QMIC program descriptions
Appendix H. QMIC Programs

The Quantum Mechanics in Chemistry (QMIC) programs whose source and executable ver sions are provided along with the text are designed to be pedagogical in nature; therefore they are not designed with optimization in mind, and could certainly be improved by interested students or instructors. The software is actually a suite of progra ms allowing the student to carry out many different types of *ab initio* calculations. The student can perform Hartree-Fock, MP2, or CI calculations, in a single step or by putting together a series of steps, by running! the programs provided. The software can be found on the world wide web in several locations:

- at the University of Utah, located at: http://www.chem.uta h.edu

- at the Pacific Northwest National Laboratory, located at: http://www.emsl.pnl.gov:2080/people/bionames/nichols_ja.html

- at the Oxford Univ ersity Press, located at: http://www.oup-usa.org

These programs are designed to run in very limited environments (e.g. memory, disk, and CPU power). With the exception of "integral.f" all are written in single precision and use min imal memory (less than 640K) in most instances. The programs are designed for simple systems e.g., only a few atoms (usually less than 8) and small basis sets. They do not use group symmetry, and they use simple Slater det! erminants rather than spin-adapted configuration state functions to perform the CI. The programs were all originally developed and run on an IBM RISC System 6000 using AIX v3.2 and Fortran compilers xlf v2 and v3. All routines compile untouched with gnu compilers and utilities for work stations and PCs. The gnu utilities were obtained from the ftp server: ftp.coast.net in directory: simtel/vendors/gnu. Except for very minor modifications all run untouched when compiled using Language Systems Fortran for the Macintosh. The intrinsics "and", "xor", and "rshift" have to be replace by their counterparts "iand", "ixor", and "ishft". These intrinsic functions are only used in program hamilton.f and their replacement functions are detailed and commented in the hamilton program source. No floating point unit has been turned on in the compilation. Because of this, computations on chemical systems with lots of basis functions performed on an old Mac SE can be tiring (the N^5 processes like the transformation! can take as long as a half hour on these systems). Needless to say all of these run in less than a minute on the fancier workstations. Special thanks goes to Martin Feyereisen (Cray Research) for supplying us with very compact subroutines which evaluate one- and two-electron integrals i n a very simple and straight forward manner. Brief descriptions of each of the programs in QMIC follow:

Current QMIC program limits:

- Maximum number of atoms: 8

- Maximum number of orbitals: 26

- Maximum number of shells: 20

- Ma ximum number of primitives per shell: 7

- Maximum orbital angular momentum: 1

- Maximum number of active orbitals in the CI: 15

! xt• Maximum number of determinants: 350

- Maximum matrix size (row or column): 350

Program INTEGRAL This program is designed to calculate on e- and two-electron AO integrals and to write them out to disk in canonical order (in Dirac <12|12> convention). It is designed to handle only S and P orbitals. With the program limitations described above, INTEGRAL memory usage is 542776 bytes.

Program MOCOEFS This program is designed to read in (from the keyboard) the LCAO-MO coefficient matrix and write it out to disk. Alternatively, you can choose to have a unit matrix (as your initial guess) put out to disk. With the program l imitations described above, MOCOEFS memory usage is 2744 bytes.

Program FNCT_MAT This program is designed to read in a real square matrix, perform a function on it, and return this new array. Possible fun! ctions, using X as the input matrix, are:

(1) $X^{(-1/2)}$, NOTE: X must be real symmetric, and positive definite.

(2) $X^{(+1/2)}$, NOTE: X must be real symmetric, and positive definite.

(3) $X^{(-1)}$, NOTE: X must be real symmetric, and have non-zero ei genvalues.

(4) a power series expansion of a matrix to find the transformation matrix:
$$U = \exp(X) = 1 + X + X^{**}2/2! + X^{**}3/3! + ... + X^{**}N/N!$$

With the program limitations described above, FNCT_MAT memory usage is 1960040 bytes.

Program FOCK This program is designed to read in the LCAO-MO coefficient matrix, the one- and two-electron AO integrals and to form a closed shell Fock matrix (i.e., a Fock matrix for species with all doubly occupied or bitals). With the program limitations described above, FOCK memory usage is 255256 bytes.

Program UTMATU This program is designed to read in a real matrix, A, a real transformation matrix, B, perform the ! transformation: X = B(transpose) *A * B, and output the result. With the program limitations described above, UTMATU memory usage is 1960040 bytes.

Program DIAG This program is designed to read in a real symmetric matrix (but as a square matrix on disk), diagonalize it, and return all eigenvalues and corresponding eigenvectors. With the program limitations described above, DIAG memory usage is 738540 bytes.

Program MATXMAT This program is designed to read in two real matrices; A and B, and to mul tiply them together: AB = A * B, and output the result. With the program limitations described above, MATXMAT memory usage is 1470040 bytes.

Program FENERGY This program is designed to read in the LCAO-MO coefficient matrix, the one- a nd two-electron AO integrals (in Dirac <12|12> convention), and the Fock orbital energies. Upon transformation of the one- and two-electron integrals from the AO to the MO basis, the closed shell Hartree - Fock energy is c! alculated in two ways. First, theenergy is calculated with the MO integrals,

Sum(k) 2*<k|h|k> + Sum(k,l) (2*<k,l|k,l> - <k,l|l,k>) + ZuZv/Ruv.

Secondly, the energy is calculated with the Fock orbital energies and one electron energies in the MO basis,

Sum(k) (eps(k) + <k|h|k>) + ZuZv/Ruv.

With the program limitations described above, FENERGY memory usage is 1905060 bytes..

Program TRANS This program is designed to read in the LCAO-MO coefficient matrix, the one- and two-elect ron AO integrals (in Dirac <12|12> convention), and to transform the integrals from the AO to the MO basis, and write these MO integrals to a file. With the program limitations described above, TRANS memory usage is 1905060 bytes.

Progra m SCF This program is designed to read in the LCAO-MO coefficient matrix (or generate one), the one- and two-electron AO integrals and form a closed shell Fock matrix (i.e., a Fock matrix for species with all doubly ! occupied orbitals). It then solves the Fock equations; iterating until convergence to six significant figures in the energy expression. A modified damping algorithm is used to insure convergence. With the program limitations described above, SCF memory usage is 259780 bytes.
r Program MP2 This program is designed to read in the transformed one- and two-electron integrals and the Fock orbital energies after which it will compute the second order Moller Plesset perturbation theory energy (MP2). With the program limitat ions described above, MP2 memory usage is 250056 bytes.

Program HAMILTON This program is designed to generate or read in a list of determinants. You can generate determinants for a CAS (Complete Active Space) of orbitals or you can inp ut your own list of determinants. Next, if you wish, you may read in the one- and two-electron MO integrals and form a Hamiltonian matrix over the determinants. Finally, if you so choose, you may diagonalize the Hamiltoni! an matrix constructed over the determinants generated. With the program limitations described above, HAMILTON memory usage is 988784 bytes.

Program RW_INTS This program is designed to read the one- and two- electron AO integrals (in Dirac <12|12> convention) from use r input and put them out to disk in canonical order. There are no memory limitations associated with program RW_INTS.

QMLIB This is a library of subroutines and functions which are used by the QMIC programs.

"limits.h" This is an include file containing ALL the parameters which determine memory requirements for the QMIC programs.

Makefile There are a few versions of Makefiles available: a generic Makefile (Makefile.gnu) which works with Gnu make on a unix box, a Makefile (Makefile.486) which was used to make the programs on a 486 PC using other Gnu utilities like "f2c", "gcc", etc. and a Makefile (Makefile.mac) which was used on the Macintosh.

BasisL! ib This is a library file whichcontains gaussian atomic orbital basis sets for Hydrogen - Neon. The basis sets available to choose from are:

1.) STO3G by Hehre, Stewart, and Pople, JCP, 51, 2657 (1969).
2.) 3-21G by Brinkley, Pople, and Hehre, JACS, 102, 939 (1980).
3.) [3s2p] by Dunning and Hay in: Modern Theoretical Chemistry Vol 3, Henry F.
       Schaefer III, Ed., 1977, Plenum Press, NY.

The QMIC software is broken up into the following folders (directories):

```
|------- Doc (potential contributed teaching material)
          |
          |         Source
          |        /
          |       /
        | /
        | /
| /
QMIC --- Examples
        | \
        | \      RS6000
        |    \         /
        |     \ !/
|       \      /
        |      Execs ------ Mac
        |      |      \
        |      |       \
par     |      |         PC486
        |      |
        |      |
        |      |
        |      |
        |      |--- Other platforms as
        |      requested and available
        |
        |
        |------Readme.1st, Readme.2nd
```

Source - This folder (directory) contains all FORTRAN source code, include files, Makefiles, and the master copy of the basis set library.

Execs - This folder (directory) contains all the executables as well as the basis set library file accessed by the "integral" executa! ble (BasisLib). The executables are stored as a self-extracting archive file. The executables require about 1.3 Mbytes and cannot be held once extracted on a floppy disk (therefore copy the files to a "hard drive" before extracting ...).

Examples - This folder (directory) c ontains input and associated output examples.