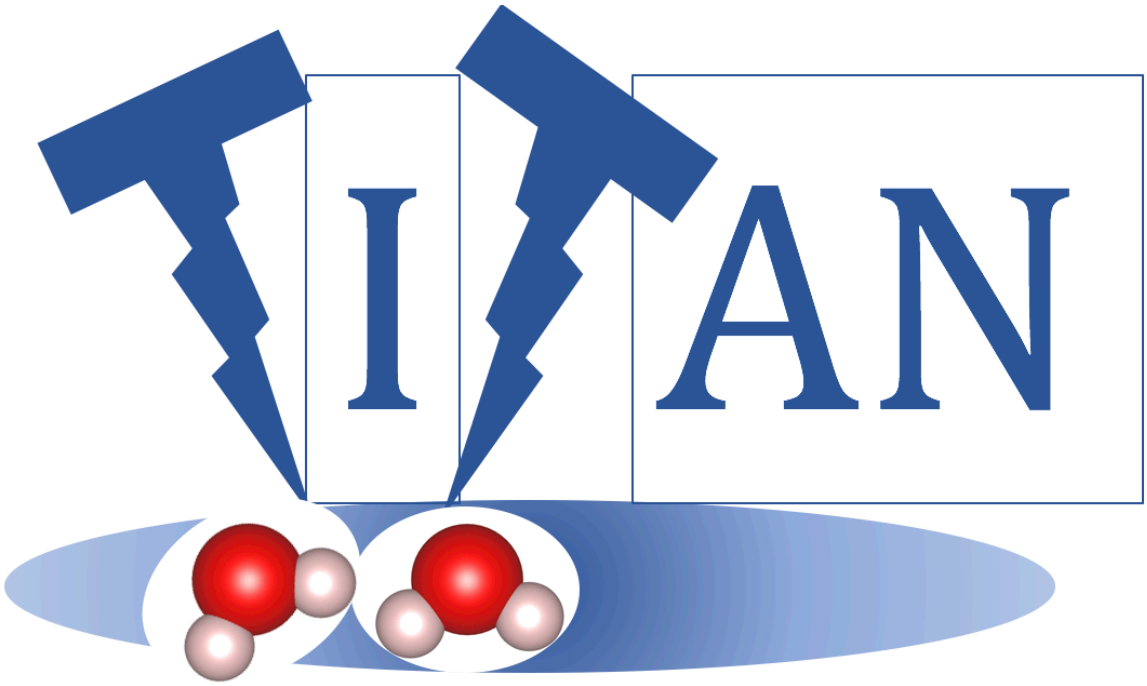

—

v1.2

A code for the generation and manipulation of electric fields

Authors:

Thijs Stuyver
Jing Huang
Dibyendu Mallick
Sason Shaik

21 – July – 2019

Table of Contents

1. Introduction

This section presents an overview of the TITAN package with all its features and acknowledges the funding and collaborators who have contributed to the code.

1.1. What is TITAN?

TITAN is an open-source software package capable of (a) generating external electric fields by creating collections of point charges around chemical systems, and (b) quantifying electric fields at any location based on a charge distribution, e. g. the local electric fields (LEF), caused by the charge distribution in distant regions, experienced at the active site of a (bio)chemical system.

More specifically, two types of point charge distributions can be generated. On the one hand, it is possible to create two circular plate charge distributions on opposite sides of the active site to generate a uniform oriented electric field. On the other hand, one can create a chiral charge distribution in the shape of spiral line with mixed positive and negative charges.

The quantification of electric fields can either start from an existing charge distribution, or alternatively, one can provide the structure of a protein, which is subsequently transformed into a charge distribution.

Additionally, a number of tools are included in the TITAN code, capable to transform coordinates between spatial units and to generate AMBER[1] and CHARMM[2] libraries with non-standard residues.

TITAN is distributed under the conditions of the GPL License version 3 (GPLv3).

1.2. What are the features in the current version?

- Generate a uniform oriented electric field with the help of circular plate charge distributions; output in AMBER or CHARMM format. Alternatively, the output also can be generated in Cartesian coordinates (compatible with GAUSSIAN09 [3] and TURBOMOLE[4]; *vide infra*).
- Generate a chiral (either left- or right-handed) charge distribution in the shape of a spiral line; output in Cartesian coordinates (compatible with GAUSSIAN09 and TURBOMOLE; *vide infra*).

- Calculate the strength of the net electric field at any point for a given charge distribution; input in .txt-format required.
- Calculate the strength of the local electric field present within a protein; a .pdb-format is required as input (the residues in the .pdb-file need to be compatible with either AMBER and CHARMM; non-standard residues can be included in the library with the help of one of the tools included in the package).

1.3. Limitations

The TITAN code does not include any Graphical User Interface (GUI). However, many of the output-files generated can be visualized by freely-available graphic software packages. As an example, the generated .pdb files with the charge distributions can be visualized by either AVOGADRO,[5] CHIMERA,[6] VMD,[7] etc. The code is written in the standard PYTHON language and should thus run smoothly on either LINUX, WINDOWS or MAC computers. Some features may not be working properly with PYTHON3; PYTHON2.6 or PYTHON2.7 are recommended.

1.4. Acknowledgements, contact and citation

TITAN has been developed within the Shaik group at the Hebrew University in Jerusalem, Israel. The code builds on a collection of algorithms which had been developed throughout the years by a variety of former research associates of Prof. Shaik. The first code for generating uniform external electric field was written by Dr. K. B. Cho. Subsequently it was applied by Dr. W. Z. Lai, Dr. C. Hui, and Dr. K. B. Cho to P450cam. Later Dr. D. Usharani and Dr. C. Li made some improvements to these scripts. Dr. H. Hirao wrote a script to calculate local electric fields. These individual scripts have been organized, rewritten and a number of new features have been included; finally resulting in the TITAN code. The main developers of the present version of the code are Thijs Stuyver and Jing Huang. Support throughout the project has been provided by Dibyendu Mallick and Prof. Sason Shaik.

If you wish to make use of the TITAN code in your research project, please include the following citation in the resulting scientific publications:

Thijs Stuyver, Jing Huang, Dibyendu Mallick, Sason Shaik, *J. Comput. Chem.* **2019**, *XX*, XXX-XXX

Questions, suggestions, bugs and more can be sent to either of the following contributors:

- Prof. Sason Shaik: Director, The Lise Meitner-Minerva Center for Computational Quantum Chemistry, Institute of Chemistry, The Hebrew University of Jerusalem, Givat Ram Campus, Jerusalem 91904, ISRAEL

  E-mail: sason.shaik@gmail.com
  FAX: +972-2-6584680
  Phone: +972-2-6585909
  http://yfaat.ch.huji.ac.il/sason.html

- Thijs Stuyver: Post-doctoral research associate within the Shaik group, Institute of Chemistry, The Hebrew University of Jerusalem, Givat Ram Campus, Jerusalem 91904, ISRAEL

  E-mail: Thijs.Stuyver@huji.ac.il

1.5. How to obtain the TITAN code?

The TITAN code is an open-source package distributed by the Shaik group from the Hebrew University of Jerusalem for free. The current version of the package can be downloaded from:

https://titan-code.github.io

2. Installation

Once the code has been downloaded, no specific installation steps are required; the examples can be run from within either of the example directories through the following command:

Python THE-PATH-TO-THE-SRC-DIRECTORY/titan.py TITAN_INPUT_NAME.inp

To facilitate straightforward execution of the code, one can add the path to the src-directory to .bash_profile or .bash_rc.

The tools are stand-alone scripts which can be found in a separate *tools*-directory. The specifics of these tools are outlined in Section 4.

3. Overview of the code

The current version of the TITAN code can perform 3 different types of operations:

- Circular plate charge distribution generation (CPC)
- Spiral line charge distribution generation (SL)
- Quantification of an electric field (QUANT)

For each of these types, a template of the input file can be found in the "input_file_template"-directory of the package. Below, an overview of the different keywords for each of the types of operations is provided. Note that while each operation requires every keyword listed in their respective subsection to be specified (unless indicated otherwise), there is no inherent order required in their listing in the input-file.

3.1. CPC

INPUT:
A .inp-file (template-file = TITAN_CPC.inp)

OUTPUT:

- *"OUTFORMAT = CHARMM"*: NAME.info and NAME.pdb
- *"OUTFORMAT = AMBER"*: NAME.info, NAME.pdb, NAME.frcmod, NAME.lib and leap.in
- *"OUTFORMAT = GAUSSIAN"*: NAME.info and NAME.txt

All these files are combined in a new directory, made within the active directory, and is called *"CHARMM_FORMAT_CPC"*, *"AMBER_FORMAT_CPC"* or *"GAUSSIAN_FORMAT_CPC"* respectively.

KEYWORDS:

- *"TYPE"*: This keyword indicates the type of operation that is requested. For a CPC-operation, this keyword is set to *"CPC"*.

- *"R"*: This keyword indicates the increase in radius between each successive ring in the circular plate charge distribution (cf. Fig. 1). The unit of this keyword is Angstrom.
- *"N"*: This keyword indicates the total number of circular rings in the plate.
- *"DIS"*: This keyword indicates the distance between the center of the plate and the point of at which the electric field should be calculated. The unit of this keyword is Angstrom.
- *"NAME"*: This keyword indicates the name for the output files (*NAME.info*, *NAME.pdb* etc.).
- *"OUTFORMAT"*: This keyword indicates the format of the output
  - *Options: "CHARMM", "AMBER"* or *"GAUSSIAN"*
- *"POINT1_X"*: This keyword indicates the x-coordinate of the first point of the vector along which the electric field will be aligned (e.g. the Fe=O bond in the active site of a P450 enzyme). Furthermore, POINT1 is the point at which the electric field strength is set, *vide infra*.
- *"POINT1_Y"*: This keyword indicates the y-coordinate of the first point of the vector along which the electric field will be aligned.
- *"POINT1_Z"*: This keyword indicates the z-coordinate of the first point of the vector along which the electric field will be aligned.
- *"POINT2_X"*: This keyword indicates the x-coordinate of the second point of the vector along which the electric field will be aligned.
- *"POINT2_Y"*: This keyword indicates the y-coordinate of the second point of the vector along which the electric field will be aligned.
- *"POINT2_Z"*: This keyword indicates the z-coordinate of the second point of the vector along which the electric field will be aligned.
- *"FIELD"*: This keyword indicates the desired field strength (in a. u.) at POINT1.

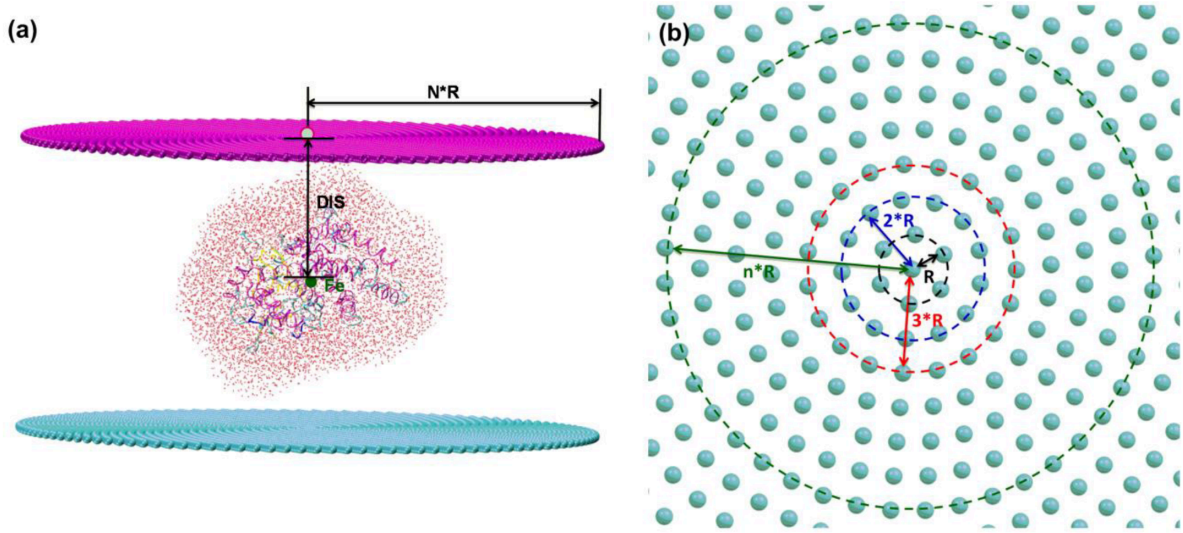

Figure 1. (a) The charges distribution generated in two circular plates with the keywords “R”, “N” and “DIS”; (b) The relationship between “R”, “N” and the position of the point charges in the circular plate.

The template input file for this kind of calculations as it can be found in the "input_file_template"-directory, is shown below:

```
# TYPE OF OPERATION (CPC/SL/QUANT)
TYPE = CPC

# "R" IS THE INCREASE IN RADIUS BETWEEN
# SUCCESIVE CIRCULAR RINGS IN THE PLATE (ANGSTROM)
R = 2.8

# "N" IS THE NUMBER OF CIRCULAR RINGS IN THE PLATE
N = 33

# THE DISTANCE BETWEEN THE CENTRE OF THE PLATE AND THE POINT OF CHOICE
# (IN ANGSTROM)
DIS = 46.8279

# THE NAME FOR THE OUTPUT FILES
# FOR EXAMPLE, IF WE SET THE “NAME” KEYWORD TO "chrg",
# IT MEANS THAT THE OUTPUT WILL BE chrg.info AND chrg.pdb
NAME = chrg

# THE FORMAT OF THE OUTPUT (“CHARMM”, “AMBER”, or “GAUSSIAN”)
OUTFORMAT = AMBER
```

```
# DEFINE THE COORDINATES OF THE VECTOR (POINT1 -> POINT2)
# ALONG WHICH THE ELECTRIC FIELD HAS TO BE ALIGNED
# FOR EXAMPLE, THE Fe=O VECTOR IN THE ACTIVE SITE OF A P450 ENZYME

POINT1_X = 0.000
POINT1_Y = 0.000
POINT1_Z = 0.000

POINT2_X = 5.000
POINT2_Y = 9.000
POINT2_Z = 6.000

# THE FIELD STRENGTH DESIRED AT POINT1 (a.u.)
FIELD = 0.0025
```

3.2. SL

INPUT:

A .inp-file (template-file = TITAN_SL.inp)

OUTPUT:

NAME.txt, NAME.pdb and NAME.info. All these files are combined in new directory, made within the active directory, and is called *"GAUSS_FORMAT_CPC"*.

KEYWORDS:

- *"TYPE"*: This keyword indicates the type of operation that is requested. For an SL-operation, this keyword is set to *"SL"*.
- *"RADIUS"*: This keyword indicates the increase in radius between each successive ring in the circular plate charge distribution (cf. Fig. 2). The unit of this keyword is Angstrom.
- *"STEP"*: This keyword indicates the pitch between two neighboring atoms (cf. Fig. 2). The unit of this keyword is in Angstrom.
- *"N"*: This keyword indicates the total numbers of atoms included in one full circle.
- *"NAME"*: This keyword indicates the name for the output files (*NAME.info, NAME.pdb* etc.).
- *"XP"*: This keyword indicates the x-coordinate of the point at which the electric field will be calculated.
- *"YP"*: This keyword indicates the y-coordinate of the point at which the electric field will be calculated.
- *"ZP"*: This keyword indicates the z-coordinate of the point at which the electric field will be calculated.
- *"CHIRALITY"*: This keyword indicates the chirality of the spiral line.
  - *Options: "RIGHT-HAND" or "LEFT-HAND"*
- *"NAME"*: This keyword indicates the name for the output files (*NAME.info*, *NAME.txt* and *NAME.pdb*).
- *"SEQUENCE"*: This keyword indicates the sequence of the charges in the spiral line. The sequence has to consist of a succession of *"P"* and *"N"* fragments (the number of *"P"* and

"N" fragments should be equal in order to make sure that the net charge of the spiral line is zero).

- *"FATOM"*: This keyword indicates the number of dummy atoms included in a single *"P"* or *"N"* fragment.
- *"CHARGE"*: This keyword indicates the charge of each dummy atom.

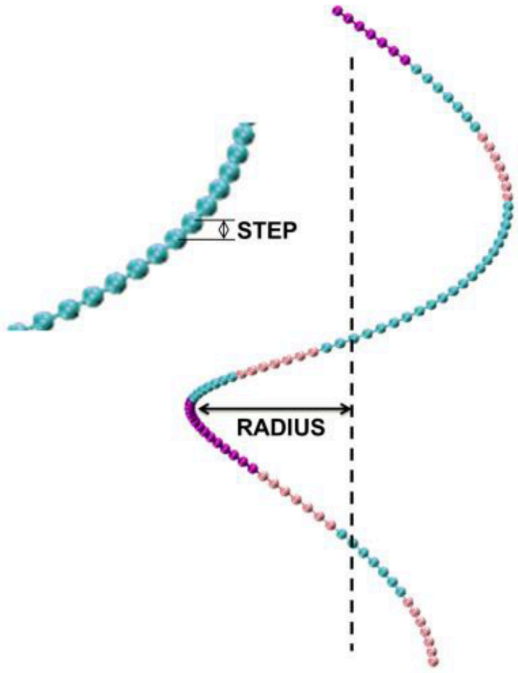

Figure 2. The point charges in the shape of a spiral line.

The template input file for this kind of calculations as it can be found in the "input_file_template"-directory, is shown below:

```
# TYPE OF OPERATION (CPC/SL/QUANT)
TYPE = SL

# THE RADIUS OF SPIRAL LINE (ANGSTROM)

RADIUS = 10.0

# THE PITCH BETWEEN TWO NEIGHBORING ATOMS (ANGSTROM)

STEP = 0.01
```

```
# THE NUMBER OF ATOMS INCLUDE IN ONE CIRCLE

N = 700

# THE POSITION FOR THE ELECTRIC FIELD CALCULATION
# THE ELECTRIC FIELD WILL BE OBTAINED AT THE POINT (XP,YP,ZP)

XP = 0.00
YP = 0.00
ZP = 0.00

# CHIRALITY OF THE SPIRAL LINE

#   CHIRALITY   |   NOTE
# ---------------------------------------
#  RIGHT-HAND   |  right-handed helices
#   LEFT-HAND   |  left-handed helices
# ---------------------------------------

CHIRALITY = RIGHT-HAND

# THE NAME FOR OUTPUT FILES
# FOR EXAMPLE, IF WE SET THE “NAME” KEYWORD TO "chrg",
# IT MEANS THE OUTPUT WOULD BE chrg.txt, chrg.info AND chrg.pdb

NAME = chrg

# THE "SEQUENCE" IS DEFINED AS A SUCCESSION OF
# POSITIVE AND NEGATIVE CHARGE FRAGMENTS, WITH EACH FRAGMENT
# REPRESENTED BY "P" AND "N" RESPECTIVELY. THE NUMBER OF "P" AND "N"
# SHOULD BE EQUAL IN ORDER TO MAKE SURE THE NET CHARGE OF THE SPIRAL LINE IS
ZERO.
# "FATOM" CORRESPONDS TO THE NUMBER OF POINTS IN EACH FRAGMENT.
# "CHARGE" CORRESPONDS TO THE CHARGE OF EACH DUMMY ATOM.

SEQUENCE =
PPNNPPNPNPNPNNPPNNPNNPPNNPNPNPNPPPPPNPNNPNPNNPNNPNPPNNPPNPNPNPN
NPPNNPNNPPNNPNPNPNPPPPPNPNNPNPNNPNNPNPPNNPPNPNPNPNNPPNNPNNPPNN
PNPNPNPNPPPPPNPNNPNPNNPNNPNPPNNPPNPNPNPNNPPNNPNNPPNNPNPNPNPPPPP
NPNNPNPNNPNNPNPPNNPPNPNPNPNNPPNNPNNPPNNPNPNPNPPPPPNPNNPNPNNPNN
PNPPNNPPNPNPNPNNPPNNPNNPPNNPNPNPNPPPPPNPNNPNPNNPNNPNPPNNPPNPNPN
PNNPPNNPNNPPNNPNPNPNPNPPPPPNPNNPNPNNPNNPN
```

3.3. QUANT

INPUT:

A .inp-file (template-file = TITAN_SL.inp), together with either a .txt file (in case *"FILE = TXT"*), or a .pdb file (in case *"FILE = PDB"*) as starting point for the electric field calculation,[1] a DIRECTION_FILE.txt file (in case *"DIRECTION = SELECT"*).

OUTPUT:

NAME.ef (contains the output of the electric field calculation), NAME.save (saves the part of the charge distribution which has been selected for the electric field calculation from the charge distribution .txt-file), NAME_CHARGE_DISTRIBUTION.txt (in case *"FILE = PDB"*; this file contains the charge distribution resulting from the .pdb file), NAME_sequence.save (in case *"FILE = PDB"*; this file contains the list of the selected atoms with their coordinates and assigned charge, i.e. all atoms in case *"ATOM_SELECT = ALL"* and the selected set of atoms in case *"ATOM_SELECT = PART"*).

KEYWORDS:

- *"TYPE"*: This keyword indicates the type of operation that is requested. For a QUANT-operation, this keyword is set to *"QUANT"*.
- *"FILE"*: This keyword indicates the starting point of the electric field calculation.
  - *Options: "TXT" or "PDB"*
- *"NAME"*: This keyword indicates the name of the input file containing the starting point of the electric field calculation (either *NAME.txt* or *NAME.pdb*).
- *"CHARGE_SELECT"*: This keyword indicates whether all the point charges in the charge distribution are taken into account during the electric field quantification or not (in case *"CHARGE_SELECT = PART"*, the keyword *"CHARGE_SEQ"* has to be specified as well).

[1] the MM region, i.e. the protein matrix, is generally modified close to the link atoms, which impacts the corresponding charge distribution. As such, in order to probe the actual role played by the electrostatics of a protein matrix on the active site in QM/MM calculations, it is advisable to quantify the electric field starting from this modified charge distribution so as to retain full consistency. ChemShell produces this modified point charge distribution in a format that is compatible with the charge distribution .txt-input of TITAN.

- *"CHARGE_SEQ"*: This keyword indicates which charges are taken into account during the electric field quantification. The sequence is defined as a combination of ranges, denoted by "*R(begin,end)"* (e.g. *"R(3,10)"*), and points, denoted by *"P(number)"* (e.g. *"P(300)"*). Consult the templates below for a full example of such a sequence-string.
- *"DIRECTION"*: This keyword indicates whether the direction vector V, along which the oriented electric field is calculated, will be introduced manually or by selecting atoms in an additional input file.
  - *Options: "SELECT" or "MANUAL"*
- *"V1X"*: This keyword indicates the x-coordinate of the first point of the vector V along which the oriented electric field will be calculated (e.g. the Fe=O bond in the active site of a P450 enzyme). This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"V1Y"*: This keyword indicates the y-coordinate of the first point of the vector V along which the oriented electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"V1Z"*: This keyword indicates the z-coordinate of the first point of the vector V along which the oriented electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"V2X"*: This keyword indicates the x-coordinate of the first point of the vector V along which the oriented electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"V2Y"*: This keyword indicates the y-coordinate of the first point of the vector V along which the oriented electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"V2Z"*: This keyword indicates the z-coordinate of the first point of the vector V along which the oriented electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"XP"*: This keyword indicates the x-coordinate of the point at which the electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"YP"*: This keyword indicates the y-coordinate of the point at which the electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.
- *"ZP"*: This keyword indicates the z-coordinate of the point at which the electric field will be calculated. This keyword is only relevant in case *"DIRECTION = MANUAL"*.

- *"DIRECTION_FILE"*: This keyword indicates the coordinate .txt-file from which the parameters for the direction vector V and the location for the electric field evaluation will be read. This keyword is only relevant in case *"DIRECTION = SELECT"*.
- *"ATOM1"*: This keyword indicates the number of the first atom defining vector V: ATOM1(V1X,V1Y,V1Z). This keyword is only relevant in case *"DIRECTION = SELECT"*.
- *"ATOM2"*: This keyword indicates the number of the second atom defining vector V: ATOM2(V2X,V2Y,V2Z). This keyword is only relevant in case *"DIRECTION = SELECT"*.
- *"ATOM_CENTER"*: This keyword indicates the number of the atom defining the location at which the electric field will be calculated: ATOM_CENTER(XP,YP,ZP). This keyword is only relevant in case *"DIRECTION = SELECT"*.
- *"NAME_CHARGE_DISTRIBUTION"*: This keyword indicates the name of the charge distribution (NAME_CHARGE_DISTRIBUTION..txt-file) which will be generated from the .pdb-file. This keyword is only relevant in case *"FILE = PDB"*.
- *"ATOM_SELECT"*: This keyword indicates whether all the atoms in the protein matrix are taken into account during the charge distribution construction or not (in case *"ATOM_SELECT = PART"*, the keyword *"ATOM_SEQ"* has to be specified as well). This keyword is only relevant in case *"FILE = PDB"*.
- *"ATOM_SEQ"*: This keyword indicates which atoms are taken into account during the charge distribution construction. The sequence is defined as a combination of ranges, denoted by "*R(begin,end)"* (e.g. *"R(3,10)"*), and points, denoted by *"P(number)"* (e.g. *"P(300)"*). Consult the templates below for a full example of such a sequence-string. This keyword is only relevant in case *"FILE = PDB"*.
- *"FORCE"*: This keyword indicates the force field of choice which will be used to generate the charge distribution from the .pdb-file. This keyword is only relevant in case *"FILE = PDB"*.
  - *Options: "AMBER" or "CHARMM"*
- *"N_TERMINAL"*: This keyword indicates the number of the residue in the .pdb-file which corresponds to the N-terminal of the peptide. This keyword is only relevant in case *"FILE = PDB"*.
- *"C_TERMINAL"*: This keyword indicates the number of the residue in the .pdb-file which corresponds to the C-terminal of the peptide. This keyword is only relevant in case *"FILE = PDB"*.

- *"ASPP"*: This keyword indicates the numbers of the residues in the .pdb-file which corresponds to protonated ASP-residues. This keyword is only relevant in case *"FILE = PDB"* and *"FORCE = CHARMM"*.
- *"GLUP"*: This keyword indicates the numbers of the residues in the .pdb-file which corresponds to protonated GLU-residues. This keyword is only relevant in case *"FILE = PDB"* and *"FORCE = CHARMM"*.
- *"DISU"*: This keyword indicates the numbers of the residues in the .pdb-file which corresponds to CYS-residues involved in a disulfide bridge, i.e. unprotonated CYS-residues. This keyword is only relevant in case *"FILE = PDB"* and *"FORCE = CHARMM"*.

The complete template input file for this kind of operation, as it can be found in the "input_file_template"-directory, is shown below:

```
# TYPE OF OPERATION (CPC/SL/QUANT)
TYPE = QUANT

# INPUT FILE TYPE: TXT FILE (TXT) OR PDB FILE (PDB) AS STARTING POINT
FILE = PDB

# INPUT UNIT (BOHR OR ANS)
UNIT = ANS

# THE CHARGE DISTRIBUTION WILL BE READ FROM NAME.txt OR NAME.pdb
# THE RESULT WILL BE WRITTEN IN NAME.ef

NAME = opt

#  CHARGE_SELECT    |                NOTE
#-------------------------------------------------------------------
#     "ALL"       | ALL THE POINT CHARGES IN THE CHARGE DISTRIBUTION
#                 | ARE SELECTED FOR THE ELECTRIC FIELD CALCULATION.
#-------------------------------------------------------------------
#     "PART"      | A PART OF CHARGES ARE SELECTED FROM THE CHARGE DISTRIBUTION.
#                 | (IN THIS CASE, THE "CHARGE_SEQ" KEYWORD NEEDS TO BE DEFINED)
#-------------------------------------------------------------------

CHARGE_SELECT = ALL

# THE “CHARGE_SEQ" KEYWORD IS USED TO SELECT THE POINT CHARGES FOR
CHARGE_SELECT = "PART"
```

IT IS NOT NECESSARY TO SET THE "CHARGE_SEQ" KEYWORD WHEN CHARGE_SELECT = "ALL"
FOR EXAMPLE, ”CHARGE_SEQ = R(3,10)+P(20)+P(3000)+R(400,403)+P(50)" MEANS:
THE POINT CHARGES (PC) FROM NO. 3 TO NO. 10, THE PC NO. 20, THE PC NO. 3000, THE PC FROM
NO. 400 TO NO. 403 AND PC NO. 50 ARE SELECTED FOR THE EF CALCULATIONS.

CHARGE_SEQ = R(3,10)+P(20)+R(400,403)+P(50)

THE DIRECTION VECTOR V CAN BE DEFINED IN TWO WAYS: EITHER BY SELECTING ATOMS IN A COORDINATE FILE (SELECT), OR BY INTRODUCING THE COORDINATES MANUALY (MANUAL)

DIRECTION = SELECT

#_____
____
THE FOLLOWING PARAMETERS ARE ONLY RELEVANT IN CASE DIRECTION = “MANUAL”

DIRECTION VECTOR V
V = (V2X-V1X, V2Y-V1Y, V2Z-V1Z)
THE VALUE OF THE EF ALIGNED WITH THIS VECTOR IS CALCULATED.

V1X = 0.0000
V1Y = 0.0000
V1Z = 0.0000
V2X = 0.0000
V2Y = 0.0000
V2Z = 3.1010

THE VALUE OF THE EF IS CALCULATED AT (XP,YP,ZP).

XP = 0.0000
YP = 0.0000
ZP = 0.0000

#_____
____
THE FOLLOWING PARAMETERS ARE ONLY RELEVANT IN CASE DIRECTION = “SELECT”

DIRECTION_FILE = heme

DIRECTION VECTOR V
V = (V2X-V1X, V2Y-V1Y, V2Z-V1Z)
THE NUMBER OF THE ATOMS DETERMINING V: ATOM1(V1X, V1Y, V1Z) AND ATOM2(V2X, V2Y, V2Z)

```
# SELECTED FROM DIRECTION_FILE
ATOM1 = 2
ATOM2 = 1

# THE VALUE OF THE EF IS CALCULATED AT (XP,YP,ZP).
# THE NUMBER OF THE ATOM DETERMINING THIS POINT SELECTED FROM DIRECTION_FILE
(ATOM_CENTER)

ATOM_CENTER = 2

_____
_____
# THE FOLLOWING PARAMETERS ARE ONLY RELEVANT IN CASE FILE = “PDB”

# NAME OF THE CHARGE DISTRIBUTION GENERATED (NAME_CHARGE_DISTRIBUTION.txt BY
DEFAULT).

NAME_CHARGE_DISTRIBUTION = opt

#  ATOM_SELECT   |                NOTE
#-------------------------------------------------------------------
#     "ALL"       | ALL THE ATOMS IN THE PDB-FILE
#                 | ARE SELECTED FOR THE ELECTRIC FIELD CALCULATION.
#-------------------------------------------------------------------
#     "PART"      | A PART OF THE ATOMS IN THE PDB-FILE ARE SELECTED.
#                 | (IN THIS CASE, THE “ATOM_SEQ" KEYWORD NEEDS TO BE DEFINED)
#-------------------------------------------------------------------

ATOM_SELECT = PART

# THE “ATOM_SEQ" KEYWORD IS USED TO SELECT THE ATOMS FOR ATOM_SELECT = "PART"
# IT IS NOT NECESSARY TO SET THE “ATOM_SEQ" KEYWORD WHEN ATOM_SELECT = "ALL"
# FOR EXAMPLE, ”ATOM_SEQ = R(3,10)+P(20)+P(3000)+R(400,403)+P(50)" MEANS:
# THE ATOMS FROM NO. 3 TO NO. 10, THE ATOM NO. 20, THE ATOM NO. 3000, THE ATOM
FROM
# NO. 400 TO NO. 403 AND ATOM NO. 50 ARE SELECTED FOR THE EF CALCULATIONS.

ATOM_SEQ = R(2,20)+P(35)

# FORCE FIELD OF CHOICE (AMBER/CHARMM)
FORCE = CHARMM

# THE RESIDUE NUMBER OF N TERMINAL AND C TERMINAL

# FOR EXAMPLE:
# IF NO. 4 RESIDUE IS THE N-TERMINAL OF THE PEPTIDE, PLEASE SET " N_TERMINAL = 4 "
```

USE THE COMMAND: " grep "HT1" PDB_FILE   " TO COMFIRM THE RESIDUE NUMBER OF N-TERMINAL.
IF NO. 500 RESIDUE IS THE C-TERMINAL OF THE PEPTIDE, PLEASE SET " C_TERMINAL = 500 "
USE THE COMMAND: " grep "OT1" PDB_FILE   " TO COMFIRM THE RESIDUE NUMBER OF C-TERMINAL.

N_TERMINAL = 4
C_TERMINAL = 422

THE FOLLOWING PARAMETERS ARE ONLY RELEVANT IN CASE FILE = “PDB” & FORCE = “CHARMM”

THE RESIDUE NUMBER OF ASPP, GLUP AND DISU

IF THE RESIDUES OF THE PROTONATED ASP ARE E.G. NO. 235, 246 RESIDUES, PLEASE SET " ASPP = 235,246 "
USE THE COMMAND: " grep "HD2 ASP" PDB_FILE   " TO COMFIRM THE RESIDUE NUMBER OF PRONONATED ASP.
IF THE RESIDUES OF THE PROTONATED GLU ARE E.G. NO. 250, 266 RESIDUES, PLEASE SET " GLUP = 250,266 "
USE THE COMMAND: " grep "HE2 GLU" PDB_FILE   " TO COMFIRM THE RESIDUE NUMBER OF PRONONATED GLU.
IF NO. 300 CYS RESIDUE IS BONDED WITH NO. 340 CYS RESIDUE THROUGH A DISULFIDE BOND,
THEN NO. 300 AND NO. 340 CYS ARE NOT PROTONATED. IN THIS CASE, PLEASE SET " DISU = 300,340 "
USE THE COMMAND: " grep "SG CYS" PDB_FILE   " AND " grep "HG1 CYS" PDB_FILE   "
TO COMFIRM THE RESIDUE NUMBER OF UNPRONONATED CYS.

ASPP = 350,398
GLUP = 53,417
DISU = 365

4. Tools

Currently, four tools are included within the TITAN code:

- ans2bohr.py
- bohr2ans.py
- generate_amber_library.py
- generate_charmm_library.py

The former 2 tools are scripts to convert the coordinates in a charge distribution (.txt) file from one unit of length to the other and vice versa. The latter two tools are scripts to construct alternative AMBER/CHARMM libraries with non-standard residues, which can be used as replacements for the regular libraries included in the main code of TITAN by default.

The coordinate conversion tools can be executed through the following commands:

Python THE-PATH-TO-THE-SRC-DIRECTORY/ans2bohr.py NAME.txt

Python THE-PATH-TO-THE-SRC-DIRECTORY/bohr2ans.py NAME.txt

The library generation scripts are required to be present in a directory together with three different input files. For AMBER, these files are respectively: *C_terminal.lib*, *N_terminal.lib*, *top_amino94.lib*. For CHARMM, these files are respectively: *C_terminal.txt, N_terminal.txt, top_protein.txt* (see the tools directory for example directories). Subsequently, the scripts can be executed within this directory through the following commands:

Python generate_charmm_library.py

Python generate_amber_library.py

In order to activate these generated alternative libraries, the original *library_amber.py/ library_charmm.py* files in the *src* directory need to be replaced by these alternative libraries. The alternative libraries should be given the exact same name as the original library-files present in the *src* directory.

IMPORTANT NOTE. In the current input files for library generation, a number of non-standard residues have already been appended at the end of the files. Additional non-standard residues can be introduced by adhering to the same pattern. Note that the final entries in *top_amino94.lib* and *top_protein.txt* are currently the *"HPC"* and *"HNC"* dummy atoms associated to CPC/SL operations (cf. Sections 3.1 and 3.2). Having these dummy atoms in the CHARMM- or AMBER-library can be useful in case one wants to calculate the combined effect of both a local electric field and an external electric field starting from a .pdb-input file in which both the protein and the generated circular plate (or spiral line) are present. However, the charges associated to these dummy atoms are subject to change, depending on the settings in the CPC/SL input files. As such, one needs to make sure to update the charges for *"HPC"* and *"HNC"* set in *top_amino94.lib* and *top_protein.txt* everytime one wants to quantify an electric field starting from a .pdb-file containing dummy atoms.

5. References

[1] Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M. Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. *J. Comput. Chem.* **2005**, *26*, 1668-1668.

[2] Brooks, B. R., Brooks III, C. L., Mackerell Jr, A. D., Nilsson, L., Petrella, R. J., Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S. et al. *J. Comput. Chem.* **2009**, *30,* 1545-1614.

[3] Frish, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A. et al. *Gaussian 09, revision D.01*; Gaussian, Inc.: Wallingford, CT, 2013.

[4] Ahlrichs, R.; Bär, M.; Häser, M.; Horn, H.; Kölmel, C. *Chem. Phys. Lett.* **1989**, *162*, 165-169.

[5] Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R. *J. Cheminformatics* **2012**, *4*, 17.

[6] Pettersen, E. F.: Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. *J. Comput. Chem.* **2004**, *25*, 1605-1612.

[7] Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graph.* **1996**, *14*, 33-38.