

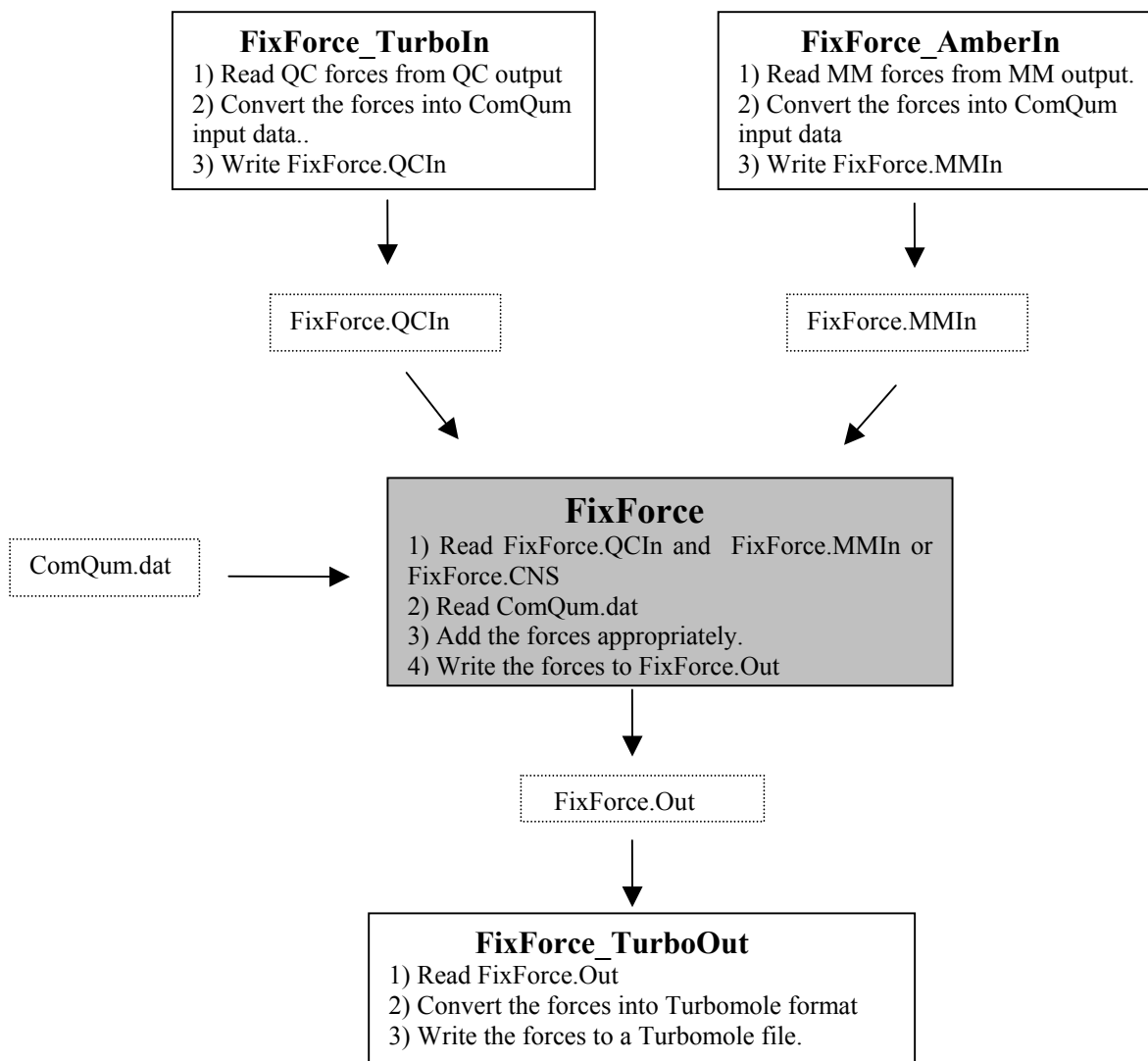
1	THE NEW VERSION OF COMQUM	2
1.1	THE STRUCTURE OF THE FIVE PROGRAMS IN COMQUM-01. EXAMPLE : FIXFORCE.	2
1.2	DESCRIPTION FILE FORMAT OF THE TEXT FILES IN COMQUM	3
1.3	EXAMPLE OF THE FORMAT A TEXT FILE : FIXFORCE.MMIN	4
2	FILES READ AND WRITTEN BY COMQUM.....	5
2.1	COMQUM.DAT	6
2.2	FIXFORCE	8
2.2.1	<i>The format of the files read and written by the general FixForce program.</i>	<i>9</i>
2.2.2	<i>A short description of the data converting programs for FixForce.....</i>	<i>10</i>
2.3	FIXCOORD1.....	11
2.3.1	<i>The format of the files read and written by the general FixCoord1 program.</i>	<i>12</i>
2.3.2	<i>A short description of the data converting programs for FixCoord1.....</i>	<i>13</i>
2.4	FIXCOORD2.....	14
2.4.1	<i>The format of the files read and written by the general FixCoord2 program</i>	<i>15</i>
2.4.2	<i>A short description of the data converting programs for FixCoord2.....</i>	<i>16</i>
2.5	FIXENERGY	17
2.5.1	<i>The format of the files read and written by the general FixEnergy program.....</i>	<i>18</i>
2.5.2	<i>A short description of the data converting programs for FixEnergy.....</i>	<i>19</i>
2.6	FIXCHARGE.....	20
2.6.1	<i>The format of the files read and written by the general FixCharge program.</i>	<i>20</i>
2.6.2	<i>A short description of the data converting programs for FixCharge.....</i>	<i>22</i>
3	FIGURE FOR THE COMQUM-X ARTICLE.....	23

1 The new version of ComQum

This section gives a description of the new modular construction of ComQum-01. This new structure in ComQum-01, which made inclusion of CNS into ComQum-X straightforward, is described in Section 1.1. In Section 1.2 the general format of the text files which are read and written by the programs in ComQum-01 and ComQum-X is described. In Section 1.3 an example of the format is given.

1.1 The structure of the five programs in ComQum-01. Example : FixForce.

Compared to ComQum-00, only the program structure is different in ComQum01 – the QC/MM calculation is performed in exactly the same manner. In ComQum-01, each of the five programs FixForce, FixCoord1, FixCoord2, FixEnergy, and FixCharge have been split into four new programs, such that the procedures taking care of reading and writing the specific QC and MM information are separated from the ComQum specific routines. The new structure of the programs made the inclusion of the CNS program and thereby ComQum-X straightforward.



Figur 1. The flow of FixForce in ComQum-01. The boxes with dashed lines show the files names and and the boxes with solid lines the program names (in bold face letters) and tasks of the programs.

The principle of the construction of the programs in ComQum-01 is described with FixForce as an example, where the QC program Turbomole and MM program Amber are coupled. The principle of the construction of the four other ComQum programs is the same.

The idea behind the modularisation of ComQum-01 is shown in Fig.1. It shows that different small programs (FixForce_TurboIn and FixForce_MMIn) provide the core of the FixForce program (see the gray box in Figure 1) with data. These data converting programs read the forces from the QC output, the MM output. The forces are written to text files FixForce.QCIn and FixForce.MMIn in standard format and units. The central FixForce program is completely general and independent of the type of QC or MM program. FixForce reads the data from the text files FixForce.QCIn and FixForce.MMIn and adds the forces appropriately after the scheme described in previous ComQum articles. The result is written to the text file FixForce.out file in standard format and units. The file FixForce.Out is then read by the program FixForce_TurboOut, which converts the forces into the right format and units for the QC program. Note that all the central ComQum programs need specific information which is independent of the QC and MM programs (for example which atoms that are the link atoms, etc.). This information is read from the text file ComQum.dat.

The aim of constructing the program in this manner is to minimize the effort when other QC and MM are coupled. It means that only the small data reading programs FixForce_TurboIn and FixForce_MMIn and the data writing program FixForce_TurboOut need to be modified when other QC and MM programs are used for the ComQum calculation. Therefore, in ComQum-X, where Turbomole is coupled with the CNS program, FixForce_MMIn is changed with a program called FixForce_CNSIn, which converts the data into ComQum format.

In Section 2 flow schemes of all five ComQum programs are shown for the coupling of Turbomole with Amber or CNS.

1.2 Description file format of the text files in ComQum

The design of the format of the text files that the central ComQum programs reads (i.e the files FixForce.QCIn and FixForce.MMIn for the FixForce) and the output format of ComQum (i.e. the files FixForce.Out for the FixForce program) is a very important part to consider. This format should be simple, unambiguous, and informative enough to work with many different programs. With the information about the format of the text files read and written by the ComQum programs, it is straightforward to make the small data converting programs yourself and thereby couple other QC or MM programs.

All files contains keywords (**\$keyword**) describing the information following it. Unless only one line of information follows, the line below the keyword contains an integer with the number of lines with the information that is to be read. Below this integer the information is given. After this information follows either a new keyword or the keyword **\$end** denoting the end of the file. The lines containing keywords must begin with \$; i.e. blanks are not accepted before the keyword.

The following notation will be used to describe the contents of the information:

- d: floating point real of double precision
- d(*n,m*): array of double precision of dimension (*n,m*). In the file, there will be a *n* columns and *m* rows.
- i: integer.
- c: character
- *cn*: string of length *n*.

Arrays of integers and characters have similar notation as the array of double precision. Energies and forces are written to the files in the FORTRAN format e20.14, which means that they are written with 14 significant digits. Coordinates of the atoms and point charges are written with f12.8; i.e. with 8 significant digits. This should be sufficient accuracy, since for example energies normally are not more precise than 10^{-6} au, and they are normally not larger than 10^4 au, which means a need of 10 significant digits. Arrays cannot be dynamically allocated in FORTRAN77, and therefore the arrays are defined with predefined length. Therefore there is a maximum number of atoms, residues and so forth in this implementation. For the number of atoms this is for example 20000. However, this can easily be changed, if necessary. Variables are read in free format (with one exception). The units used in ComQum is kJ/mol for energies and Ångströms for coordinates.

1.3 Example of the format a text file : FixForce.MMin

The text file FixForce.MMin contains the cartesian forces of system 1 and system 2 calculated at the MM level. Below, an example of the format describing the contents of FixForce.MMin and an explicit example is given. In the box describing the general format of FixForce.MMin (the box to the left) it is seen that after the keywords **\$force_mm1** and **\$force_mm2**, there is an integer denoted “i”. Then follows an array of double precision reals of dimension (natoms1 x 3) and (natoms3 x 3) denoted by “d(natoms1,3)” and “d(natoms3,3)” with the cartesian forces. In the explicit example, (the box to the left) the number of atoms in system 1 (natom1) is “2” and in system 2 (natom2) it is “4”. The file ends with **\$end**.

General format and example of FixForce.MMin.

```
$force_mm1
i
d(natom1,3)
$force_mm2
i
d(natom2,3)
$end
```

```
$force_mm1
2
0.36232485109456E+02 0.11108461915851E+02 0.18624856471895E+02
0.11681316181692E+03 -.12253901972884E+02 0.43307464922536E+02
$force_mm2
4
0.18576683363133E+02 -.45523460164240E+02 0.41652276773876E+02
0.33732140477893E+02 -.92431128919330E+01 0.11572701377346E+02
-.80488142695200E+01 0.12457633117119E+02 -.26140681623734E+02
-.16107317138842E+03 0.54562841911943E+02 -.70391761450024E+02
$end
```

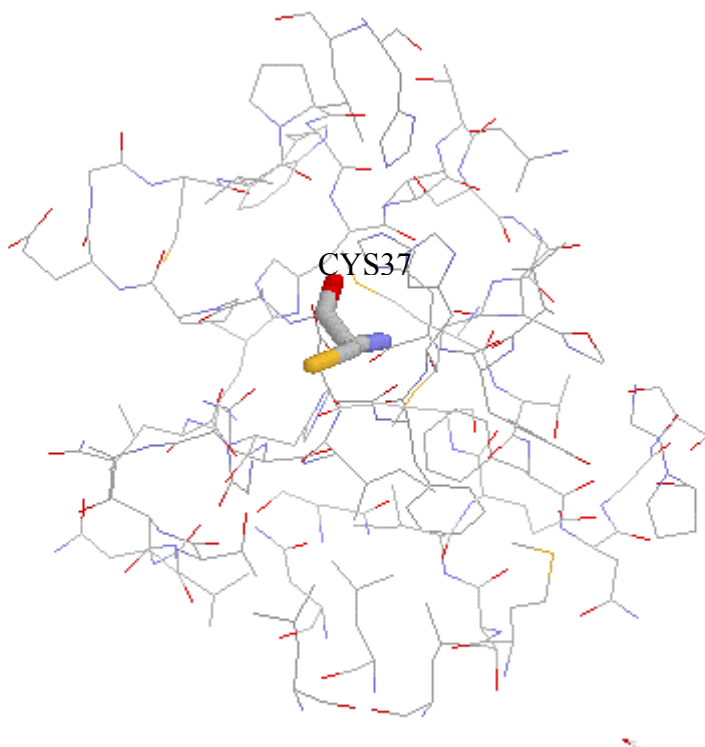
Reading the files is particularly easy with the construction as shown above, where one type of information is found in a block surrounded by keywords. This is done using a subroutine which finds the keywords.

The format of all the files read and written by the ComQum programs are given in Section 2.

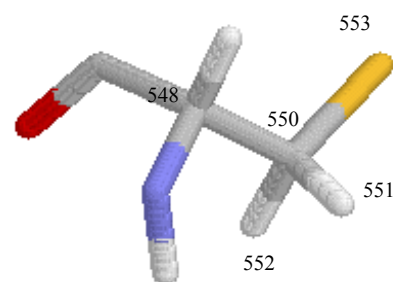
2 Files read and written by ComQum.

Section 2 gives a detailed description of the flow of each of the five ComQum programs (FixForce, FixCoord1, FixCoord2, FixEnergy, and FixCharge) and the format of the text files which are read and written by them. In Section 2.1, the type and format of the data in the text file ComQum.dat, which is necessary for all the ComQum programs, is described. In Section 2.2-6, flow schemes of the five ComQum programs are given. In all the schemes, the programs are shown in boxes with full drawn lines, the programs containing the general ComQum procedures have gray background, the names of the files are shown in boxes with dashed lines, and the names of the files specific for the QC or MM program are written with italic letters (i.e. similar to Figure 1). In the flow schemes, the programs used for coupling Turbomole with either Amber (ComQum-01) or CNS (ComQum-X) is shown. Each flow scheme of the ComQum programs is followed by an explanation of the format of the files read and written by the core of the ComQum programs containing the general procedures, and a short description of the data converting programs used when coupling of Turbomole with Amber or CNS. The notation from Section 1.2 will be used to describe the formats of the files.

All examples of file formats used in Section 2.1-6 uses the molecular system as shown below in the Figure 2 and 3.



Figur 2 Nitrate reductase



Figur 3 The QC system. The numbers of the atoms are shown

The molecule is the enzyme blue copper protein nitrite reductase, which contain a copper ion, a cysteine, methionine, and a histidine in the active site. The QC system in these test calculations is seen in Fig. 3. Atom number 548 (the link atom, CA), 550 (CB), 551 and 552 (HB), and 553 (SG) were selected as quantum system in the cysteine 37 residue. System 2 was selected as any amino acid within 5 Å from any atom system 1, and system 3 as the rest. The number of atoms in system 1, system 2, and system 3 are in this appendix denoted “*natom1*”, “*natom2*”, and “*natom3*”, the number of pointcharges “*nptchg*”, and the number of amino acids in system 3 “*dimres*”. In this example *natom1*=5, *natom2*=203, *natom3*=829, *nptchg*=824, and *dimres*=60.

2.1 *ComQum.dat*

ComQum.dat is a text file containing information about which atoms that are QC atoms, junction atoms, the type of junction, the lengths of the bond between the junction atom and its QC neighbor, and a correspondence list of the QC and MM atoms is needed for the *ComQum* calculation. It contains the following keywords:

- **\$title** : After this keyword there is a title of up to 10 lines.
- **\$protein**: The keyword is either "fix" or "free". It tells whether system 2 is fixed or relaxed in the MM optimisations.
- **\$junction=**: The type of junction. It means on which basis the bond length of the bond that is broken has been determined.
- **\$junction_atoms**: In each row there are three numbers, two integers and one floating point real. The first integer is the number of junction atom. This is bonded to a QC atom, which number is the second integer. The real number gives the ratio between the normal XC and XH bond (X is the QC atom). This information is needed for each junction (the number of junction atoms is *njunct*)
- **\$correspondence_list**: A correspondence list of the QC and MM atoms. The first integer gives the number of atoms in system 1 (*natom1*). In the next lines, the first integer gives the number of the MM atom corresponding to the first QC atom, the second integer the number of the MM atom corresponding to the second QC atom, etc.

Below there is a figure with the general structure and an example of the *ComQum.dat* file. In this file, the title is “Test, 1/11-00” system 2 is relaxed in the calculations, the type of junction is 6, the number of the atoms of the junction atom and the QC atom bonded to it are 1 and 2, respectively, and the the ratio between the C-X and HX bond lengths is 1.381495. There are in total five QC atoms and number 1,2,3,4,5 corresponds to MM atoms 548, 550, 551, 552, and 553 in that order.

General format and example of ComQum.dat

```
$title  
c80(dimtit)  
$protein c3/c4  
$junction= i  
$junction_atoms  
i(njunct) i(njunct) d(njunct)  
$correspondance_list  
i  
i(natom1)  
$end
```

```
$title  
Test, 1/11-00  
$protein free  
$junction= 6  
$junction_atoms  
1 2 1.381495  
$correspondance_list  
5  
548 550 551 552  
553  
$end
```

2.2 FixForce

Below the flow of the programs used for adding the forces appropriately with ComQum is shown. This is done by the action of programs `FixForce_TurboIn`, `FixForce_AmberIn`, `FixForce`, and `FixForce_TurboOut` when coupling Turbomole and Amber using ComQum-01, and the `FixForce_TurboIn`, `FixForce_CNSIn`, `FixForce`, and `FixForce_TurboOut` when coupling Turbomole and CNS using ComQum-X. All the general ComQum procedures concerning addition of the QC and MM forces is found in `FixForce` program – see the box with the gray background in Figure 4.

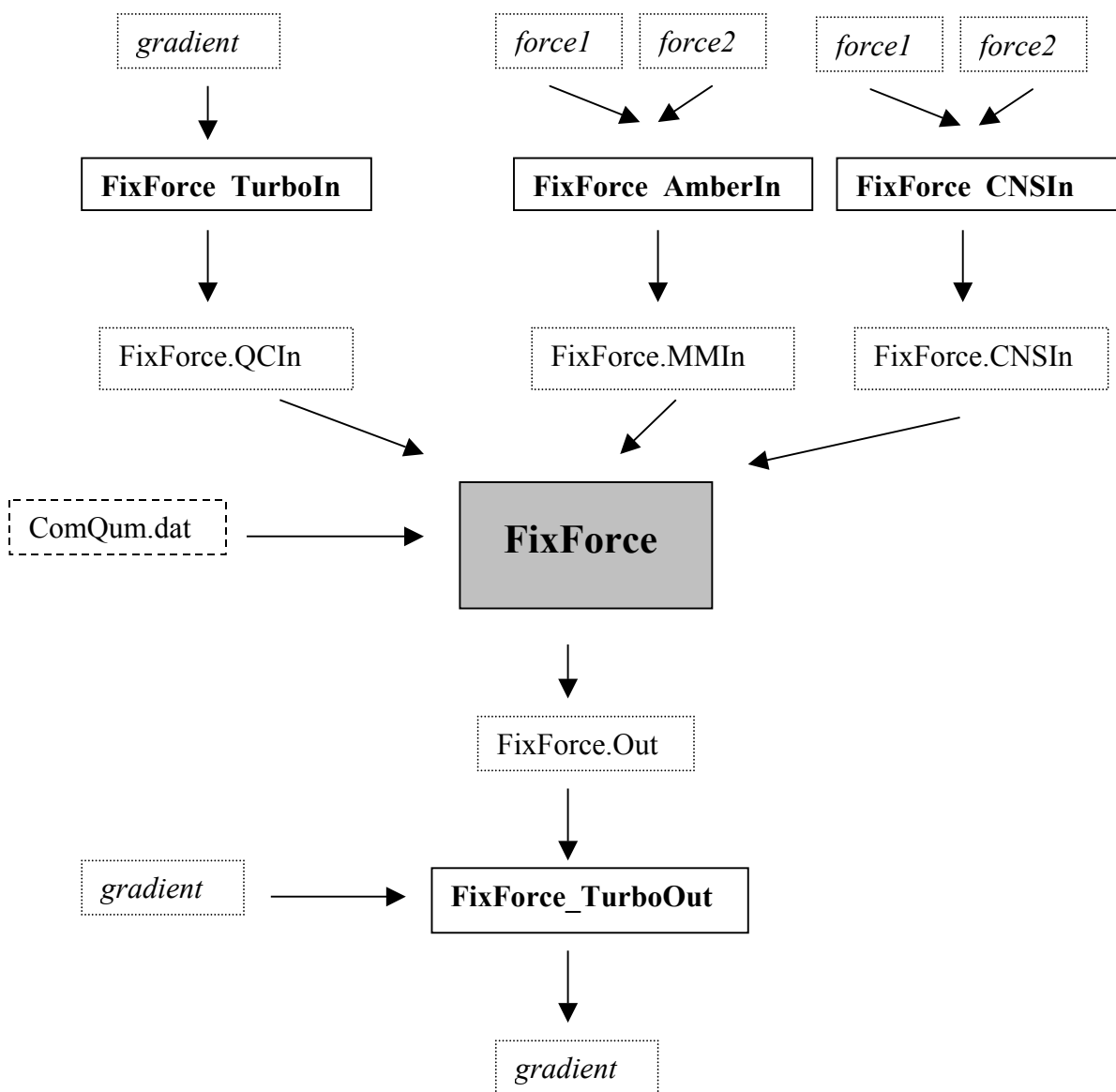


Figure 4. The flow of `FixForce` coupling Turbomole with Amber or CNS.

2.2.1 The format of the files read and written by the general FixForce program.

The program FixForce reads the QC and MM forces from the text files FixForce.QCIn and FixForce.MMIn in ComQum-01 and FixForce.QCIn and FixForce.CNSIn in ComQum-X. Adding the forces also requires information about the junction atoms, which is read from the file ComQum.dat. After adding the QC and MM forces appropriately the forces, the QC/MM forces are written to the text file FixForce.Out. Below there is a description of the format of the input and output files for FixForce.

- **FixForce.QCIn**

The file FixForce.QCIn contains the QC forces. It contains the keyword **\$force_qc**. After the keyword, the first integer gives the number of QC atoms (natom1), and after that follows the three columns with the cartesian QC forces in kJ/mol/Å and one column with the atoms names. In the example, the number of atoms (natom1) is 5, and the cartesian forces are given for the atoms with the names h,c,h,h, and s.

General format and example of FixForce.QCIn

```
$force_qc  
i  
d(natom1,3) c4(natom1,1)  
$end
```

```
$force_qc  
5  
0.52955977013031E+02 -.29760503064972E+01 0.42285814723517E+02 h  
-.25025590652238E+02 -.73288606096896E+02 0.24964506433287E+02 c  
-.11837036362445E+02 0.55138068691807E+02 0.19244673782649E+02 h  
0.11133820514702E+02 0.17135280067474E+02 -.11186892512454E+03 h  
0.36232485109456E+02 0.11108461915851E+02 0.18624856471895E+02 s  
$end
```

- **FixForce.MMIn / FixForce.CNSIn**

The format and type of information in file FixForce.MMIn and FixForce.CNSIn is the same. They contain the keywords **\$force_mm1** and **\$force_mm2**. After the keyword **\$force_mm1**, the number of atoms in system 1 is given, which is followed by three columns containing the forces of system 1; similarly for system 2. The units of the forces is kJ/mol/Å. In the example the number of atoms of system 1 (natom1) and system 3 (natom3) is 5 and 829, respectively. Not all the forces for the 829 atoms are shown.

General format and example of FixForce.MMIn and FixForce.CNSIn

```
$force_mm1  
i  
d(natom1,3)  
$force_mm2  
i  
d(natom3,3)  
$end
```

```
$force_mm1  
5  
0.11681316181692E+03 -.12253901972884E+02 0.43307464922536E+02  
0.18576683363133E+02 -.45523460164240E+02 0.41652276773876E+02  
0.33732140477893E+02 -.92431128919330E+01 0.11572701377346E+02  
-.80488142695200E+01 0.12457633117119E+02 -.26140681623734E+02  
-.16107317138842E+03 0.54562841911943E+02 -.70391761450024E+02  
$force_mm2  
829  
0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00  
0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00  
.  
$end
```

- **FixForce.Out**

The file Fixforce.Out contains the QC/MM forces calculated by FixForce. After the keyword **\$force_comqum**, the number of atoms in system 1 is given, followed by three columns with the cartesian QC/MM forces. The forces are given in kJ/mol/Å. In the example, the number of atoms in system 1 (natom1) is 5, and thereafter the forces are given in three columns.

General format and example of FixForce.out

```
$force_comqum
i
d(natom1,3)
$end
```

```
$force_comqum
5
0.60452681520967E+02 -.66636187665231E+01 0.35561388996870E+01
-.48782471222052E+02 -.64899527596184E+02 0.67727292299270E+02
-.91181333098199E+00 0.57411498681567E+02 0.13211482045437E+02
0.18352209861070E+02 0.79939640133080E+01 -.96194083230921E+02
-.77531208957399E+00 -.90846991050010E+01 0.20486156491719E+02
$end
```

2.2.2 A short description of the data converting programs for FixForce.

Coupling Turbomole with Amber and CNS in ComQum-01 and ComQum-X, respectively, requires some input and output programs for converting the data into the standard format used in ComQum and for transforming the data back to a format readable for Turbomole. For this purpose the programs FixForce_TurboIn, FixForce_AmberIn, FixForce_CNSIn, and FixForce_TurboOut are used. Below follows a short description of these input and output programs for general FixForce program.

- **FixForce_TurboIn**

The program FixForce_TurboIn reads the gradients (in atomic units) of system 1 from the Turbomole output file "gradient". It converts the gradients to forces (in kJ/mol/Å) and writes the data to the file FixForce.QCIn.

- **FixForce_AmberIn / FixForce_CNSIn**

The program FixForce_AmberIn or FixForce_CNSIn read the forces of system 1 and system 2 from Amber or CNS output files with the names "force1" and "force2". The programs write the forces in kJ/mol/Å to the files FixForce.MMIn or FixForce.CNSIn.

- **FixForce_TurboOut**

The program FixForce_TurboOut reads the QC/MM forces from the file FixForce.Out, converts them to gradients, and writes them to the file = "gradient" used for the relaxation of system 1 by Turbomole. Information from the Turbomole input file "control" is needed for that operation.

2.3 FixCoord1

In Figure 5 the flow of the programs used for inserting the new coordinates after optimisation of system 1 from the QC representation into the MM representation. This is done by the action of the programs `FixCoord1_TurboIn`, `FixCoord1_AmberIn`, `FixCoord1`, and `FixCoord1_AmberOut` when coupling Turbomole and Amber using ComQum-01, and `FixCoord1_TurboIn`, `FixCoord1_CNStn`, `FixCoord1`, and `FixCoord1_CNStnOut` when coupling Turbomole and Amber using ComQum-X. All the general ComQum procedures concerning creation of the new coordinates in the MM representation is found in `FixCoord1` – see the box with the gray background in Figure 5.

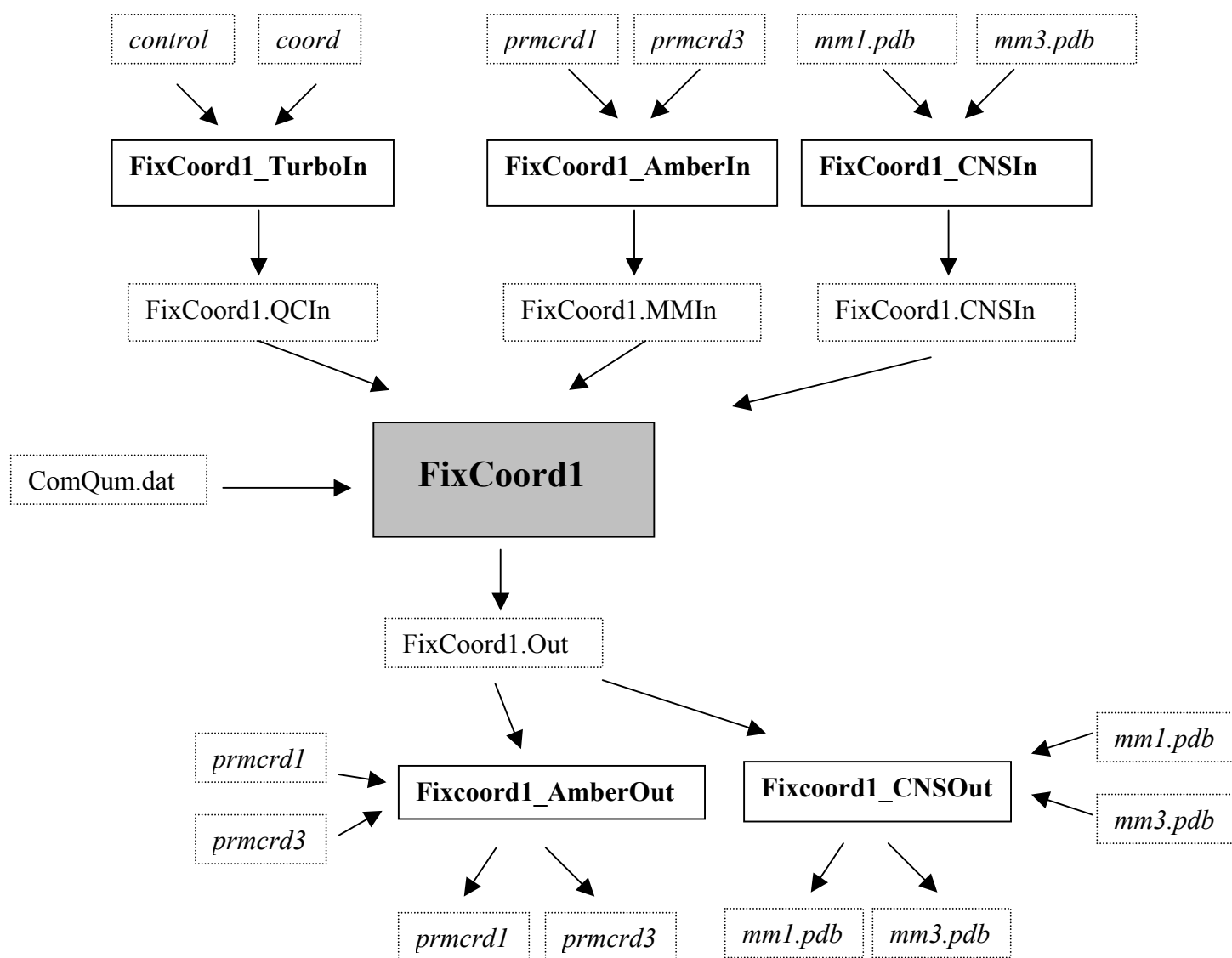


Figure 5. The flow of FixCoord1 coupling Turbomole with Amber or Cns.

2.3.1 The format of the files read and written by the general FixCoord1 program.

The program FixCoord1 reads the coordinates in the QC representation of system 1 from the file FixCoord1.QCIn and the coordinates in the MM representation of system 1 and 3 from the file FixCoord1.MMIn in ComQum-01. This information is read from FixCoord1.QCIn and FixCoord1.CNSIn in ComQum-X. The new coordinates in the MM representation of system 1 and 3 are written to the text file FixCoord1.Out. Below there is a description of the format of the input and output files for FixCoord1.

- **FixCoord1.QCIn**

The file FixCoord1.QCIn contains the coordinates of the atoms of system 1 in the QC representation. After the keyword **\$coord_qc**, an integer gives the number of atoms (natom1) of system 1, which is followed by three columns containing the xyz coordinates and one column with the atom names. In the example, the number of atoms of system 1 is 5, and the xyz coordinates are given for the atoms with the names h,c,h,h, and s.

General format and example of FixCoord1.QCIn

<pre> \$coord_qc i d(natom1,3) c4(natom1,3) \$end </pre>	<pre> \$coord_qc 5 -3.84775227668697 -1.28866628508396 -1.11987299097462 h -2.85548143476017 -0.77417729939134 -1.11801294000413 c -3.06758656797341 0.28977094315711 -0.88819625123372 h -2.47842064410493 -0.81440362169368 -2.15857968008049 h -1.50293480606301 -1.50107692508749 0.00178005444316 s \$end </pre>
--	---

- **FixCoord1.MMIn/FixCoord1.CNSIn**

The format and type of information in the files FixCoord1.MMIn and FixCoord1.CNSIn is the same. They contain the keywords **\$xyz_mm1** and **\$xyz_mm2**. After the keyword **\$xyz_mm1**, the number of atoms in system 1 (natom1) is given and in the following lines, the xyz coordinates of the atoms of system 1 are given; similarly for system 3. Not all the coordinates are shown for system 3.

General format and example of FixCoord1.MMIn/FixCoord1.CNSIn

<pre> \$xyz_mm1 i d(natom1,3) \$xyz_mm3 i d(natom3,3) \$end </pre>	<pre> \$xyz_mm1 5 -3.83873190000000 -1.29176450000000 -1.12392040000000 -2.83614570000000 -0.78209420000000 -1.11970820000000 -3.04832110000000 0.25728030000000 -0.85071150000000 -2.46241810000000 -0.80532280000000 -2.13532930000000 -1.44261520000000 -1.52229450000000 -0.06122150000000 \$xyz_mm3 829 1.97100000000000 -7.07100010000000 -11.54000000000000 2.01000000000000 -7.80900000000000 -10.85200020000000 2.10500000000000 -5.68300010000000 -11.10900020000000 2.41400000000000 -5.07100010000000 -11.95600030000000 . \$end </pre>
---	--

- **FixCoord1.Out**

The file FixCoord1.Out contains the new coordinates of system 1 and 3 in the MM representation. The format is similar to the format of FixCoord1.MMIIn and FixCoord1.CNSIn.

2.3.2 A short description of the data converting programs for FixCoord1.

- FixCoord1_TurboIn

The program FixCoord1_TurboIn reads the cartesian coordinates of system 1 from the file "coord". It is also necessary to read the Turbomole input file "control" to write the text file FixCoord.QCIn.

- FixCoord1_AmberIn/FixCoord1_CNSIn

The program FixCoord1_AmberIn and FixCoord1_CNSIn read the coordinates in the MM representation for system 1/system 3 from the files prmcrd1/prmcrd3 and mm1.pdb/mm3.pdb. The files FixCoord1.MMIIn and FixCoord1.CNSIn are written containing these coordinates.

- FixCoord1_AmberOut/FixCoord1_CNSOut

The program FixCoord1_AmberOut/FixCoord1_CNSOut write the new coordinates in the MM representation in format readable for Amber to the files prmcrd1 and prmcrd3, and for CNS to the files mm1.pdb and mm3.pdb.

2.4 FixCoord2

In Figure 6 the flow of the programs used for changing the positions of the pointcharges surrounding system 1 in the QC calculation is shown. Note that this operation is not performed in ComQum-X. The general procedures for changing the position of the pointcharges is found in the program FixCoord2, see the gray box in Figure 6.

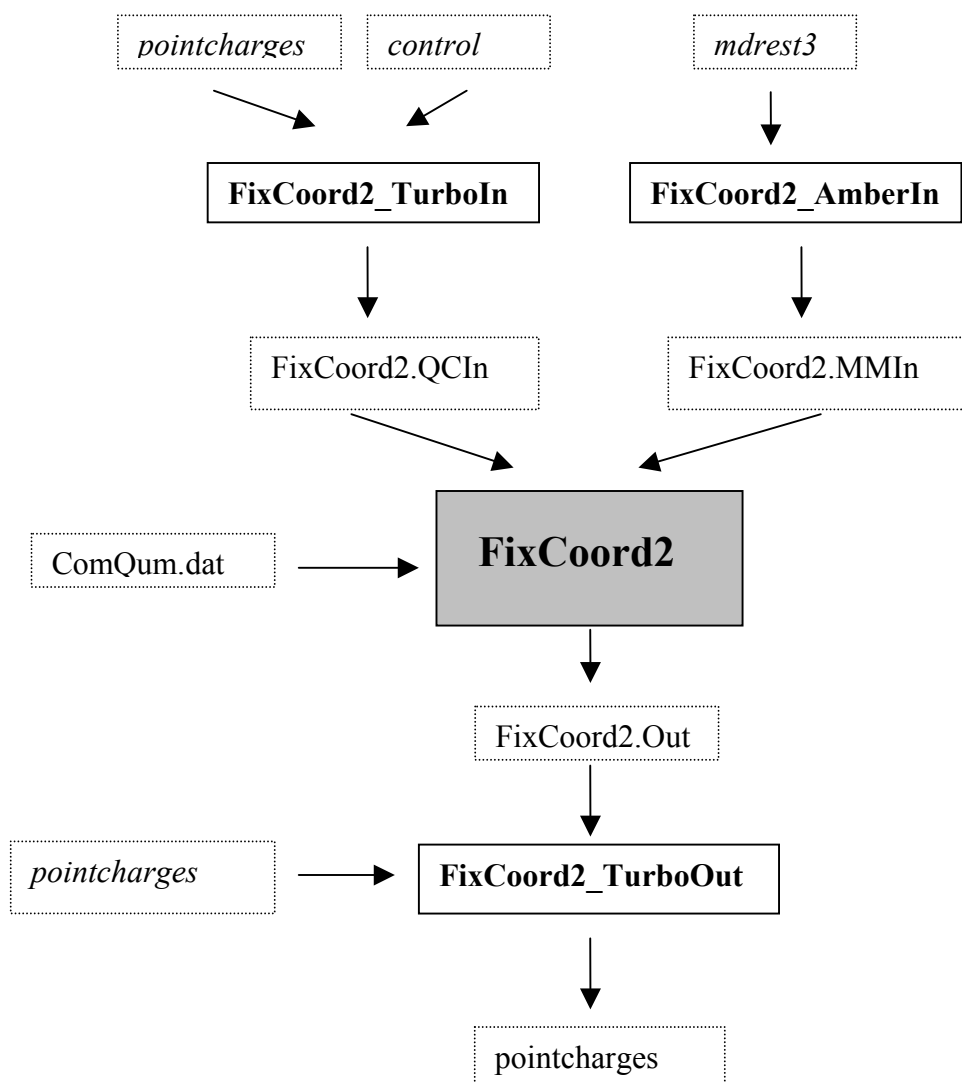


Figure 6. The flow of FixCoord2 coupling Turbomole and Amber.

2.4.1 The format of the files read and written by the general FixCoord2 program

The program FixCoord2 reads the old positions of the charges from the file FixCoord2.QCIn and the new positions from the file FixCoord2.MMIn. The old positions of the pointcharges are only used for comparison reasons. The new positions of the pointcharges are written to the FixCoord2.Out. Below there is a description of the format of FixCoord2.QCIn, FixCoord2.MMIn, and FixCoord2.Out.

- **FixCoord2.QCIn**

The file FixCoord2.QCIn contains the values and old positions of the pointcharges used in the QC calculation. After the keyword **\$point_charges** the integer gives the number of pointcharges (nptchg). Then follows three columns with the positions of the pointcharges and one column with the values of the charges. In the example there are 824 (829 - 5) pointcharges. Not all are shown in the example.

General format and example of FixCoord2.QCIn

<pre>\$point_charges i d(nptchg,3) d(nptchg,1) \$end</pre>	<pre>\$point_charges 824 1.97100004048785 -7.07100028298200 -11.54000004857806 -0.41570000 2.00999987456190 -7.80899987045063 -10.85200030265844 0.27190000 2.10499989101138 -5.68300010948994 -11.10899999623053 -0.02370000 2.41400024531095 -5.07100027089445 -11.95600004262544 0.08800000 . \$end</pre>
--	--

- **FixCoord2.MMIn**

The file FixCoord2.MMIn contains the new positions of the pointcharges. After the keyword **\$xyz_mm3** an integer gives the number of atoms in system 3 (natom3), followed by three columns containing the positions of the atoms. In the example, it is seen, that there are 829 atoms in system. Not all the coordinates are shown in the example.

General format and example of FixCoord2.MMIn

<pre>\$xyz_mm3 i d(natom3,3) \$end</pre>	<pre>\$xyz_mm3 829 1.9710000000000000 -7.071000100000000 -11.540000000000000 2.0100000000000000 -7.809000000000000 -10.852000200000000 2.1050000000000000 -5.683000100000000 -11.109000200000000 2.4140000000000000 -5.071000100000000 -11.956000300000000 0.7970000000000000 -5.100999800000000 -10.579000500000000 . \$end</pre>
--	--

- **FixCoord2.Out**

The file FixCoord2.Out contains the new positions of the pointcharges and their values. The format is identical to the format of the file FixCoord2.QCIn.

2.4.2 A short description of the data converting programs for FixCoord2.

- FixCoord2_TurboIn

The program FixCoord2_TurboIn reads the file “pointcharges” and “control” to get the positions and old values of the pointcharges. They are written to the file FixCoord2.QCIn.

- FixCoord2_AmberIn

The program FixCoord2_AmberIn reads the new positions of the pointcharges after relaxation of system 2 from the Amber output file “mdrest3”. They are written to the file FixCoord2.MMIn.

- FixCoord2_TurboOut

The program FixCoord2_TurboOut reads the new positions and the values of the pointcharges from the file FixCoord2.Out. They are written to the Turbomole file “pointcharges”.

2.5 FixEnergy

The flow of the programs adding the QC and MM energies appropriately are shown in Figure 7. This is done with the action of the programs `FixEnergy_TurboIn`, `FixEnergy_AmberIn`, `FixEnergy`, and `FixEnergy_TurboOut` when coupling Turbomole and Amber using ComQum-01, and `FixEnergy_TurboIn`, `FixEnergy_CNSIn`, `FixEnergy`, and `FixEnergy_TurboOut` when coupling Turbomole and CNS in ComQum-X. The general ComQum procedures for adding the energies appropriately are found in the program `FixEnergy`— see the gray box in Figure 7.

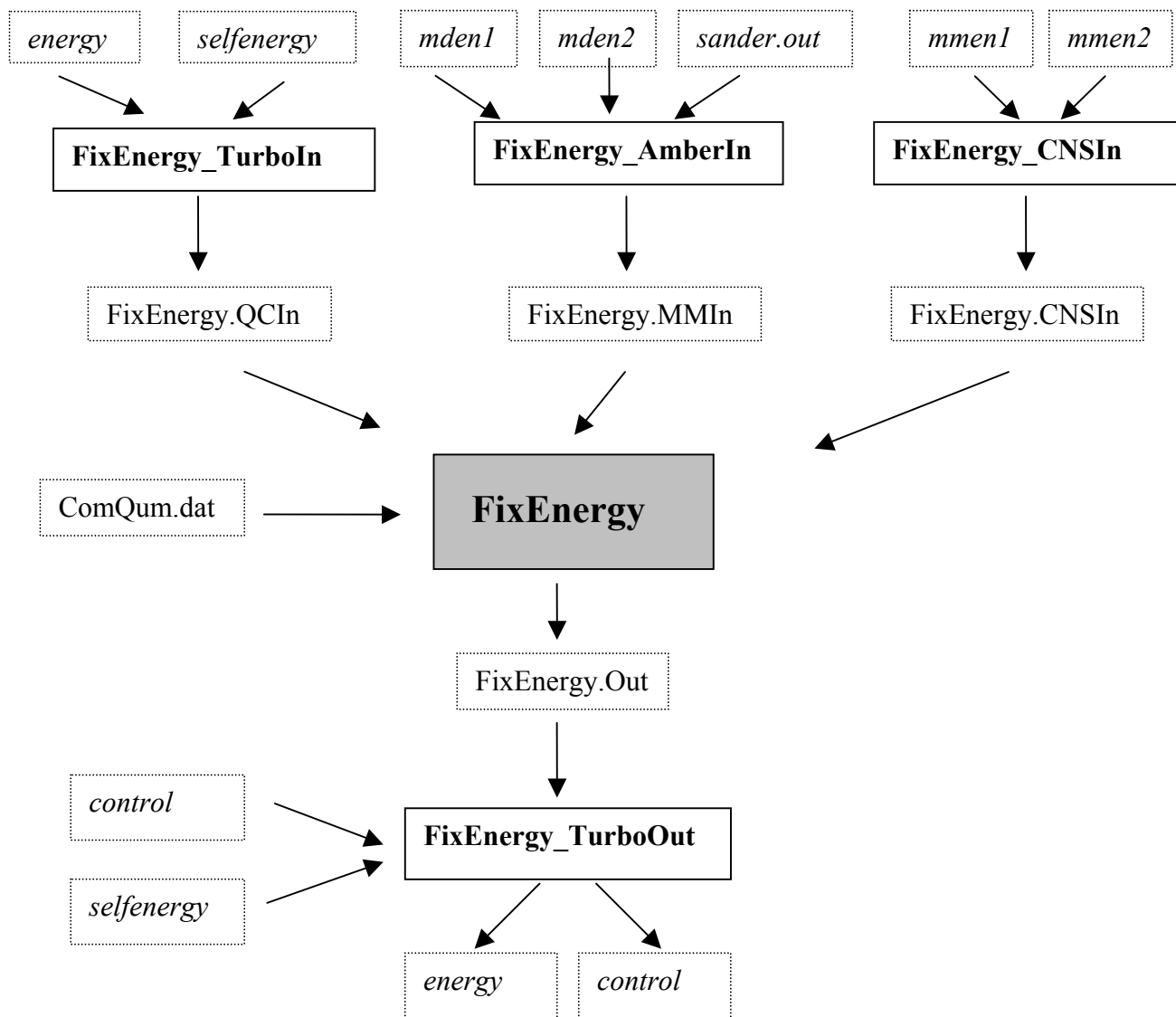


Figure 7. The flow of `FixEnergy` coupling Turbomole with either Amber or CNS.

2.5.1 The format of the files read and written by the general FixEnergy program.

The program FixEnergy reads the QC energy from the file FixEnergy.QCIn and the MM energy of system 1, system 2, and system 3 from FixEnergy.MMIn in ComQum-01 and FixEnergy.CNSIn in ComQum-X. After adding the QC and MM energies appropriately, the QC/MM energy is written to the text file FixEnergy.Out. Below there is a description of the formats of the input and output files of FixEnergy.

- **FixEnergy.QCIn**

The file FixEnergy.QCIn contains the QC energy. After the keyword **\$energy_qc** the energy is written in kJ/mol.

General format and example of FixEnergy.QCIn

<pre>\$energy_qc d \$end</pre>	<pre>\$energy_qc -.12124913427955E+07 \$end</pre>
--------------------------------	---

- **FixEnergy.MMIn/FixEnergy.CNSIn**

The files FixEnergy.MMIn and FixEnergy.CNSIn contain the MM energies of system 1, system 2, and system 3. The energies are written in kJ/mol in the lines following the three keywords, **\$energy_mm1**, **\$energy_mm2**, **\$energy_mm3**. In the example, the energy of system 1, system 2, and system 3 are 16.001 kJ/mol, -37.005, and -2356-596 kJ/mol kJ/mol, respectively.

General format and example of FixEnergy.MMIn and FixEnergy.CNSIn

<pre>\$energy_mm1 d \$energy_mm2 d \$energy_mm3 d \$end</pre>	<pre>\$energy_mm1 0.16014178232088E+02 \$energy_mm2 -.37048966267904E+02 \$energy_mm3 -.23565961600000E+04 \$end</pre>
---	--

- **FixEnergy.Out**

The file FixEnergy.Out contains the QC/MM energy calculated by FixEnergy, and the energy of system 3, which is only needed for the output of ComQum. The QC/MM energy is written after the keyword **\$energy_comqum**, and the MM energy of system 3 is written after the keyword **energy_mm3**. In the example, the QC/MM energy is -1212544.406 kJ/mol, and the MM energy is -2356.595 kJ/mol.

General format and example of FixEnergy.out

<pre>\$energy_comqum d \$energy_mm3 d \$end</pre>	<pre>\$energy_comqum -.12125444059400E+07 \$energy_mm3 -.23565961600000E+04 \$end</pre>
---	---

2.5.2 A short description of the data converting programs for FixEnergy.

- FixEnergy_TurboIn

The program FixEnergy_TurboIn reads the QC energy from the Turbomole output files “energy” and “selfenergy” in Hartrees. The QC energy and the selfenergy are added, converted to kJ/mol and written to the text file FixEnergy.QCIn.

- FixEnergy_AmberIn / FixEnergy_CNSIn

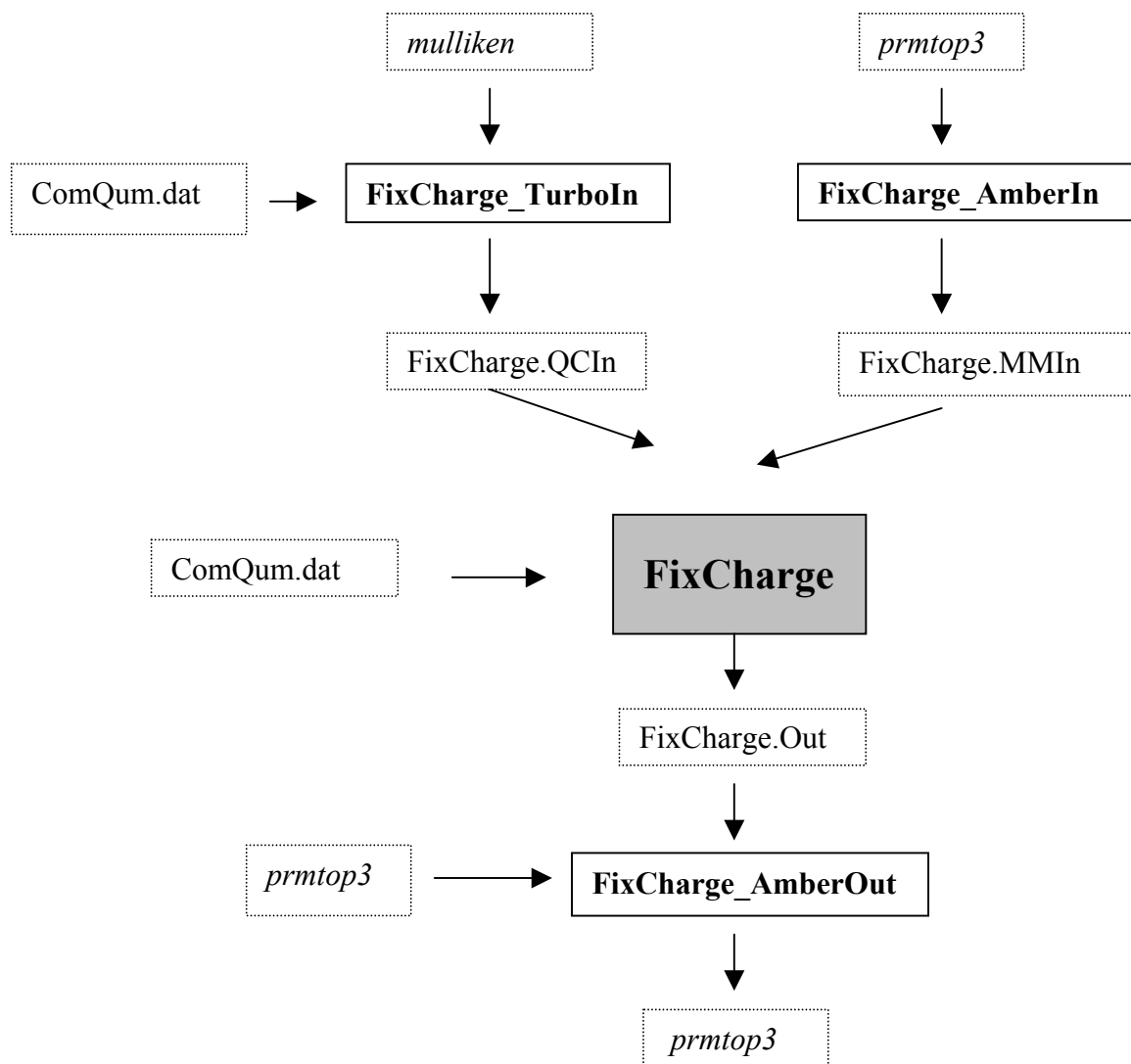
The programs FixEnergy_AmberIn and FixEnergy_CNSIn read the energies of system 1, 2, and 3 from the Amber files “mden1”, “mden3”, and “sander.out” and the CNS files “mmen1”, “mmen2”, and “mmen3”. The energies are converted into kJ/mol and written to the files FixEnergy_MMIn or FixEnergy_CNSIn.

- FixEnergy_TurboOut

The program FixEnergy_TurboOut reads the QC/MM energy from FixEnergy.Out. Reading also the Turbomole files “control” and “selfenergy”, the energy (in Hartrees) and the difference in energy from the previous iteration are written to the Turbomole files “energy” and “control”, respectively.

2.6 FixCharge

In Figure 8, the flow of the programs inserting the charges calculated at the QC level in the MM representation is shown. Note that this operation is only needed in ComQum-01. The general procedures of ComQum performing this operation are found in the gray box in Figure 8.



2.6.1 The format of the files read and written by the general FixCharge program.

The program FixCharge reads the charges calculated at the QC level from the file FixCharge.QCIn, and the old charges in the MM representation from the file FixCharge.MMIn. The new charges are inserted in the MM representation of system 3. This information is written to the text file FixCharge.Out. Below there is a description the format of the input and output files of the program FixCharge.

- **FixCharge.QCIn**

The file FixCharge.QCIn contains the QC charges. After the keyword **\$charge_qc**, an integer gives the number of charges to be read (natom1). Then follows one column with the charges and one column with the atom names. In the example there are five atoms with the names "h","c","h","h", and "s" having charges of 0.1067, -0.3780, 0.0172, 0.0041, and -0.7499.

General format and example of FixCharge.QCIn

```

$charge_qc
i
d(natom1) c4(natom1)
$end

```

```

$charge_qc
5
0.10670000 h
-0.37800000 c
0.01720000 h
0.00410000 h
-0.74990000 s
$end

```

- **FixCharge.MMIn**

FixCharge.MMIn contains the old charges of system 3. After the keyword **\$charge_mm3**, the integer gives the number of charges to be read (natom3). Then follows two columns, one containing the atomic charges and one containing the names of the the corresponding atom. In the example below there are 829 charges, the first one has the value -0.4157 and the name of the corresponding atom is "N", etc. Not all the charges and names of the atoms are shown.

General format and example of FixCharge.QCIn

```

$charge_mm
i
d(natom3) c4(natom3)
$end

```

```

$charge_mm3
829
-0.41569998 N
0.27189999 H
-0.02370000 CA
0.08800000 HA
0.03420000 CB
.
$end

```

- **FixCharge.Out**

The file FixCharge.Out contains the new charges in the MM representation and some information about the amino acid residues. After the keyword **\$charge_mm3**, an integer gives the number of charges to be read (natom3). This integer is followed by one column with the charges in the MM representation and one column with the atom names of the atoms of system 3. There is also the keyword **\$residue_mm3** in the file. After the keyword an integer gives the number of amino acid residues (dimres), which is followed by one column with the names of the residues and the number of the first atom in the residue. In the example below, there are 829 charges shown starting with the charge of the atom with the name "N" having the value -0.4157. Not all the charges are shown in the example. There are 60 amino acid residues; the first has the name "MET" and the first atom in this residue has the number "1". Not all the residues are shown in the example.

General format and example of FixCharge.out

<pre>\$charge_mm3 i d(natom3) c4(natom3) \$residue_mm3 i c4(dimres) i(dimres) \$end</pre>	<pre>\$charge_mm3 829 -0.41569998 N 0.27189999 H . \$residue_mm3 60 MET 1 THR 18 ILE 32 . \$end</pre>
--	--

2.6.2 A short description of the data converting programs for FixCharge.

- FixCharge_TurboIn

The program FixCharge_TurboIn reads the QC charges from the Turbomole file “mulliken”. Furthermore information from the file ComQum.dat is needed to write the charges and names of the QC atoms to the file FixCharge_TurboIn.

- FixCharge_AmberIn

The program FixCharge_AmberIn reads the MM charges of system 3 from the topology file “prmtop3”. The program writes the charges and names of the MM atoms to the file FixCharge.MMIIn.

- FixCharge_AmberOut

The program FixCharge_AmberOut reads the new charges of system 3 from the file FixCharge.Out. Furthermore information from the old file “prmtop3” is needed to write the new “prmtop3” file.

3 Figure for the ComQum-X article.

