

feeling of why he or she should buy it.

Comments. In order to test the general concept and "psychology" of the program, we first tried to use ALCHEMY without reading the manual. The use of functions and their mnemonics seems to be natural. We were able to use most of ALCHEMY's functions as they were intended. It can be said that ALCHEMY is a user-friendly package. When the user's action is not in agreement with ALCHEMY's demands, an appropriate message is usually given.

We appreciate the concept that the user can start from atoms and later on generate atom groups and fragments of his or her own in order to effectively apply ALCHEMY to a specific area of research. It should be said, however, that as a consequence of this flexibility electrostatic potential as an additional energy term cannot be included in the energy calculations.

We also appreciate that the time-consuming processes (minimization and space-filling imaging) can be interrupted within ALCHEMY.

The rectangles to be hit with the mouse in order to govern ALCHEMY's commands are too small to allow rapid communication. The areas in Figure 1 are relatively larger on the scheme than they are on the screen.

Orientation of unfilled valence bonds of the last appended atom can be changed interactively by step rotation. It would be nice to extend this interactive real time rotation on any bond.

Applying an energy optimization procedure to a molecular conformation results in a conformation with low energy and small forces acting on atoms. On a few occasions the conjugated gradient method built into ALCHEMY did not succeed in improving the conformation of a molecule, in spite of the fact that the energy of conformation was extremely high and very large forces were acting on atoms. The number of atoms that could be efficiently used in the molecule is not very large; hence, some other optimization method might work better, although not that fast (for example, the Newton-Raphson method). The user would like to have a choice among more methods.

ALCHEMY's editing concept is not consistent throughout the program. If there is more than one molecule contained on the screen, the user entering the Build and Edit mode should pick the molecule he or she wants to edit. All the functions except the one for deleting atoms can be activated only on the chosen

molecule. Atoms can be deleted on any molecule.

We were not able to crash ALCHEMY and could not make the program get lost by using built-in data files, but we did lose the molecule structure. When the user is changing the dihedral angle of the atoms for which the bond angle is set close to zero, atom centers in a molecule are fused into a straight line. Such a molecule could not be put to the coordinate center with the command "Center molecule" or "Center atom" any longer.

If the user saves a molecule with all filled valences as a fragment, it cannot be connected to already existing molecular structures. The user should get an appropriate message but not one saying "Sorry, but the bond types are not of the same".

If atom labels are excluded from the image, this should apply as well for a molecule appended to already existing ones from a disk file.

Conclusion. Due to the fact that intensive handling of chemical structures is an everyday occupation for many chemists, spectroscopists, pharmacologists, and others, ALCHEMY implemented on a PC can be of very good help for many of them. Apart from some minor shortcomings that could easily be corrected in following versions, ALCHEMY is a package that should be seriously considered as a part of accessible software for many laboratories.

ALCHEMY is copy protected. No source code is available. A special license for updates of ALCHEMY would be recommended. The user can return ALCHEMY within a 3-day period if he or she is not satisfied with the program.

Product: ALCHEMY

Price: Tripos Associates, Inc., 6548 Clayton Road, St. Louis, MO 63117 [(314) 647-1099].

Hardware requirements: IBM-PC/XT/AT or compatible; at least 512 kbytes of processor memory; one 1.2-Mbyte floppy disk drive or two 360-kbyte floppy disk drives; serial port; mouse from either Mouse Systems, Inc., or Microsoft Corp., enhanced graphic adaptor (EGA) or compatible with at least 256 kbytes of graphic memory; appropriate EGA monitor.

Hardware recommendations: hard disk drive; floating point coprocessor.

Plotters supported: plotter that uses the HPGL plotter commands as defined by Hewlett-Packard, Inc., with a serial interface connected to a second serial port on PC.

GRAPH X

J. ZUPAN

Boris Kidrič Institute of Chemistry, 61115 Ljubljana, Yugoslavia

Received December 4, 1987

GRAPH X is a library of subroutines (procedures) for generating graphics on screen (0-719, 0-347 pixels) and on dot-matrix printers if the PC is equipped with a Hercules graphics card. The library consists of 15 on-screen programs and a hardcopy program for printing the generated graphic images (both pages, 0 and 1, in normal or reverse video form) on a dot-matrix printers (see Table I). The routines (procedures) can be used in FORTRAN, Pascal, Turbo Pascal, BASICA, Compiled BASIC, and Assembly language. Although C language is not mentioned in the manual, the software supplied to us contained the graphics library for C language as well.

GRAPH X is a well-defined and concise set of routines that programmers (with the exception of Turbo Pascal users) of

PC's are waiting for. I like the easiness of implementation and linking with already compiled programs as well as the possibility to include the graphics library into the existing high level language libraries (FORTRAN, Pascal, etc.). The manual is clearly written and easy to follow. There are few typographical errors in the programming guidelines, and some of them are already corrected on the supplied floppy disk. The hardcopy routine can be activated from the program or executed on the system prompt.

The assortment of routines is parsimonious, although it must be admitted that the existing set is sufficient to program far more than meets the eye. However, the lack of any "comfort" keeps the programmer very busy. Even only a slightly complicated task (as, for example, deleting a line) must be pro-

Table I. GRAPH X Functions^a

name of function	no. of arguments	action
1 ARC	4	draws quarter of a circle with radius R at the position X, Y in a given quadrant
2 BLKFIL	4	draws solid rectangle at a specified position
3 CIRC	3	draws circle at the defined position
4 CLRSCR (2-5)	0	clears page buffer
CLRSCH (1)		
5 DISP	1	displays given buffer page (0 or 1)
6 DLINE	2	draws relative line to new position
TEXT (2, 5)	2	fills area of the convex polygon with the reverse color
8 GETPT	3	returns status of the addressed position
9 GMODE	0	puts HGC into graphics mode
10 GPAGE	1	defines buffer page to be written into
11 LEVEL	1	sets color value (black, bright, XOR)
12 MOVE (1, 2, 4, 5)	2	moves cursor to specified position
PUTPT (3)	2	
13 PLOT	2	performs LEVEL operation on specified position
14 HTEXT (1)	3	
TEXT (2, 5)	3	writes string of text beginning with specified position
TEXT B (3)	3	
TEXTF (4)	4	
15 TMODE	0	puts HGC into normal text mode
16 HARDCPY	1	generates output to printer from program

^aNumbers in parentheses mark the language (1, Turbo Pascal; 2, Pascal; 3, Basic and BASICA; 4, FORTRAN; 5, Assembler) in which the formation can be used. If no number is given, the function can be used in all five languages in the same format.

grammed in a number of steps using different graphics routines. Especially annoying is that passing values directly to graphics routines is not allowed. Turbo Pascal is an exception from this rule, but the users of Borland Turbo Pascal will probably continue to use a much more comfortable set of graphics routines provided in Borland's Turbo Graphix Toolbox. Typing numerous assignment statements to specify

each parameter for every employed graphics function is more than a majority of potential users are willing to accept.

Obviously, the package in its present form is intended for application programmers who will build their own routines to carry out more complex tasks, rather than for general users who would like to have a variety of different functions immediately at hand. The general user needs functions to draw arbitrary arcs and ellipses, pie charts, and rectangles; functions for saving, downloading, moving, and recalling windows; functions for zooming, scaling, rotation, and mirroring of chosen areas; the choice of different line types and widths (solid, dots, dashes, blank, etc.); the ability to write characters in different directions; but, above all, the graphics programmer needs good communication during the drawing.

A good manual should give more examples for different cases. There are always users not familiar with graphics but willing to try it. Nowhere in the manual is a description of how to implement a simple dialogue (asking a question, retrieving the answer, and continuing with the drawing on the same page) between the user and the program found. We tried a few standard ways, like switching the pages or the graphics mode off and on to catch the typed characters and display them back at the desired position, but unfortunately with no success.

All the shortcomings notwithstanding, I would like to welcome the GRAPH X, and I am looking forward to the announced new version. I hope that this package is (like Borland Turbo Graphix Toolbox) the beginning of a new concept in which all high languages (Pascal, FORTRAN, C, BASIC, etc.) employ standardized calls for graphics and screen manipulation routines and that the graphics hardware differences will be resolved with the installation of the specific compiler on the PC.

It is encouraging that Hercules Computer Technology, primarily a producer of hardware, recognized this trend before the majority of software houses have. Again, in order to implement good and informative input or display of intermediate or final results, the programmer must have at hand powerful and easy to use graphics because "a picture is worth more than a thousand words (or numbers)"!

WordStar Professional 4[†]

CHARLES F. HAMMER* and ROBERT DE LEVIE

Department of Chemistry, Georgetown University, Washington, D.C. 20057

Received December 4, 1987

The new version 4 of WordStar, now called WordStar Professional (WS4), contains many improvements over its predecessor. Among the most useful of these are an "unerase" instruction that will undo the last deletion, a "shorthand" feature that can generate up to 36 user-defined words or instructions with just two keystrokes, 40 reprogrammable function keys, a thesaurus for help in finding synonyms, direct onscreen display of bold and underlined text, enhanced printer support including the ability to use two printers, better documentation, and higher speed. There is also the convenience of using DOS-style commands in specifying filenames as well as accessing DOS while in WordStar and support for mathematical operations. That is quite a lot, and the above list is far from complete. Also included in the package, as a separate

[†] MicroPro International Corp., 33 San Pablo Ave., San Rafael, CA 94903.

program that can be installed and used independently from WordStar, is a fast and efficient word-counting program, wc.

In general, the new version retains the old instructions with which WordStar users have learned to live. This makes it easy to move up to version 4, as one can slowly explore the newly added instructions and incorporate those that prove beneficial.

In doing so, a few problems with the new version show up. The most important of these, and in our opinion a quite annoying and serious one, is that the Professional version no longer has the ability of earlier versions to make a reasonable guess about the placement of soft hyphens. There are, actually, two changes here. In the first place, the soft hyphen now appears automatically at the location of the right-hand margin, even if this would place it in an absolutely ridiculous position, such as before the last letter of a word or between the last word