DCQTP User Manual

Large Scale Semi-Emprical Calculations
Last updated February, 2007

Seth Hayik

Table of Contents

Reference Information

Should you publish data generated using DivCon, please include the following references:

- B. Wang, K. Raha, N. Liao, M.B. Peters, H. Kim, L.M. Westerhoff, A. M. Wollacott, A. van der Vaart, V. Gogonea, D. Suarez, S.L. Dixon, J.J. Vincent, E.N. Brothers and K.M. Merz Jr., DivCon
- "Semiempirical Molecular Orbital Calculations with Linear System Size Scaling" S.L. Dixon, K.M. Merz Jr., J. Chem. Phys. 1996, **104**p: 6643-6649.
- "Fast, Accurate Semiempirical Molecular Orbital Calculations for Macromolecules." S.L. Dixon, K.M. Merz Jr., J. Chem. Phys. 1997, **107**p: 879-893.

When using the solvation module, add the following reference:

- "Fully Quantum Mechanical Description of Proteins in Solution. Combining Linear Scaling Quantum Mechanical Methodologies with the Poisson-Boltzmann Equation." V. Gogonea, K.M. Merz Jr., J. Phys. Chem. A. 1999, **103**p:5171-5188.

When executing a pairwise energy decomposition calculation using the PWD module, add the following references:

- A. van der Vaart, K.M. Merz Jr., J. Phys. Chem. A. 1999, **103**p: 3321-3329.
- K. Raha, A. van der Vaart, K. E. Riley, M. B. Peters, L. M. Westerhoff, H. Kim, and K.M. Merz Jr., "Pairwise Decomposition of Residue Interaction Energies Using Semiempirical Quantum Mechanical Methods in Studies of Protein-Ligand Interaction." J. Am. Chem. Soc. 2005, **127**(18)p: 6583-6594

When using the NMR module, you should add the following references:

- B. Wang, E.N. Brothers, A. Vaart, and K.M. Merz Jr., J. Chem. Phys. 2004, **2004**p: 11392-11400.
- B. Wang, K. Raha, K.M. Merz Jr., J. Am. Chem. Soc., 2004 **126**p: 11430-11431.

1 Introduction

DivCon is a linear scaling semi-empirical program for calculation of energies, charges and geometries of systems up to ~20,000 atoms. Available features include:

- Linear scaling Divide and Conquer (D&C) calculations [1-3].
- Cubic scaling standard calculations [4-6].
- Single point AM1[5], PM3[6], or MNDO[4] calculations.
- Geometry Optimization(steepest decent, conjugate gradient, BFGS, and LBFGS available)
- Implicit solvation and Self Consistent Reaction Field(SCRF) calculations[7]
- Mulliken,[8] CM1[9] and CM2[10] charge analysis
- Particle Mesh Ewald[11] for long range electrostatics
- Divide and Conquer interaction energy decomposition(PWD)[12]
- Nuclear Magnetic Resonance prediction and simulation
- Parallel processing using MPI[13]

The program was mainly developed by Steve Dixon. His work includes the development of the semiempirical Divide and Conquer algorithm, implementation of the D&C and standard energy and gradient calculations, geometry optimization routines, Mulliken charge analysis, cluster based subsetting strategy and front end of the program. Arjan van der Vaart added the Monte Carlo routines (single and multi processing), Particle Mesh Ewald routines, grid based subsetting routines, extension of the cluster based subsetting schemes, CM1 and CM2 charge analysis, density matrix build routines, density of state analysis, frozen density matrix routines the interaction energy decomposition routines (serial and parallel), and Talman's algorithm. Valentin Gogonea added the SCRF routines. Jim Vincent parallellized the single point energy and geometry optimization routines, the transition state routines and the sodium parameters. Ed Brothers added dipole and ionization potential routines, the parametrization routines and the sodium parameters. Dimas Suárez added the LBFGS optimization routines, the transition state routines and the frequency calculation routines. Ning Liao has added support for a native Poisson-Boltzmannn(PB) implementation, and Andrew Wollocott has added support for restrained minimization. Subsequently, Hwanho Kim and Lance Westerhoff of QuantumBio Inc. full audited, optimized, and modernized much of the source code in order to impart increased stability and extensibility upon the application. QuantumBio continues to develop DivCon with these same principles in mind.

2 Installation of DivCon

The Binary Distribution requires some knowledge of the Linux/Unix environment. The following steps will show you the types of commands you will need to execute in order to properly install DivCon.

The distribution is founed with a tar.gz file. This file must be uncompressed in order to install the application. To begin, put the tar.gz file in a place where you like it. Anywhere will do, but a good place for it may be /usr/local. Once you put the file in /usr/local you will need to type the following:

% tar xzvf divcon-<platform>-<version>.tar.gz

Next, cd into the directory in order to view its contents. You should familiarize yourself with the directories found within the distribution.

% cd /usr/local/divcon-<platform>-<version>

DivCon requires a number of environment variables. For this purpose, we have provided chemix.csh and chemix.sh files to set these variables. You will therefore need to edit this file to set CHEMIXDIR to the correct path (ie. /usr/local/divcon-<platform>-<version>). We have included the directions for both tcsh/csh and bash.

If tcsh/csh:

1. Edit chemix.csh and change CHEMIXDIR to the directory you uncompressed divcon into.
2.
   ```
   Add the following line to your ~/.cshrc file:
     source /usr/local/divcon-<platform>-<version>/etc/chemix.csh
   ```
3. Relogin

If bash:

1.
   ```
   Following directions for csh but put
     source /usr/local/divcon-<platform>-<version>/etc/chemix.sh
   in your ~/.bashrc file.
   ```
2. Relogin

The set of test cases found in this directory should provide the user with a sample of the inputs required by DivCon. Please enter each directory and run through the tests in order to familiarize yourself with the workings of DivCon.

% cd DivCon_Tests

In order to finally run DivCon, you will need a license file. This file will ....

3 Keywords

3.1 Hamiltonians

AM1 AM1 Hamiltonian to be used.[5]

PM3 PM3 Hamiltonian to be used.[6]

MNDO MNDO Hamiltonian to be used.[4]

MNDO/d MNDO/d Hamiltonian to be used.

PRTPAR print the AM1/PM3/MNDO parameters for all atom types that occur in the input file

NOTE: One Hamiltonian must be selected. There is no default.

3.2 Unrestricted Hartree-Fock

UHF Turns on the unrestricted Hartree-Fock calculations.

IMULT Specifices the multiplicity of a UHF calculation. Must use in conjunction with the UHF keyword See [UHF], page 4.

3.3 Geometry Optimization

DivCon includes various optimization techniques[45] to locate both minima and first- order saddle points (transition structures). On the one hand, the algorithms for locating minima are specially adapted to treat large optimization problems. On the other hand, the methods for optimizing transition structures in DivCon work reasonably well for small and medium size systems.

MAXOPT=INTEGER

do a maximum of INTEGER cycles of optimization. Default: 10 x Number of Geometrical Parameters. (Note: MAXOPT and OPTMAX are interchangeable)

RESTART

read in restart file from previous run. To create a restart file See [DUMP], page 9.

3.4 Convergence Criterion

ETEST=FLOAT

user defined geometry optimization energy change criterion. Default : 0.002 kcal/mol.

GTEST=FLOAT

user defined maximum gradient component criterion. Default : 0.500 kcal/(mol ~A).

XTEST=FLOAT

user defined geometry optimization coordinate change criterion. Default : 0.001 ~A / 0.001 degrees.

3.5 Minimum Optimization

For the semiempirical minimization of large biomolecules, it is advisable to take the starting point from a previously Force-Field minimized structure. Note also that there is no default optimization method.

OPT=STEEP

Geometry optimization proceeds by means of function evaluations along the search direction defined as the minus gradient vector. The steepest descent algorithm is very simple and requires only storage of the gradient. It is well known that this method can oscillate around the minimum path towards the critical point and its rate of convergence slows down as the minimum is approached. STEEP should be used mainly to partially relax a poor starting point during a given number of optimization cycles.

OPT=CONJGRAD

The conjugate gradient method performs quite robust energy minimization along a line that is constructed so that it is "conjugate" to the previous search directions. This avoids the partial "undoing" on every optimization step observed in the steepest descent method. The conjugate gradient method avoids handling of the Hessian and requires only storage for two gradient vectors. Usually, this has been the method of choice for treating very large systems (hundreds and thousand of atoms) by means of classical Force Field or Quantum Mechanical Semiempirical methods.

OPT=BFGS

A fast pseudo Newton-Raphson algorithm, which expands the Energy function to second order around the current point using an approximate Hessian, is also implemented in DivCon. The BFGS (Broyden- Fletcher-Goldfarb-Shanno) formula is the most popular Hessian updating scheme to minimize molecular geometries and ensures the positive definite character of the updated Hessian matrix. However, the method requires the storage, inversion and updating of the full Hessian matrix. This algorithm is practical for only medium size systems and the corresponding BFGS routines are dimensioned to manage only a subset of atoms (see the divcon.dim file in the DivCon distribution for further details).

OPT=LBFGS

The Limited memory BFGS minimizer can be used to minimize large systems efficiently[46]. The LBFGS method stores only diagonal Hessian and small number m of previous steps and gradient vectors. The amount of storage can be controlled by the user (m=5 by default, see also the divcon.dim and lbfgs.F files in the DivCon distribution). This pseudo Newton-Raphson algorithm performs quite well for highly non-linear functions, requiring about 2 times less function and gradient evaluations than the Conjugate Gradient method implemented in DivCon. This method should be used routinely when optimizing large systems.

LSEARCH=FLOAT

LSEARCH is a variable with default value of 1.0, which controls the accuracy of the line search routine in the LBFGS minimizer. Lower values result in a more accurate linear search at the cost of requiring more energy and gradient evaluations. If the linear search fails during a LBFGS minimization run, changing the value of LSEARCH may be a remedy of the problem.

NOLNSR linear search during a LBFGS minimization is not performed. Pure LBFGS steps are taken by the minimizer.

DIAGTERM=FLOAT

scales the initial diagonal form of the inverse Hessian in the LBFGS minimizer. Low values are recommended when optimizing large systems. Default value of 0.0001

DIIS/NODIIS

The DIIS method attempts to find an optimum linear combination of previous geometries[47]. The corresponding linear coefficients are determined by minimizing the norm of an interpolated error vector built as a combination of error vectors. Among the different approaches used to define the DIIS error vectors, DivCon uses the quasi Newton-Step error vector and the energy error vector for OPT=LBFGS and OPT=BFGS, respectively. For the LBFGS method, NODIIS is the default option (see below) whereas DIIS is the default for the standard BFGS algorithm. Note that the DIIS technique needs to store coordinates and gradients of the last mdiis optimization steps (see the divcon.dim file).

The usual implementation of the DIIS technique for the OPT=BFGS method has been included in DivCon while the DIIS module for the LBFGS method has been especially adapted for treating large systems. In order to increase the efficiency of the LBFGS-DIIS method, the computation of the error vector takes advantage of the LBFGS updating scheme of the inverse Hessian matrix and is only activated at the latter stages of a minimization run. For some systems, the DIIS technique consistently reduces the number of optimization cycles maintaining a favorable CPU rate with respect to the NODIIS LBFGS method. However, the DIIS-LBFGS method is more unstable given that sudden increases of the gradient norm may occur, halting thus the optimization run.

FORCE-IT

Geometry optimization will halt if the energy increases on three successive cycles unless the user overrides it with the keyword FORCE-IT.

ADDMM add MM correction to peptide torsional barrier (on by default).

NOMM do not use MM correction for peptide torsions

ROTATE set the rotation angle for energy barrier

3.6 Transition Structure Optimization

Transition structure (TS) optimizations can be carried out with the Partitioned Rational Function Optimization, the Newton Raphson or the Quasi Newton Algorithm. All of these are common methods which are based on some form of augmented Hessian Newton-Raphson approach and use the Powell update formula. Since these methods require the full Hessian matrix, the corresponding routines, as implemented in DivCon, are dimensioned to manage only a subset of atoms (see the divcon.dim file). Location of a TS demands a quite accurate Hessian which can be obtained from an updating scheme (Powell formula is used by DivCon) and/or the numerical calculation of the second derivative matrix.

OPT=TS Use the Partitioned Rational Function optimized (See [OPT=TSPRFO], page 7.).

OPT=TSNR
The standard Newton Raphson (NR) formula is used to calculate the optimization step within a trust radius of 0.10 ~A. Provided that the initial geometry is close to the TS and the Hessian has exactly one negative eigenvalue, this method should converge to the correct solution.

OPT=TSPRFO
With respect the NR formula, the augmented Hessian methods are designed to generate a search towards a saddle point, even when started in a region where the Hessian has not the correct structure. The P-RFO algorithm employs two shift parameters of the Hessian eigenvalues in order to ensure a proper maximization along the TS mode and minimization along the rest of modes. The norm of P-RFO step is scaled down when its value is greater than 0.10 ~A. Identical to OPT=TS (See [OPT=TS], page 7.).

OPT=TSQNA
The QNA method (also known as the Trust Radius Image Minimization) uses only one shift parameter and restricts the total step length to the trust radius whose optimum value is changed along the optimization.

HESS=CALCFC
Hessian is calculated numerically at the initial point and subsequently updated using the Powell formula (Default option).

HESS=CALCDUMP
Hessian is calculated numerically at the initial point and every DUMP cycles of optimization. (See [DUMP], page 9.)

HESS=CALCALL
Hessian is calculated numerically every optimization step.

HESS=READ
The initial Hessian is read from the HESS parameter in the input file with format READ(INPT,'(5E16.8)') ((HESS(I,J),J=1,I),I=1,3*NATOMS). Default units of the Hessian elements are kcal/(mol ~A2).

NOMODEFOLLOW
disable the use of the mode following technique. To ensure that the TS mode is being followed smoothly from one iteration to the next, the P-RFO and QNA algorithms follow the mode which has the greatest overlap with mode followed on the previous cycle.

3.7 Restrained Atoms

BELLY A subset of the atoms in the system, the belly group, will be allowed to relax their position during optimization while the rest of the atoms will be kept at fixed positions by zeroing the corresponding forces. Currently, the BELLY option requires optimization of both minimum or transition structures using cartesian coordinates (a FREQ calculation can be also subjected to the BELLY option).

The BELLY parameter must be included in the input file in order to specify the BELLY group. Two formats are possible:

```
BELLY
  ATOMS  144-178 310-332
END_BELLY
```

This means that the BELLY group of moving atoms will be constituted from atom 144 to atom 178, and from atom 310 to atom 332. Alternatively, the BELLY group can be selected using residue numbering:

```
BELLY
  RESIDUES 10-13 20
END_BELLY
```

Only residues from 10 to 13 and residue 20 will be allowed to move during minimization.

3.8 Frequency Calculations

FREQ computes force constants and the resulting vibrational frequencies using double numerical differentiation with a step size of 0.01 ~A. This value of the step size represents a good compromise between accuracy and numerical problems. The resultant Hessian matrix in cartesian coordinates is written in the divcon.hss output file with units of kcal/(mol ~A2). Vibrational frequencies are computed by determining the second derivatives of the energy with respect to nuclear coordinates followed by transformation into mass- weighted coordinates. Of course, this is only valid at a critical point. By specifying both FREQ and OPT keywords, the frequencies will be computed after successful completion of the optimization task.

THRMO report thermodynamic quantities from a frequency calculation.

3.9 Output

PRTSUB print subsystem atom lists.

PRTVEC print final eigenvectors. All eigenvectors and eigenvalues will be printed by default. If the input file contains PRTVEC parameters, only some eigenvectors will be printed:

```
PRTVEC
  1-458 all
  558-960 -15.0 -10.0
  45-460 ef 10.0
END_PRTVEC
```

The first line indicates that only the eigenvectors for atoms 1-458 need to be printed. These are all eigenvectors when a standard calculation is performed. For a D&C calculations, these are the eigenstates for subsystems that contain atoms 1-458. The second line indicates that the eigenstates for atoms 558 through 960 will be printed if the associated eigenvalues are between ?15.0 and -10.0 eV. The third line indicates that the eigenvectors of atoms 45-460 will be printed if the associated eigenvalues are within 10 eV of the Fermi energy.

DOS perform a density of state analysis. By default, a DOS analysis will be performed on all eigenvalues for all atoms, with interval of 0.5 eV. Intervals and extend of the DOS analysis can be set by the DOS parameters:

```
DOS
        1-435 0.2
        1015-4452 0.3
END_DOS
```

Here the DOS will be printed for all subsystems that contain atoms 1-435 with interval of 0.2 eV and for all subsystems that contain atoms 1015 through 4452 with interval of 0.3 eV. Note that for a standard calculation the DOS will always extend over all atoms.

DIPOLE calculate the magnitude of themolecular dipole moment using all three charge methods.

IP calculate ionization potential

HOMOLUMO
calculate homo-lumo gap. For a D&C run, the homo-lumo gap of all subsystems will be printed.

PRTCOORDS
print atomic coordinates .

PRTPAR Print the AM1/PM3/MNDO parameters for all atom types that are found in the input file.

SCREEN output vital information to screen. If not included DivCon will run silently and only reture access to the user once the job is complete.

WRTPDB write final coordinates of an optimization in a “standard” pdb format.

DUMP=INT
write restart file (divcon.rst) every INT cycles.

PDUMP=INT
write density matrix file (divcon.dmx) every INT SCF iteration

SNAPGEOM
Write coordinates during energy optimization (divcon_snapshot.N) at every N-th optimization step. This can be useful when optimizing very large systems.

TRAJECTORY
dump coordinates to trajectory file (divcon.trj) at restart points.

GEOCALC

Calculates geometric parameters. Input takes the form (after the END_COORD line):

```
GEO
     DISTANCE
          1-2
     END_DISTANCE
     ANGLE
          1-2-3
      END_ANGLE
      DIHEDRAL
           1-2-4-3
       END_DIHEDRAL
END_GEO
```

Note that if an equals sign is included after the atom numbers (i.e. 1-2=2.0) then a set of differences between the calculated values and these numbers are returned.

ERROR Calculates the difference between accepted and calculated values. An example list is shown below, with each component being explained afterward. Note this is only usable with standard calculations, and this list follows the END_COORDS line.

```
ERROR
     HEAT=FLOAT
     IP=FLOAT
     DIPOLE=FLOAT
     ASSOCIATION=FLOAT
           FILExINTEGER
           FILExINTEGER
     END_ASSOC
     ETOTDIFF=FLOAT
           FILExINTEGER
           FILExINTEGER
      END_ETOTDIFF
END_ERROR
```

HEAT is the heat of formation in kcal/mol. IP is ionization potential. DIPOLE is the Mulliken dipole. ASSOCIATION is the energy of association, and the lines following it are the files to be used to calculate it. For instance, if the association energy of a methanol-2 water complex was to calculated, and methanol was in divcon001.in and water was in divcon002.in, the values on the two subsequent line would be 1x1 and 2x2. ETOTDIFF is the total energy difference, and it's files are designated the same way. Note also that a geometry list can be placed inside the ERROR/END_ERROR delimiters using the format given above.

ZMAKE output a z-matrix using the DivCon z-matrix format. Note that this uses the first three atoms as the defining atoms, and thus they may not be collinear.

3.10 General

CARTESIAN

Cartesian coordinate format. DivCon reads cartesian coordinates in the format shown in the following example:

```
 1  N   -0.26120  -0.98976   0.00000
 2  C    0.64694   0.01940   0.00000
 3  C   -0.47100   1.06738   0.00000
 4  C   -1.44202  -0.13945   0.00000
 5  O    1.83331   0.04003   0.00000
 6  H   -0.13870  -1.97802   0.00000
 7  H   -0.49385   1.68899  -0.88436
 8  H   -0.49385   1.68899   0.88436
 9  H   -2.05887  -0.23715  -0.88402
10  H   -2.05887  -0.23715   0.88402
```

Coordinates are in ~A. The specification of symbols and coordinates is format free and the maximum characters per line is 80.

RMIN=FLOAT

The minimum allowed distance between atoms (results in an error for single point calculations and geometry optimizations, configuration will be rejected in an MC-run when a smaller distance is encountered).

ECRIT=FLOAT

set the convergence for the energy in units of eV. (default value is 4x10-6 eV). The actual value of ECRIT will be relaxed if the gradient norm is large and the structure is not tiny. This should speed up convergence without any loss of accuracy

DCRIT=FLOAT

set the convergence for the density matrix in atomic units (default value 5x10-4 e). The actual value of DCRIT will be relaxed if the gradient norm is large and the strucutre is not tiny. This should speed up convergence without any loss of accuracy.

DESCF=FLOAT

related to ECRIT in that it defines the SCF energy convergence criterion. However, unlike ECRIT, this values is considered absolute(in eV). In affect, the SCF calculation will not stop until this criterion is reached.

DPSCF=FLOAT

related to DCRIT in that it defines the SCF energy convergence criterion. However, unlike DCRIT, this values is considered absolute(in eV). In affect, the SCF calculation will not stop until this criterion is reached.

CUTREPUL=FLOAT

set the [xy|xy], [xz|xz], [xx|yy], [xx|zz],[zz|xx], [xx|xx] and [zz|zz] integrals to zero when the interatomic distance is larger than FLOAT. The CUTREPUL keyword can be used to speed up a DivCon simulation by limiting the number of calculations performed.

CUTBOND=FLOAT
cutoff bonding for the H, P and F matrixes beyond FLOAT angstroms. The CUTBOND keyword can be used to speed up a DivCon simulation by limiting the number of calculations performed.

DIRECT causes all 2-electron integrals to be kept in memory and recalculated at each step instead of being written out to file. This is the suggested approad as generally with how fast processor are today and how much memory users have at their disposal, accessing disk may be more expensive.

FULLSCF turns off pseudo diagonalizations and turns on full diagonalizations. This is more expensive than pseudo diagonalizations but may be necessary sometimes.

RESIDUE stores residue pointers within DivCon. Also requires that the user denote the beginning of each residue in the input file by using the “RES” delimiter after the “z” coordinate.

CHKRES check inter-atomic distances for each residue.

TEMPK=FLOAT
user defined divide and conquer temperature. Units are Kelvin, and default is 1000K.

TESTRUN
do setup work and stop before first energy evaluation.

TMAX=FLOAT
user defined maximum CPU time in seconds. The default limit is 20^20 seconds.

GEOM=PDB
allows input file to be PDB formatted data. If the PDB file is provided, the name of the file should be “<BASE_NAME>.pdb”.

PBC periodic boundary conditions in effect. This only works when a box is specified through the BOX parameters:

```
BOX
          XBOX=20.0 YBOX=20.0 ZBOX=20.0

END_BOX
```

which means that the box dimensions are 20x20x20 angstroms. Note that for a single point calculation / geometry optimization where RESGRID (See [RESGRID], page 16.) was not specified all atoms should be contained within the box, for all other cases all center of masses of the residues should be within the box.

SHIFT=FLOAT
user defined initial dynamic level shift[43] parameter in eV.

XYZSPACE
do all operations in xyz space.

MAXIT=INT
Set the maximum number of SCF iterations (default: 100). If it takes more than 100 SCF iterations to converge, it is generally thought that the system will probably not converge and is exhibiting problems.

DOUBLE=INT

perform a double SCF step during a certain number (int) of SCF iteractions. if INT=0, a double SCF will be done for every SCF iteration (only for non-geometry optimizations). This will aid in convergence as it guarantees that the values calculated at each step are based completely on the current step. By default the first step of an SCF calculation will be a double. Please note, using DOUBLE will significantly increase the CPU time requred to execute a DivCon job.

1SCF Perform only the first SCF iteration, i.e. calculate the energy through E = 0.5(H+F)P. Note that this is not equivalent to MAXIT=1, since no diagonalization is performed.

GUESS Build the initial density matrix from one or more density matrix files. These files are listed by means of the GUESS parameters:

```
GUESS
        A.dmx
        b.dmx
END_GUESS
```

Build the initial density matrix from the files A.dmx and b.dmx. The density matrix elements of atoms 1 through a are read from A.dmx, for atoms a+1 through n from file b.dmx.

```
GUESS
        2-10 A.dmx 33-41 f
        20-30 b.dmx 1-11
END_GUESS
```

Density matrix information for atoms 2-10 is read from the density matrix elements of atoms 33-41 of file A.dmx, density matrix information for atoms 20-30 is read from the density matrix elements of atoms 1- 11 from file b.dmx. Missing density matrix elements are auto-initialized and a correction will be applied to constrain the total number of electrons. The density matrix elements of atoms 2-10 will be kept constant during the SCF iterations by using the Frozen Density Matrix approximation.[44] It is imperative that the number of orbitals on a certain atom in the divcon.in file and density matrix file are the same, i.e. the number of orbitals on atom 2 from divcon.in and atom 33 from A.dmx should be identical.

Note that the maximum length of a density matrix file name is 20 characters, no dashes (`"-"`) are allowed in the density matrix file name.

FILES Performs calculations for a set of molecules contained in files named divcon001.in, divcon002.in etc. These files can have the same keywords as the divcon.in file, with the exception that they are restricted to standard calculations (no D&C) and they may not be solvation calculations (no SCRF).

INTGLS=string

The INTGLS keyword can have two values. If the value is “INTGLS=TALMAN” then the Talman method of integrals will be used. If the Value is “INTGLS=STEWART” then the Steward integrals will be used. The default, and recommended value is Talman integrals as this approach has been found to be the most stable and accurate.

PUSH Only in combination with the CLUSTER keyword (page 22) and when multiple cores (i.e. multiple cluster groups) are defined. Push the different cluster groups apart. Default is 106 ~A, the user can define this distance by using PUSH=FLOAT. When more than two cluster groups are defined, each group is place on a gridpoint with gridspacing of 106 ~A or the user defined value.

3.11 Gradient

GRADIENT
output final gradient. (Note: The gradient for MC calculations contains only intermolecular terms. No intramolecular terms are involved.)

CENTRAL
use central difference in gradient calculation.

INTER include only intermolecular contributions for gradient calculation (skip intramolecular contributions). This is default for MC-simulations.

RECIPINTRA
only in effect with PME keywords (See [PME], page 19.). All intramolecular contributions are neglected for the gradient calculation, except for the intramolecular contribution of the Reciprocal energy (this saves computation time). This is the default if INTER is not used.

NOTE: There is no direct equation to calculate the intramolecular part of the reciprocal (PME) energy. As an approximation the total gradient is calculated after which the gradient for each individual residue is subtracted. The gradient for each individual residue is approximated by the gradient of a system with configuration of the first residue in the group and charges equal to the average partial charges of the group. This has a considerable effect on computation time while only introducing a small error. The group parameter is defined as follows:

```
GROUP
                                GROUP 1
                                        1-10 15
                                GROUP 2
                                        11-14
END_GROUP
```

Which means that residues 1-10, 15 are in the same group and residues 11-14 are in the same group.

3.12 Atomic Charges

CHARGE=INT
a net charge is to be placed on system.

MULLIKEN

use Mulliken[8] charges for PME and/or SCRF, write Mulliken charges to charge output file of MC-run.

CM1 use CM1[9] charges for PME and/or SCRF, write CM1 charges to charge output file of MC-run.

CM2 use CM2[10] charges for PME and/or SCRF, write CM2 charges to charge output file of MC- run.

NOTE: For a single point calculation and geometry optimization both Mulliken, CM1 and CM2 charges are calculated, for a MC-run only the one specified.

3.13 Subsetting

This is the basis of divide and conquer methodology.[1-3] Subsetting can be performed by hand through the SUB parameters (page 30), or automatically through the keywords listed below. Subsystems consists of a core surrounded by an inner and outer buffer.[3]

CLUSTER

do cluster based subsetting. Specification of the cluster based subsetting is through the cluster parameters:

```
CLUSTER
    NCORE=3 DBUFF1=4.0 DBUFF2=2.0
END_CLUSTER
```

This means that the cores will be build from 3 residues, the first buffer region extends 4.0 ~A from the core, the second buffer region 2.0 ~A from the first buffer region. Multiple cores (i.e. multiple cluster groups) can be defined by:

```
CLUSTER
       NCORE=2 (1-6 7 8 12-14)
       NCORE=3 (9 10 15-25 )
       NCORE=1 (26 27)
       DBUFF1=4.3 DBUFF2=2.0
END_CLUSTER
```

Cores will be build from 2 residues for residues 1-6, 7, 8, 12-14, from 3 residues for residues 9, 10, 15-25 and from 1 core for residue 26 and 27. The first buffer region is 4.3 ~A, the second 2.0 ~A. Note that all residues should be used (and only once) in this syntax.

```
CLUSTER
        NCORE=1 (1-20) [-1]
        NCORE=1 (21-100) [0]
END_CLUSTER
```

Cores will be build from 1 residue for residues 1-20 and from 1 residue for residues 21-100. Moreover, the charge of the subsystems build from residues 1-20 will be contrained to ?1 electron and the charge of the subsystems build from residues 21-100 to 0 electrons. Only effective when the NO-OVERLAP keyword is used (see page 23). Charges are constrained by use of multiple Fermi energies.[12]

NO-OVERLAP

Only in combination with the CLUSTER keyword.

```
CLUSTER
        NCORE=1 (1-10)
        NCORE=1 (11-20)
END_CLUSTER
```

When the NO-OVERLAP keyword is used, subsystems made from residues 1-10 only overlap with subsystems made from residues 1-10 and subsystems made from residues 11-20 only overlap with subsystems made from residues 11-20. In other words, density matrix elements between subsystems 1-10 and 11-20 are zero.

ATGRID do grid based, atom-wise subsetting (core and buffers will be build from atoms).

RESGRID

do grid based, residue-wise subsetting (core and buffers will be build from residues).

MIXGRID

do grid based, residue-wise subsetting for cores, grid based, atom-wise subsetting for buffers.

NOTE: Specify Grid parameters when a grid based subsetting is selected. The syntax for these parameters is:

```
GRID
   XCORE=4.0 YCORE=4.0 ZCORE=4.0 OVERLAP=0.5
   DBUFF1=2.5 DBUFF2=1.0
END_GRID
```

Meaning that the total system will be divided into rectangular boxes of 4.0_4.0_4.0 ~A that overlap 0.5 ~A in each dimension. The first buffer region is 2.5 ~A wide, the second 1.0 ~A.

NOTE: In Monte Carlo simulations only a residue-wise grid-based subsetting scheme is allowed. Reason for this is rather subtle: Imagine that during the MC-simulation a molecule would penetrate the box, such that the geometric center is still inside the box, but some atoms are outside the box. If an atom based subsetting was performed, the atoms outside the box wouldn't be included in any subsystem. Making the "grid-subsetting"-box artificially larger than the pbc-box wouldn't work either: in that case there's is an artificially larger distance between the molecules and the (virtual, pbc) images of other molecules. This would mean that some atoms will be skipped in making the buffer regions: atoms that, according to their pbc-image should be included. This will lead to non-optimal subsettings and can have a rather drastic effect on energies as was found experimentally.

COMBSUB

do a combination subsetting; certain residues will be subsetted grid based, others cluster based. Use the combsub parameters to specify this subsetting:

```
COMBSUB
      CLUSTER
              1-10 13
      RESGRID
              11-12 14-20
END_COMBSUB
```

Here, cluster based subsetting will be done for residues 1-10 and 13, gridbased residue-wise subsetting will be done for residues 11-12 and 14-20. The cores of the subsystems will be selected from the specified residues, buffers from all residues / atoms of the system. COMBSUB can only be defined as a combination of CLUSTER with one of RESGRID, ATGRID of MIXGRID. Note that you have to specify the CLUSTER and GRID parameters when you use COMBSUB. Note that all residues should be used (and only once) in COMBSUB.

STANDARD

standard closed-shell calculation (no divide and conquer). All subsetting parameters are ignored, only one subsystem containing all atoms will be generated.

3.14 Monte Carlo

DivCon can perform simple Monte Carlo simulations in the NVT and NPT ensemble. It is assumed that each molecule corresponds to one residue. Phase space is only sampled for translational and rotational degrees of freedom.

SMARTMC=INT

do a smart-MC run of INT cycles. This number should be divisible by the number of processors. In a smart MC the translation dr = (_i, _j, _k) is picked from_i = SMARTA_fi,j/kBT + _G with fi,j the jth component of the (intermolecular) force on residue i and _G a random number from a Gaussian distribution with mean 0 and variance 2_SMARTA See [SMARTA], page 17.

MC=INT do a Metropolis-MC run of INT cycles. This number should be divisible by the number of processors. In the Metropolis method the translation dr = (_i, _j, _k) is picked from a uniform random distribution. See [DR], page 17.

DR=FLOAT

set DR for the Metropolis MC method.

SMARTA=FLOAT

set SMARTA for the smart-MC method.

NMOVE=INT

number of residues to move per MC step [nstep = (nres-1)/nmove + 1].

NSUBMOVE=INT

number of subsystems that will be picked for the MC move. Only residues in the cores will be moved, the number of steps now depends on the number of subsystems [nstep ((nsub-1)/nsubmove + 1].

NSUBSTEP=INT

number of steps per MC cycle (not including a boxmove), move the residues in the cores of randomly picked subsystems [nstep = nsubstep].

NOTE: NSUBMOVE and NSUBSTEP may only be used in divide and conquer type calculations, i.e. no STANDARD calculations.

NOTE: NSTEP should be increased with one when an NPT simulation is performed (for the boxmove). An inequality for nstep is used for the NSUBMOVE and NSUBSTEP keywords, because one residue might be in the core of more than one residue.

NPT do a MC simulation in the NPT ensemble.

NVT do a MC simulation in the NVT ensemble.

DBOX=FLOAT

set the maximum boxmove (one dimension) for the MC-NPT simulation.

MCTEMP=FLOAT

set the temperature at which the MC-simulation will be performed (default: 298.15 K).

MCPRES=FLOAT

set the pressure at which the NPT-simulation will be performed (bar, default: 1 bar).

FROZENMC

use a frozen density matrix approach for MC.[44] This means that a full diagonalization is only performed for subsystems in which core residues were moved.

SEED=INT

seed for the random number generator (default: 0).

UPD=INT

update pair list and redo subsetting after UPD MC-cycles.

ADJ=INT

evaluate the ratio between accepted/total steps after AJD cycles, increase DR, DANGLE, DBOX, SMARTA when this ratio is larger than RATIO, decrease them when the ratio is smaller than RATIO.

RATIO=FLOAT

set the preferred ratio of between accepted/total steps (default: 0.5).

DANGLE=FLOAT

set maximum rotation angle (degrees) The maximum rotational angle must be less than 180_.

NWRITE=INT

write coordinate, charge, timing files etc. every NWRITE MC-cycle.

MCECRIT=FLOAT

set the convergence criterion for energy for MC-steps when no coordinate, charge, timing files etc. are written (eV).

MCDCRIT=FLOAT

set the convergence criterion for the density matrix for MC-steps when no coordinate, charge, timing files etc. are written.

3.15 Particle Mesh Ewald (PME)

Long range electrostatics can be handled by the PME method.[11] Hereto, a box has to be specified (See [BOX], page 24.) and periodic boundary conditions are assumed (See [PBC], page 12.). The PME implementation is semi-classical: the long range correction is performed in a classical way using the charges obtained from the quantum calculation. These charges are either Mulliken (See [MULLIKEN], page 14.) , CM1 (See [CM1], page 15.) or CM2 (See [CM2], page 15.) , this needs to be specified by one of the charge keywords (See Section 3.12 [Atomic Charges], page 14.).See the RECIPINTRA (See [RECIPINTRA], page 14.) and the GROUP (See [GROUP], page 24.) parameters for important information concerning the calculation of gradients when PME is used.

BETAPME=FLOAT

Set the beta-parameter for PME.

BETAPME=HBOX

Set the beta-parameter for PME to 3.5/hbox, with hbox half the box dimension of the shortest box edge.

NSPLINE=INT

set the order of the Cardinal B-Spline for the charge grid interpolation. NSPLINE _ [2,KPME] with KPME = MIN(K1PME, K2PME, K3PME) and NSPLINE should be even.

K1PME=INT

Set the grid size for PME in the x-direction.

K2PME=INT

Set the grid size for PME in the y-direction

K3PME=INT

Set the grid size for PME in the z-direction.

K1PME, K2PME and K3PME should be powers of 2 and less or equal to the value of maxpme found in the file divcon.h. Also, the number of grids in any direction should be greater than or equal to the NSPLINE value.

PME Do a PME summation for the long range interaction (not needed to specify when K1PME, K2PME, K3PME is present).

3.16 Self Consistent Reaction Field (SCRF)

There are two different approaches to SCRF calculations supported in DivCon. In the first method, a Delphi license is required as DivCon will call upon this application for the SCRF calculation[7]. In the second method, the SCRF calculation is fully implemented within DivCon. This method does not require a DelPhi license, however both methods do require a proper licesne from QuantumBio Inc.

3.16.1 Delphi

SCRF requests a self-consistent reaction field calculation. This gives the charges in solution and solvation free energy. See the DelPhi manual for more details. The default SCRF method is Delphi.

Files required for SCRF calculations:

pb.prm contains DelPhi parameters (dielectric constant, grid spacing, options for output, etc.)

v3_txt.dat a binary file used to generate the molecular surface (Connolly's surface)

divcon.siz file that contains the van der Waals radii (only used when mentioned in pb.prm)

The SCRF calculation will output the file divcon.pb with information on the SCRF iterations.

SCRF-INSIDE

only fill cavities with a continuum, do not place a continuum around the system. This requires the file solute.in, with the coordinates of the molecule(s) that fits in the cavity. The format of this file is: number of atoms on the first line.

All other lines contain

1. the atomnumber (1 for H, 6 for C etc)
2. the x-coordinate
3. the y-coordinate
4. the z-coordinate
5. atomic Mulliken charge
6. atomic CM1 charge
7. atomic CM2 charge

(in this order). Maximum line length is 80 characters.

DRAD=FLOAT

Only in combination with SCRF-INSIDE. Only include surface elements within DRAD times the vanderWaals radius of the atoms. Suggested value of DRAD: between 1.2 and 1.5 (DRAD should be larger than 1.0), default is 1.5.

PRTVDWAALS

Output van der Waals radii.

3.16.2 DivPB SCRF

DivPB use in addition to SCRF keyword (See [SCRF], page 20.) to use internal beta code instead of Delphi. Note: this code is still being validated and is subject to change in future DivCon releases.

INDI=INT

used to set the internal dielectric constant to the value specified. The default value for this keyword is 1.

EXDI=INT

used to set the external dielectric constant to the value specified. The default values for this keyword is 80.

SCALE=DOUBLE

allows the user to set the grid space (per angstrom). The default value for this keyword is 1.8.

PERFILL=INT

sets percentage of the box that is filled by the system being studied (ie. PERFILL=80 means the protein will take up 80% of the total box), thus controlling the size of the surrounding water box. A smaller value will give more accurate results but take longer to calculate. The default value for this keyword is 60% which is good for small molecules, but the user may want to increase to 80% for large calculations.

ION=DOUBLE

sets the ionic strengh of the solvent surrounding the system of interest. The default value for this keyword is 0 (ie. water).

PROBRAD=DOUBLE

allows the user to set the radius of the probe for the interior surface charges. The default values of this keyword is 1.4.

FINESURF

turns on non-uniform grid (aka Fine Surface) algorithm in DivPB. This algorithm applys the grid density specified by SCALE (See [SCALE], page 21.) only around the solute surface, but cuts the grid density elsewhere to 0.5*SCALE saving time during DivPB calculations. Due to the nature of the algorithm and implementation, non-uniform grid is not suitable for small molecules. The recommended SCALE value when using FINESURF is 2 or more.

3.17 Nuclear Magnetic Resonance(NMR)

This section details the current NMR facilities found in the DivCon application. This functionality is under active research and development, and is only available through a special license.

NMR used to activate NMR functionality. The keyword will cause DivCon to perform an NMR shielding calculation on the atoms denated in the NMR parameters entry at the end of the divcon.in file. In the first example, the shielding calculation will be performed on a number of atoms. In the second example below the calculation will instead be performed on a residue basis.

Example 1:

```
NMR
  Atom 1-100 150-255
END_NMR
```

Example 2:

```
NMR
  Residue 1-10 12-34
```

```
                  Residue 40-45
                END_NMR
```

CALNUC set what atom type chemical shift calculations are done on

CALNUC=1:calculate proton chemical shifts CALNUC=2:calculate carbon-13 chemical shifts

3.18 Parameter Optimization

This section deals with DivCon's ability to optimize semiempirical parameters, which is useful for creating atom types not available currently and for optimization of atom types to reflect certain situations, such as in reaction sites of proteins.[48,49] A caveat is required: the parameter sets developed will have exactly two of each type of gaussian width in the core-core repulsion function, a restriction used in the PM3 parameterization. Note that when either of these are used, the divcon.param file must be present. Note also that each divcon***.in file should contain the error keyword.

GA=INTEGER

Uses a Genetic Algorithm[50] to optimizes the semiempirical parameters. The integer is the number of generations. This writes the restart file creatively named RESTART.GA

POWELL=INTEGER

Uses a Powell optimization routine with a bisection search.[51] The number of cycles is what follows the command. Note that this routine stops before the number of cycles is reached if no further improvement is caused. Note also that this is used to refine the best parameter set of the GA's results.

TESTPOINT

Takes the best parameter set of RESTART.GA and tests it versus a group of molecules specified in divcon.param. This is the way to get full output for files being used in optimization and to see the results of the parameter set in files not used in the optimization.

3.19 Divcon.param

This section deals with the contents of the divcon.param file necessary for parameterization. The following keywords should all be placed on the first line of divcon.param. Following these keywords, there should be an atom list of the following form:

```
ATOMS
       H
       O
       ZN
END_ATOMS
```

FILES=INTEGER

See [FILES], page 13.

SEED=INTEGER

Random number seed. Required for GA.See [SEED], page 18.

AM1 Use the AM1 parameters for all atom types not being optimized. If this is not present, PM3 parameters are used.See [AM1], page 4.

REDO Recalculate the error of all the parameter sets in RESTART.GA. This is necessary when changing the "basis set" of molecules tested against.

3.20 Default Keywords

The keywords in the following section represent keywords on by default in DivCon and the values that they are given when applicable. The sections above should be consulted for more information on the keywords presented below.

ECRIT Default value=4.0e-6. See [ECRIT], page 11.

DCRIT Default value=5.0e-4. See [DCRIT], page 11.

MAXIT Default value=100. See [MAXIT], page 12.

TEMPK Default value=1000.0. See [TEMPK], page 12.

ADDMM On by default. See [ADDMM], page 6.

RMIN Default value=0.5. See [RMIN], page 11.

INTGLS Default setting is TALMAN. See [INTGLS], page 13.

When using SCRF, either Delphi or DivPB the following are the default keywords and their corresponding values:

INDI Default value=1. See [INDI], page 20.

EXDI Default value=80. See [EXDI], page 20.

PERFILL Default value=60.0. See [PERFILL], page 21.

ION Default value=0.0. See [ION], page 21.

4 Parameters

SUB Reads in atom list for handwise subsetting.

```
SUB
        SUB 1
        CORE
                1 2 3
        BUFFER_1
                7 8 9
        BUFFER_2
                4 5 6 10
END_SUB
```

There is one subsystem, the core is build from atoms 1, 2 and 3, the first buffer from atoms 7, 8 and 9, the second buffer from atoms 4, 5, 6, 10. Note that all residues should be named at least once in the cores.

BOX Reads in the box dimensions for a PBC calculation. See [PBC], page 12.

BELLY Reads in belly parameters for geometry optimization. See [BELLY], page 7.

CLUSTER

Reads the parameters for the cluster based subsetting. See [CLUSTER], page 15.

GRID Parameters for gridbased subsetting.

COMBSUB

Parameters for combination subsetting. See [COMBSUB], page 16.

GROUP Specify groups of residues that can be treated identical for calculation of the intermolecular contribution to the gradient See [RECIPINTRA], page 14. and also See [PME], page 19.

DOS Parameters for density of state analysis. See [DOS], page 9.

PRTVEC Parameters for eigenstate analysis. See [PRTVEC], page 8.

GUESS Parameters for density matrix build. See [GUESS], page 13.

NEIGHBORS

Only in combination with CLUSTER subsetting (used with the multiple core and Fermi feature, See [CLUSTER], page 15.) and NO-OVERLAP keyword See [NO-OVERLAP], page 15.

```
NEIGHBORS
        NEIGHBORS=2 1
        NEIGHBORS=1 3
END_NEIGHBORS
```

Specifies the overlap of residues in the first and third cluster groups. Residues in the first cluster group only overlap with residues of the first cluster group if they are within 2 residues away from each other. Residues in the third cluster group only overlap with residues of the third cluster group if they are within 1 residue away from each other. For example:

```
CLUSTER
        NCORE=1 (1-10) [0]
        NCORE=1 (11-20) [0]
        NCORE=1 (21-30) [0]
        DBUFF1=4.0 DBUFF2=2.0
END_CLUSTER
NEIGHBORS
        NEIGHBORS=2 1
        NEIGHBORS=1 3
END_NEIGHBORS
```

Here atoms of residue 1 (which is in the first cluster group) will only overlap with atoms of residue 1, 2 and 3 (that is, if they are within the buffer region). Atoms of residue 5 will only overlap with atoms of residue 3, 4, 5, 6 and 7 and atoms of residue 30 only with atoms of residues 29 and 30.

5 QMScore

QMScore is a quantum mechanics based scoring function for binding affinity calculations. QMScore is a scoring function which, given the structure of protein-ligand complex, can calculate a score relating to its binding affinity.

5.1 QMScore Installation

The correct installation of DivCon will yield the correct installation of QMScore. Please see the DivCon manual for more information. No further installation steps are required.

5.2 System Setup

The starting point for a QMScore calculation is a ligand/small molecule/drug complexed with the protein. However since QMScore is based on quantum mechanics the following requirements are mandatory for its execution and reliability of results.

1. The ligand and protein have to be protonated before QMScore calculation is executed. It is the users responsibility to determine the protonation state of the ionizable groups in the ligand the protein.
2. The total charge on the protein and the ligand should be provided by the user. If metal ions are involved in the protein-ligand interaction then the charge on the metal ion needs to be taken into consideration while assigning the total charge. If the total charge on either is incorrect then DivCon which is the main tool in QMScore will fail to run.
3. QMScore expects the format of the input file to be in the PDB format for both the protein and the ligand. The standard PDB format is assumed for the atoms and the heteroatoms WITHOUT THE CHAIN INFORMATION COLUMN. Hence PDB files that contain thecolumn for the the chain should be pre-processed. If the ligand format is in TRIPOS MOL2 then the application mol2pdb.pl found in $QMSCOREHOME/bin/ can be used to convert it into a PDB format file. It is recommended that before using QMScore the user should see the file formats in the directory $QMSCOREHOME/tests/ Once the above conditions are satisfied QMScore can be executed by the following command:

   ```
   % $QMSCOREHOME/bin/qmscore setup protein_pdb \ protein_charge ligand_pdb
   ligand_charge keywords
   ```

   Once you have executed the setup of the qmscore job, you should run the analysis portion through a similar command line statement:

   ```
   % $QMSCOREHOME/bin/qmscore analyze keywords
   ```

5.3 Keywords

The keywords file in the above command determines different options that QMScore has for thecalculation at the present time which are described as follows.

Hamiltonian

The hamiltonian used in the calculation can be set by this option. Supported Hamiltonians for the QM calculation are AM1 and PM3.

Grid Grid size for the Poisson-Boltzmann Self Consistent Reaction Field (PBSCRF) calculation is set using this option.

Freq Frequency calculation followed by Normal Mode Analysis is set by the option. Setting it to 1 requests a frequency calculation for vibrational entropy for the ligand. A note of caution: Frequency calculations are done after complete geometry optimization of the ligand at the QM level and could be time consuming for large molecules.

PSM This option gives the molecular mechanics interaction energy and the solvation free energy calculated using the Phenomenological Solvation Model between the ligand and the protein. Its is reported as molcular recognition model in the output.

GUESS This option allows the user to use precalculated guess density matrix for the protein in the DivCon calculation. It speeds up the final calcaultion of the complex from the pre-calculated density matrix of the protein. In order to generate a guess density matrix, you should execute DivCon once on the protein alone using the PDUMP=x keyword (See [PDUMP], page 9, for more information).

PROTOP If binding affinity calculation is done only for one protein target and numerous ligands then it suffices to do the protein calculation just once. PROTOP 1 set the conditions for such a calculation where the protein calculation is performed a single time. However for analysis the directory where this calculation is done needs to be set.

PROTDIR

This sets the directory where the protein calculation is performed. The user should provide the complete path to this directory. In the case of a single target protein and numerous ligands it is recommended the protein calculation is performed once by executing QMScore one of the ligands bound to the target. This directory can then be set for other calculations.

IONDEF If metal ions are present in the binding site they need to be defined. Setting IONDEF to 1 tells QMScore about the presence of metal ions. However these should be defined in the ions file in the directory where QMScore is being executed. Two examples of an ions file can be found in the tests directory. Ions are defined in the ions file as follows:

ION 3 2101 2045 4039

This statement defines three ions with their atom numbers. It is recommended that a new user see examples of keyword files in the test directory.

DivPB The Poisson-Boltzmann (PB) code used for the SCRF calculation in the scoring procedure can utilize one of two different implementations. The first will employ a system call to the DelPhi application, while the second approach will use a native, QuantumBio PB implementation. Currently the DelPhi-based SCRF calculation will be the default value while we continue to test our own implementation. To use the internal, DivPB code, simply add the following line to your QMScore keywords file: DivPB 1

Notes:

1. Although DivCon the Linear Scaling Semi-empirical Qunantum Mechanics program is capable of studying large biomolecular systems, in the case of protein-ligand interaction it is advisable to keep the size to the system to less than 3000 atoms for large-scale evaluation of QMScore. This can be achieved by selecting protien residues that are within 10-15 Angstrom from the ligand. Dangling ends of the residues should be capped or protonated. There are numerious applications available to accomplish this goal.
2. Hydrogens that are added to protein and ligand atoms need to be optimized with respect to their geometries before QMScore calculation is performed. The user should utilize a restrain minimization package such as that found in a molecular mechanics application such as AMBER (Sander) to accomplish this goal. Also if the heavy atoms in the complex are determined at low resolution by X-ray crystallography error in the coordinates of heavy atoms could make DivCon fail or make the results unreliable. It is therefore critical to make certain that the quality of the starting structure is good before QMScore is executed.
3. Please see the following references to study QMScore more fully:
    - K. Raha, K. M. Merz Jr., A Quantum Mechanics-Based Scoring Function: Study of Zinc Ion-Mediated Ligand Binding, J. Am. Chem. Soc. 126, 1020-1021 (2004).

    - K. Raha, K. M. Merz Jr., Large Scale Validation of a Quantum Mechanics Based Scoring Function: Predicting the Binding Affinity and the Binding Mode of a Diverse set of Protein-Ligand Complexes, J. Med. Chem. (2005) submitted.

6 Optimizing DivCon

6.1 Speed and Memory

For large systems, use the CUTBOND option to decrease memory use (See [CUTBOND], page 11.). A value larger than or equal to 8 usually gives good accuracy. Use the DIRECT keyword to avoid storage of the integrals (See [DIRECT], page 12.). DIRECT increases the computation time by approx. 15%, but severely decreases memory usage.

6.2 Convergence

Convergence can be sped up by constructing the initial density matrix from guess density matrices (See [GUESS], page 13.). Speedup is only significant when large portions are captured in the guess density matrices and when geometries are very similar. For large systems and highly charged systems, convergence can be sped up by use of multiple Fermi energies. For example, in a protein-water system, first perform a calculation of the water system without protein and the protein without water. Use these two density matrices as guess for the next calculation, but only allow polarization: use the NO-OVERLAP keyword (See [NO-OVERLAP], page 15.) and constrain the charge of protein and water to their formal charges by using multiple Fermi energies (See Section 3.13 [Subsetting], page 15.). Then, completely relax the system by using only one Fermi energy and by allowing all subsystems to overlap, using the previously generated density matrix file as guess (See Chapter 7 [Tests], page 32.). If these tricks don't work, polarize the solvent and solute separately (See Chapter 7 [Tests], page 32.).

Use increasingly larger buffer regions for systems that are very hard to converge, using the density matrix of the previous calculation as guess. Increasing the CUTBOND (See [CUTBOND], page 11.) in steps also speeds up convergence. Computation time can be decreased by using loose convergence criteria for low quality subsetting (i.e. subsetting with small buffers and small CUTBOND values) and tightening these criteria for higher quality subsetting (See Section 3.4 [Convergence Criterion], page 4.).

Use of the SHIFT keyword (See [SHIFT], page 12.) also speeds up convergence. Suggested values are 2.5 or 5.0.

As an example, we will show here the convergence of a highly charged solvated species. Note that the number of steps and tricks is very dependent on the system of interest: your system may require more or less steps.

A) converge the solute at low quality (lowsolute.in)

```
... DIRECT CUTBOND=7.0 ECRIT=1.0E-3 DCRIT=1.0E-2 SHIFT=5.0 CHARGE=10
...
CLUSTER
  NCORE=1 DBUFF1=3.0 DBUFF2=2.0
END_CLUSTER
```

B) converge the solute at high quality (highsolute.in)

```
... DIRECT CUTBOND=8.0 CHARGE=10 GUESS
...
CLUSTER
```

```
  NCORE=1 DBUFF1=4.5 DBUFF2=2.0
END_CLUSTER
GUESS
  lowsolute.dmx
END_GUESS
```

C) converge the solvent (solvent.in)

```
... DIRECT CUTBOND=8.0 CHARGE=0
...
CLUSTER
  NCORE=1 DBUFF1=4.5 DBUFF2=2.0
END_CLUSTER
```

D) solvate the system, first polarize the solvent (polsolv.in)

```
... DIRECT CUTBOND=8.0 NO-OVERLAP GUESS CHARGE=10
...
CLUSTER
  NCORE=1 (1-100) [10] F
  NCORE=1 (101-1000) [0]
  DBUFF1=4.5
  DBUFF2=2.0
END_CLUSTER
GUESS
  highsolute.dmx
  solvent.dmx
END_GUESS
```

E) now polarize the solute (polsolute.in):

```
... DIRECT CUTBOND=8.0 NO-OVERLAP GUESS CHARGE=10
...
CLUSTER
  NCORE=1 (1-100) [10]
  NCORE=1 (101-1000) [0] F
  DBUFF1=4.5
  DBUFF2=2.0
END_CLUSTER
GUESS
  polsolv.dmx
END_GUESS
```

F) converge the whole system

```
... DIRECT CUTBOND=8.0 GUESS CHARGE=10
...
CLUSTER
  NCORE=1 DBUFF1=4.5 DBUFF2=2.0
END_CLUSTER
GUESS
  polsolute.dmx
END_GUESS
```

d

7 Test Runs and Examples

The divcon directory contains some test runs and examples. These can be found in the directory Divcon_Tests. The actual tests can be found in the following subdirectories:

Info on the tests and how to set them up can be found in the README files.

7.1 GA

This directory contains files necessary to do a simple parameterization. Basically, using the methane files from the loop test (see page 44) it is possible to fit the parameters to the heat of formation of methane. It can then be shown using these methods how the "testpoint" keyword can be used to test the parameter set developed on one set of compounds on another set of compounds. This test is very simple and is explained in the "readme" file.

7.2 Acetate/Water

This directory contains the interaction decomposition for a water-acetate system. Note that in this example, we explicitly calculate the interaction energy and calculate the electrostatic energy in two ways. In an actual calculation, some of these steps may be skipped. In the examples we focus on the energetic effects. Charge effects can be obtained by comparing the charge distributions at each point of the calculation.

A) interaction energy

First calculate the heat of formation of the water molecule in vacuum:

```
cartesian am1 charge=0 direct cluster residue
water
1 O    -0.00000    -0.00000    -1.30529 RES
2 H    -0.91985     0.20830    -1.18578
3 H    0.37253    -0.08436    -0.40757
end_coord
cluster
 ncore=1 dbuff1=4.0 dbuff2=2.0
end_cluster

Save the output file:
% mv divcon.out water.out

This will also produce a density matrix file.
Save this as water.dmx:

% mv divcon.dmx water.dmx
```

Now calculate the heat of formation of the acetate system in vacuum:

```
cartesian am1 charge=-1 direct cluster residue
acetate
1 O    -0.00000    -0.00000     1.30529 RES
2 C    0.94140    -0.21321     2.12498
3 O    2.10075    -0.47583     1.73517
```

```
4 C    0.63088    -0.14285     3.62688
5 H    1.51371    -0.34276     4.24661
6 H    0.25214     0.84901     3.90115
7 H    -0.13823    -0.87475     3.90119
end_coord
cluster
 ncore=1 dbuff1=4.0 dbuff2=2.0
end_cluster

Again, save the output and density matrix files:
% mv divcon.out acetate.out
% mv divcon.dmx acetate.dmx
```

Now calculate the heat of formation of the water-acetate system:

```
cartesian am1 charge=-1 direct cluster residue
water + acetate
1 O    -0.00000    -0.00000    -1.30529 RES
2 H    -0.91985     0.20830    -1.18578
3 H    0.37253    -0.08436    -0.40757
4 O    -0.00000    -0.00000     1.30529 RES
5 C    0.94140    -0.21321     2.12498
6 O    2.10075    -0.47583     1.73517
7 C    0.63088    -0.14285     3.62688
8 H    1.51371    -0.34276     4.24661
9 H    0.25214     0.84901     3.90115
10 H    -0.13823    -0.87475     3.90119
end_coord
cluster
  ncore=1 dbuff1=4.0 dbuff2=2.0
end_cluster

Save the output files:
% mv divcon.out wat+ac.out
% mv divcon.dmx wat+ac.dmx
```

The interaction energy is: Eint = HF[wat+ac.out] ? (HF[water.out] + HF[acetate.out]) with HF the heat of formation (this simplifies notation, the total energy may be used as well). In this case (since the molecules are small) we could have performed standard calculations as well.

```
Results (these depend slightly on your compiler, machine accuracy etc.):
                HF[water.out] = -58.97673222 kcal/mol
                HF[acetate.out] = -114.30196018 kcal/mol
                HF[wat+ac.out] = -183.49743077 kcal/mol
                Eint = -10.22 kcal/mol
```

B) electrostatic energy Here we will show two ways to calculate the electrostatic energy. The first is by using the previously obtained density matrixes:

```
cartesian am1 charge=-1 direct cluster residue &
```

```
no-overlap guess 1scf
water + acetate, electrostatics
...
end_coord
cluster
  ncore=1 (1) [0]
  ncore=1 (2) [-1]
  dbuff1=4.0 dbuff2=2.0
end_cluster
guess
  1-3 water.dmx 1-3
  4-10 acetate.dmx 1-7
end_guess
```

Note that the "..." after the title line indicate the coordinates and residue information of the water and acetate molecules, which is identical to that specified in wat+ac example. We will use this convention through the remainder of these examples.

```
Save the output files:
% mv divcon.out 1b0.out
% mv divcon.dmx 1b0.dmx

The electrostatic energy can be obtained from
Ees = HF[1b0.out] ? (HF[water.out] + HF[acetate.out])

The other way uses the push keyword:

cartesian am1 charge=-1 direct cluster residue &
no-overlap push
water + acetate, infinitely seperated
...
end_coord
cluster
  ncore=1 (1) [0]
  ncore=1 (2) [-1]
  dbuff1=4.0 dbuff2=2.0
end_cluster
```

Note that again, the "..." indicates that the coordinates are identical to the ac+wat example: you do not need to separate the systems by hand.

```
% mv divcon.out far.out
% mv divcon.dmx far.dmx

cartesian am1 charge=-1 direct cluster residue &
no-overlap 1scf guess
water + acetate, electrostatics
...
end_coord
cluster
```

```
  ncore=1 (1) [0]
  ncore=1 (2) [-1]
  dbuff1=4.0 dbuff2=2.0
end_cluster
guess
  far.dmx
end_guess

% mv divcon.out 1b0a.out
% mv divcon.dmx 1b0a.dmx

Note that by definition, the far and 1b0a density matrixes are identical
% diff far.dmx 1b0a.dmx

The electrostatic energy is now obtained from:
        Ees = HF[1b0a.out] ? HF[far.out]

Of course
        HF[1b0a.out] = HF[1b0.out]
        dmx[1b0a.out] = dmx[1b0.out] = dmx[far.dmx], dmx being the density matrix

        Results:
                HF[1b0.out] = -171.50664061 kcal/mol
                HF[far.out] = -173.27862187 kcal/mol
                HF[1b0a.out] = -171.50622547 kcal/mol
                Ees = +1.77 kcal/mol
```

For a protein-water system, the first method is recommended. Not only is this faster for larger systems, but the hole in the water system can also be filled by a continuum. In these systems, one would obtain the density matrix of the water system from

```
cartesian am1 charge=0 direct cluster residue &
cm2 scrf-inside drad=1.4 cutbond=9.0
water from a protein-water system
...
end_coord
cluster
 ncore=1 dbuff1=4.0 dbuff2=2.0
end_cluster
```

C) polarization energy Since 1b0a.dmx and 1b0.dmx are equivalent, we will use 1b0.dmx here. Now allow charge flow within the acetate and water subsystems, but not between the two:

```
cartesian am1 charge=-1 direct cluster residue &
no-overlap guess
water + acetate, polarization
...
end_coord
cluster
```

```
  ncore=1 (1) [0]
  ncore=1 (2) [-1]
  dbuff1=4.0 dbuff2=2.0
end_cluster
guess
  1b0.dmx
end_guess

% mv divcon.out b0.out
% mv divcon.dmx b0.dmx

The polarization energy is given by
        Epol = HF[b0.out] ? HF[1b0.out]

        Results:
                HF[b0.out] = -172.97083685 kcal/mol
                Epol = -1.46 kcal/mol
```

D) charge transfer

```
This can now easily be obtained from
        ECT = HF[wat+ac.out] ? HF[b0.out]
```

We could also rerun the calculation, using b0.dmx as initial guess (this will ordinarily speed up calculation of completely relaxed the water-acetate system)

```
cartesian am1 charge=-1 direct cluster residue &
guess
water + acetate
...
end_coord
cluster
  ncore=1 dbuff1=4.0 dbuff2=2.0
end_cluster
guess
  b0.dmx
end_guess

% mv divcon.out eq.out
% mv divcon.dmx eq.dmx

        ECT = HF[eq.out] ? HF[b0.out]

        Results:
                HF[wat+ac.out] -183.49743077 kcal/mol
                HF[eq.out] = -183.49743909 kcal/mol
                ECT = -10.53 kcal/mol
```

E) in-field polarization Polarization of the acetate molecule in presence of the water molecule can be calculated from:

```
cartesian am1 charge=-1 direct cluster residue &
```

```
no-overlap guess
polarization of acetate in field of water
...
end_coord
cluster
  ncore=1 (1) [0]
  ncore=1 (2) [-1]
  dbuff1=4.0 dbuff2=2.0
end_cluster
guess
  1-3 1b0.dmx 1-3 f
  4-10 1b0.dmx 4-10
end_guess

% mv divcon.out b0pa.out
% mv divcon.dmx b0pa.dmx

Epol (acetate) = HF[b0pa.out] ? HF[1b0.out]
```

In-field polarization of the water molecule in field of the acetate can be obtained likewise:

```
...
guess
  1-3 1b0.dmx 1-3
  4-10 1b0.dmx 4-10 f
end_guess

        Results:
                HF[b0pa.out] = -171.64858346 kcal/mol
                Epol (acetate) = -0.14 kcal/mol
                Epol (water) = -1.28 kcal/mol
```

Note that this calculation stores the frozen eigenvectors on disk (divcon.fdmx when single node, divcon0.fdmx, divcon1.fdmx, ..., when parallel). These files can become very large (several giga bytes)when calculating the in-field polarization of large systems.

F) intramolecular charge transfer Only for large molecular systems. For a protein consisting of a single peptide chain of residues 1-100 and charge +3, solvated in water (residues 101-2000) this can be obtained from

```
cartesian am1 charge=+3 direct cluster residue &
no-overlap cutbond=8.0
intramolecular charge transfer 1
...
end_coord
cluster
  ncore=1 (1-100) [3]
  ncore=1 (101-2000) [0]
  dbuff1=4.5 dbuff2=2.0
end_cluster
neighbors
```

```
  neighbors=2 1
end_neighbors

% mv divcon.out b0pi.out
% mv divcon.dmx b0pi.dmx

cartesian am1 charge=+3 direct cluster residue &
no-overlap cutbond=8.0
intramolecular charge transfer 2
...
end_coord
cluster
  ncore=1 (1-100) [3]
  ncore=1 (101-2000) [0]
  dbuff1=4.5 dbuff2=2.0
end_cluster


% mv divcon.out b0p.out
% mv divcon.dmx b0p.dmx

        Eintra = HF[b0p.out] ? HF[bopi.out]
```

Note that in this case the protein should be divided into residues based on the amino-acid constituents.

7.3 Belly

This test shows the application of the BELLY option to carry out restrained minimization with DivCon. The input file (min_belly.in) contained in this directory looks like:

```
PM3 STANDARD OPT=LBFGS CARTESIAN CHARGE=-1 MAXOPT=300 BELLY
TS PNC + Butylamine + H2O : BELLY Minimization of side-chains
  1     C         0.689106    0.593322    0.706687
  ................................................
 58     H        -1.194692    3.426443    1.311571
 59     H        -1.693024    1.740986    1.652896
 END_COORD
 BELLY
   atoms  21-59
 END_BELLY
```

The original system corresponds to a transition structure for the water-assisted aminolysis of carboxypenam (a bicyclic model of penicillin antibiotics). To model the reaction between benzylpenicillin and butylamine, the original structure was enlarged by adding the corresponding hydrophobic side-chains using Molecular Modeling tools. Subsequently, the PM3 minimization of the added substituents is carried out in order to properly relax their relative position with respect to the initial core. To do so, the keyword BELLY (See [BELLY], page 7.) is included in the input file used. The group of moving atoms is read from the BELLY parameter at the end of the input file min_belly.in. Only atoms 21-59 are allowed to move during minimization cycles.

7.4 CSPA

This directory contains serial and parallel job samples. In this directory, two different Makefiles and divcon.dim files are given. Makefile_cspa-parallel and divcon.dim_cspa-parallel should be used for the parallel mode, Makefile_cspa-serial and divcon.dim_cspa-serial for the serial mode. All other files (the source code .F and .c file and other header .h files) are identical for the serial and parallel versions. Copy all the source code and other header files to this directory, then copy the Makefile and divcon.dim file and compile:

```
% cp ../*F ../*c ../*h .
% cp Makefile_cspa-parallel Makefile
% divcon.dim_cspa-parallel divcon.dim
% make
for the parallel mode and analogous for the serial mode.
Copy the input file cspa.in to divcon.in and run the program
% mpirun ?np 2 ./divcon
for the parallel mode, using 2 processors, or
% ./divcon
for the serial mode.
```

7.5 LBFGS

This directory contains the input and demo files for running a full minimization using the Limited memory BFGS method included in DivCon (OPT=LBFGS). The winter flounder Antifreeze Protein (AFP), which is one of the proteins responsible for the freezing point depression observed in the polar fish blood serum, is fully minimized in vacuo using the PM3 Hamiltonian. The AFP protein is electrically neutral and has a secondary structure type of alpha-helix.

The initial coordinates were taken from a previous minimization using the AMBER force field and the LBFGS minimizer included in ROAR 2.0. The optimization criterion were selected by the DivCon program automatically. Due to the size of this system (453 atoms), the Divide and Conquer strategy is employed to carry out the SCF calculations. The keyword DIIS is included in the input file (afp_lbfgs.in) so that at the later stages of minimization the rate of convergence may be increased by applying a DIIS extrapolation technique.

For this test system, the performance of the LBFGS method is notably good since the energy minimized structure was obtained after only 418 function and gradient evaluations (38 DIIS extrapolation steps were taken). With respect to an analogous NODIIS execution, this test saves around 100 function and gradient evaluations. However, we do stress that the use of the DIIS technique should be carefully tested when studying other systems. Finally, with respect to a CONJUGATE-GRADIENT minimization, more than 1000 function and gradient evaluations are saved by using the LBFGS minimizer.

The output file afp_lbfgs.demo is also provided as obtained from a DivCon execution on a SGI R12000 processor under IRIX 6.4. About 7 hours of CPU time were spent to run this test. NOTE THAT THE EXECUTION OF THIS TEST REQUIRES MUCH CPU TIME.

7.6 Loop

This test contains three input files for methane and three input files for ethane, with a different Hamiltonian used in each file of a compound type. In other words, there is MNDO/methane, AM1/methane, and PM3/methane, and the same for ethane. This test shows how the loop works, and by copying each of the six files to loop001-loop006.in you get the literature values for heat of formation back. This test is reasily understandable from examination of the README.

7.7 SCRF

This test is a self-consistent reaction field (SCRF) calculation on acetate ion embedded into a highly polar dielectric continuum (dielectric constant = 80). The SCRF calculations[Gogonea, 1999 #832] are performed with the Poisson-Boltzmann equation as implemented in the program Delphi.[Honig, 1995 #659] A SCRF calculation is invoked with the "SCRF" keyword. The solute charges used in Delphi are either Mulliken (default), CM1 or CM2 (add "CM1" or "CM2" keyword to use these charges). Delphi calculates the solvation free energy (both the electrostatic and nonpolar parts) and Divcon calculates the free energy component due to wave function distortion. The input files required by a SCRF calculation (in addition to divcon input file) are the "pb.prm" (file with Delphi commands and parameters - see the Delphi manual for a description), "divcon.pdb" and "residue.dat" (if the input coordinates are in PDB format), "jaguar.siz" which contain the van der Waals radii for atom types (used in Delphi), and "v3.dat" (a Delphi binary file used in calculating the molecular surface area). In addition to the traditional divcon output files, a SCRF calculation also generates the Delphi output file ("divcon.pb") and other files explicitly requested to Delphi (e.g. a GRASP file with the electrostatic potential map). The "divcon.out" file contains a summary of SCRF calculations (both gas phase and solution quantities) including the solvation free energy.

7.8 TS

Two different tests are contained in this directory to illustrate the optimization of Transition Structure by using DivCon (OPT=TS See [OPT=TS], page 7.).

The first test system, which is relatively simple, corresponds to the Endo [4+2] cycloaddition between furan and maleic anhydride. The initial geometry of this reactant-like TS, was taken from a MP2/6-31G* optimized geometry. The AM1 Hamiltonian is used and full SCF calculations are performed (no D&C). The P-RFO method (OPT=TSPRFO See [OPT=TSPRFO], page 7.) is used. This algorithm works quite well for non-complex systems, in which the transition vector would have a relatively large value, being not coupled with other normal modes. After successful optimization, the character of the critical point is analyzed by means of a numerical calculation of force constants since the FREQ keyword is present in the input file (da_ts.in). The demo output file (da_ts.demo) shows optimization convergence after 14 cycles, the resultant TS being properly characterized by one imaginary frequency of 686 cm-1.

The second test system corresponds to the nucleophilic attack of the zinc complex towards the C atom in CO2 in order to model one mechanistic step involved in the catalytic action of zinc Carbonic Anhydrases. The initial geometry was built by molecular modeling. The TS is optimized at the PM3 level using the quasi Newton Algorithm (OPT=TSQNA

See [OPT=TSQNA], page 7.). For this system, optimization converges very fast after only 14 cycles and the resultant TS is characterized by one imaginary frequency of 255 cm-1 (see the znco2_ts.demo file). We note that the QNA option seems quite efficient when the TS is located on a very low curvature potential energy surface.

8 References

1. Yang, W. and T.S. Lee, *A Density-Matrix Divide-and-Conqure Approach for Electronic Structure Calculations of Large Molecules.* J. Chem. Phys., 1995. **103**(13):p. 5674-5678
2. Dixon, S.L. and K.M. Merz, Jr., *Semiempirical Molecular Orbital Calculations with Linear System Size Scaling.* J. Chem. Phys., 1994. **104**:p. 6643-6649.
3. Dixon, S.L. and K.M. Merz, Jr., *Fast, Accurate Semiempirical Molecular Orbital Calculations for Macromolecules.* J. Chem. Phys., 1997. **107**(3):p. 879-893.
4. Dewar, M.J.S. and W. Thiel, *Ground States of Molecules.38.The MNDO Method, Approximations and Parameters.*J. Am. Chem. Soc., 1977. **99**(15):p. 4899-4907.
5. Dewar, M.J.S., et al., *AM1: A New General Purpose Quamntum Mechanical Molecular Model.* J. Am. Chem. Soc., 1985. **107**(13):p. 3902-3909.
6. Stewart, J.J.P., *Optimization of Parameters for Semiempirical Methods I. Method.* J. Comp. Chem., 1989. **10**(2):p. 209-220.
7. Gogonea, V. and K.M. Merz, Jr., *Fully Quantum Mechanical Description of Proteins in Solution. Combining Linear Scaling Quantum Mechanical Methodologies with the Poisson-Boltzman Equation.* J. Phys. Chem. A, 1999. **103**:p. 5171-5188.
8. Mulliken, R.S., *Electronic Populations Analysis on LCAO-MO Molecular Wave Functions.* J. Chem. Phys., 1955. **23**:p. 1833-1840.
9. Storer, J.W., et al., *Class IV Charge Modesl:A New Semiempirical Apporach in Quantum Chemistry.* J. Comput.-Aided Mol. Design, 1995. **9**:p. 87-110.
10. Li, J., C.J. Cramer, and D.G. Truhlar, *New Class IV Charge Model for Extracting Accurate Partial Charges from Wave Functions.* J. Phys. Chem. A., 1998. **102**:p. 1820-1831.
11. Essmann, U., et al., *A Smooth Particle Mesh Ewald Method.* J. Chem. Phys., 1995. **103**:p. 8577-8594.
12. van der Vaart, A. and K.M. Merz, Jr., *Divide and Conquer Interaction Energy Decomposition.* J. Phys. Chem. A, 1999 **103**:p. 3321-3329.
13. Vincent, J.J. and K.M. Merz, Jr., *Parallel Implementation of the Divide and Conquer Algorithm.* Theor. Chem. Acc., 1998. **99**:p. 220-223.
14. Honig, B. and A. Nicholls, *Classical Electrostatics in Biology and Chemistry.* Science(Wash.), 1995. **268**:p. 1144-1149.
15. Thiel, W., *MNDOC.* 1982, Quant. Chem. Prog. Exch. No. 438.
16. Dewar, M.J.S. and H.S. Rzepa, J. Am. Chem. Soc., 1978. **100**:p. 777
17. Dewar, M.J.S. and M.L. McKee, J. Am. Chem. Soc., 1977.**99**:p. 5231.
18. Dewar, M.J.S. and H.S. Rzepa, J. Am. Chem. Soc., 1978. **100**:p. 58.
19. Stewart, J.J.P., *Mopac 6.0*, QCPE.
20. Davis, L.P., et al., J. Comp. Chem., 1981. **2**:p 433.
21. Dewar, M.J.S., et al., Organometallics, 1986. **5**:p. 375.
22. Dewar, M.J.S., M.L. McKee, and H.S. Rzepa, J. Am. Chem. Soc., 1978. **100**:p. 3607.
23. Dewar, M.J.S. and C.H. Reynolds, J. Comp. Chem., 1986. **7**:p. 140.
24. Dewar, M.J.S. and H.S. Rzepa, J. Comp. Chem., 1983. **4**:p. 158.

25. Dewar, M.J.S. and K.M.J. Merz, *MNDO Calculations for Compounds Containing Zinc.* Organometallics, 1986. **5**(7):p. 1494-1496.
26. Dewar, M.J.S., G.L. Grady, and E.F. Healy, Organometallics, 1987. **6**:p. 186.
27. Dewar, M.J.S. and E.F. Healy, J. Comp. Chem., 1983. **4**:p. 542.
28. Dewar, M.J.S., G.L. Grady, and J.J.P. Stewart, J. Am. Chem. Soc., 1984. **106**:p. 6771.
29. Dewar, M.J.S., E.F. Healy, and J.J.P. Stewart, J. Comp. Chem., 1984. **5**:p. 358.
30. Dewar, M.J.S., et al., Organometallics, 1985. **4**:p. 1964.
31. Dewar, M.J.S., et al., Organometallics, 1985. **4**:p. 1973.
32. Dewar, M.J.S., C. Jie, and E.G. Zoebisch, Organometallics, 1988. **7**:p. 513-521.
33. Dewar, M.J.S. and E.G. Zoebisch, *Extension of AM1 to the Halogens.* J. Mol. Struct. (Theochem.), 1988. **180**:p. 1-21.
34. Brothers, E.N. and K.M. Merz, Jr, *Sodium Parameters for AM1 and PM3 Optimized Using a Modified Genetic Algorithm.*J. Phys. Chem. B. 2001. **106**p: 2779-2785.
35. Dewar, M.J.S. and A.J. Holder, Organometallics, 1990. **9**:p. 508.
36. Dewar, M.J.S. and C. Jie, Organometallics, 1987. **6**:p. 1486-1490.
37. Dewar, M.J.S. and C. Jie, J. Mol. Struct.(Theochem), 1989. **187**:p. 1.
38. Dewar, M.J.S. and Y.C. Yuan, Inorg. Chem., 1990. **29**:p. 3881-3890.
39. Dewar, M.J.S. and K.M.J. Merz, *AM1 Parameters for Zinc.* Organometallics, 1988. **7**(2):p. 522-524.
40. Dewar, M.J.S. and C. Jie, Organometallics, 1989. **8**:p. 1544.
41. Dewar, M.J.S. and C. Jie, Organometallics, 1989. **8**:p. 1547.
42. Stewart, J.J.P., *Optimization of Parameters for Semiempirical Methods. III Extension of PM3 to Be, Mg, Zn, Ga, Ge, As, Se, Cd, In, Sn, Sb, Te, Hg, Tl, Pb and Bi.* J. Comp. Chem., 1991. **12**(3):p. 320-341.
43. Mitin, A.V., *The Dynamic Level Shift Method for Improving the Convergence of the SCF Procedure.* J. Comp. Chem., 1988. **9**:p. 107-110.
44. Ermolaeva, M.D., A. van der Vaart, and K.M.J. Merz, *Implementation and Testing of a Frozen Density Matrix - Divide and Conquer Algorithm.* J. Phys. Chem., 1999. **103**:p. 1868-1875.
45. Jensen, F., *An Introduction to Computational Chemistry.* 1999, New York, NY:John Wiley & Sons.
46. Baysal, C., H. Meirovitch, and M. Navon, *Performance of Efficient Minimization Algorithms as Applied to Models of Peptides and Proteins.* J. Comput. Chem., 1999. **20**:p. 354-364.
47. Eckert, F., P. Pulay, and H.J. Werner, J. Comput. Chem., 1997. **18**:p. 1473-1483.
48. Rossi, I. and D.G. Truhlar, *Parameterization of NNDO Wavefunction Using Genetic Algoriths. An Evolutionary Approach to Parameterizing Potential Energy Surfaces and Direct Dynamics Calculations for Organic Reactions.* Chem. Phys. Lett., 1995. **233**:p. 231-236.
49. Hutter, M.C., J.R. Reimers, and N.S. Hush, *Modeling the Bacterial Photosynthetic Reaction Center. 1.Magnesium Parameters for the Semiempirical AM1 Method Developed Using a Genetic Algorithm.* J. Phys. Chem. B., 1998. **102**:p 8080-8090.

50. Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning.* 1989, San Mateo, California: Addison-Wesley.

51. Press, W.H., et al., *Numerical Recipes:The Art of Scientific Computing (FORTRAN version).* 1989, Cambridge, M.A.: Cambridge Press.

52. van der Vaart, A. and K.M. Merz, Jr., *The Role of Polarization and Charge Transfer in the Solvation of Biomolecules.* J. Am. Chem. Soc., 1999. **121**(39):p. 9182-9190.

53. van der Vaart, A., et al., *Are Many-Body Effecs Important in Protein Folding?* J. Phys. Chem. B., 2000. **104**(40):p. 9554-9563.

Index

QMScore Index