Molecular Orbital Calculations on Microcomputers

by G. Scott Owen*

I have been very enthusiastic about the potential cabability of the new generation of microcomputers (8086/8088) for number crunching tasks. This is because of the increased memory space available and because of the existence of the 8087 Numeric Data Processor. I felt that it was possible to perform mainframe type calculations on the IBM PC but until now this has been frustated by the lack of a good FORTRAN compiler.

This problem has now been solved with the arrival of the new MICROSOFT FORTRAN 77 compiler. This compiler supports the 8087 and also supports the large amount of memory available for the IBM PC. With this compiler, each named COMMON block can occupy up to 64 K bytes of space.

We (actually my student, Jannetta Bowden) have implemented the QCPE program CNINDO on the PC. In its current version the program will perform CINDO/INDO calculations on molecules of up to 20 atoms and 50 orbitals. The program requires a minimum of 256 K bytes of RAM. It executes fairly rapidly and the results are identical to our VAX 11/780 implementation. Some representative execution times are given below(these are "wall clock" times, i.e. they include the time for the file input and output and not just CPU time):

MOLECULE	ATOMS	ORBITALS	TIME (min:sec)
Methane	4	8	0:55
Acetone	8	20	6:42
Butane	14	26	8:02
Cyclohexane	18	36	16:30
Ribose	20	50	45:03

All of the above times are with the 8087 NDP (our current version of the FORTRAN libray for use without the 8087 doesn't work - we have a bad copy.) I would suspect that without the 8087 the times would be an order of magnitude larger. We are currently working on a version that would allow up to 40 atoms and 90 orbitals. This will probably require about 384 K bytes of RAM and a 90 orbital calculation would take about 4 1/2 hours. I haven't tested it but the program will probably run on many of the IBM PC compatible machines.

If you are interested in this program please send me $20 and I'll send you a disk containing the FORTRAN source program, the compiled Fortran program, documentation and a set of sample input data.

* Department of Chemistry
Atlanta University
Atlanta, GA 30314