M. Karplus, *J. Chem. Phys.,* **30,** 11 (1959).
40. W. Hasel, T. Hendrickson, and W. C. Still, *Tetrahedron Comput. Method.,* **1,** 103 (1988).
41. M. Karplus and J. N. Kushick, *Macromolecules,* **14,** 325 (1981); A. D. Nola, H. J. C. Berendsen and O. Edholm, *ibid.,* **17,** 2044 (1984).
42. MacroModel was developed with support and assistance from researchers at the following companies: Rohm & Haas; Smith, Kline & French; Merck, Sharp and Dohme; Upjohn; Eastman Kodak; Burroughs Wellcome; Eli Lilly; Ciba-Geigy; Schering Plough; G. D. Searle, and Glaxo.