EASYPARM

User manual

Version 3.0 – January 2025

Abdelazim M. A. Abdelgawwad

abdelazim.abdelgawwad@uv.es

Institut de Ciència Molecular, Universitat de València, P.O. Box 22085, València 46071, Spain

Table of Contents

1. Introduction

easyPARM is a computational tool designed with four key functionalities:

1- **Force Field Parameter Derivation for Metal-Containing Systems**
easyPARM derives bond-stretching and bond-angle bending force field parameters for metal-containing molecular systems using the Seminario method.[1] This ensures accurate parameter generation tailored for the AMBER software suite[2], based on the Hessian matrix produced by frequency calculation in Gaussian[3], Orca[4] or Gamess[5] Software.

2- **Force Field Parameterization for Metalloproteins**
The tool facilitates the derivation of bond-stretching and bond-angle bending force field parameters for metalloproteins using the Seminario method. This capability streamlines the integration of these parameters into protein force fields.

3- **Amber Format Conversion to GROMACS or OpenMM.**
easyPARM utilizes ParmEd[6] to convert AMBER-format files into GROMACS or OpenMM formats, facilitating compatibility with other molecular dynamics engines.

4- **Charge Restraints with REsP Fitting**
easyPARM supports the application of charge restraints on specific atoms using the Restrained Electrostatic Potential (REsP) fitting approach.[7] This helps achieve a more accurate electrostatic potential around the molecule while keeping charge distribution physically meaningful.

With these capabilities, easyPARM is an efficient tool for accurately modeling metal-containing systems, ensuring both structural and electrostatic precision.

The easyPARM code was written by Abdelazim M. A. Abdelgawwad under supervision Dr. Antonio Francés-Monerris. This code version 3.00 is free: you can redistribute it and/or modify it under the GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999.

1.1 Key Features

- Generates parameters for metal-containing systems and metalloprotein systems using a combination of the general AMBER force field (GAFF) or the AMBER force field and the missing parameters for bond stretching and angle bending involving metals and linked atoms with Seminario method.
- Capable of generating parameters for entire structures in non-metal systems using either GAFF or AMBER
- Outputs essential files (frcmod, pdb, mol2, and lib) for initiating simulations or preparing metal-containing systems

1.2 Citation

Please cite the following references if you use the easyPARM code:

- Abdelgawwad AMA, Francés-Monerris A. easyPARM: Automated, Versatile, and Reliable Force Field Parameters for Metal-Containing Molecules with Unique Labeling of Coordinating Atoms. ChemRxiv. 2024; doi:10.26434/chemrxiv-2024-f8wp4
- Github (https://github.com/Abdelazim-Abdelgawwad/easyPARM.git)

2. Installation

2.1 Prerequisites

To successfully install and run easyPARM, ensure that the following prerequisites are met:

- Python 3 or higher: easyPARM requires Python version 3 or above to function properly. You can download the latest version of Python from the official Python website.
- periodictable package: This package provides access to the periodic table of elements, which is essential for the metal-specific calculations in easyPARM. To install it, use the following command in your terminal or command prompt:

```
pip install periodictable
```

- scipy package: SciPy is a scientific computing library that easyPARM utilizes for various numerical calculations, including those related to force fields and molecular mechanics. To install SciPy, use the following command:

```
pip install scipy
```

Additional package requirements may arise during execution such as biopython. If prompted, install them via pip or contact us for assistance.

Once the necessary dependencies are installed, you'll be ready to run easyPARM to generate the required force field parameters.

2.2 Running easyPARM

Choose one of the following methods:

1. **Run the script directly**: Navigate to the directory containing the easyPARM script and execute the following command.

```
./easyPARM.sh
```

2. **Set up an alias for easy access**: To run easyPARM from any location in your terminal, you can create an alias by adding the following line to your .bashrc file:

```
alias easyPARM='/full/path/to/easyPARM.sh'
```

3. Usage Guide

3.1 Required Input Files

1. **Hessian Matrix Files**: The Hessian matrix is required for subsequent calculations and can be provided in one of the following formats:
    a. **Checkpoint File (.chk):** Contains the Hessian matrix from a Gaussian frequency calculation.
    b. **Formatted Checkpoint file (.fchk):** Contains the Hessian matrix from a Gaussian frequency calculation.

c. **Gaussian ouput (.log):** Contains the Hessian matrix from a single-point frequency calculation. Ensure that the keyword `iop(7/33=1)` is included in the input.
d. **Orca output (.hess):** Contains the Hessian matrix from a single-point frequency calculation in Orca.
e. **Gamess output (.dat):** Contains the Hessian matrix from a single-point frequency calculation in Gamess.

2. **Optimized Structure (XYZ Format):** Contains the optimized molecular geometry.
3. **Charge Output:** This file provides information on charge distribution (ESP charges) and can be in one of the following formats:
   a- Gaussian Charge Output (RESP charges)
   b- Orca Charge Output (CHELPG charges)
   c- Orca Charge Output (RESP charges)
   d- Gamess Charge Output (RESP charges)
   e- Gamess Charge Output (Gamess Charges)

3.2 Step-by-Step Tutorial

easyPARM Menu

When you run **easyPARM**, you will be prompted to select one of its core functionalities. The following menu will appear:

```
================================
      easyPARM Menu
================================
Select your option:
1- Generate molecular complex parameters
2- Generate metalloprotein .xyz stucture
3- Convert AMBER parameters to OpenMM or GROMACS format
Enter your choice:
```

- **Option 1**:
  Select this option to generate force field parameters for your system. This applies to:
  1. Metal-containing complexes
  2. Metalloproteins
  3. Non-metal complexes

  After selecting this option, you will be guided to provide the required input files and parameters for the generation process.
- **Option 2:**
  This option is crucial for metalloprotein systems. It prepares the `.xyz` file of the system, which is needed for:
  1. Geometry optimization
  2. Frequency calculations
  3. Charge calculations
- **Option 3:**
  Use this option to convert parameters generated in the AMBER format into either OpenMM or GROMACS formats. This functionality facilitates compatibility with

other molecular dynamics engines. Once selected, you will need to specify the AMBER files to be converted and choose the target format.

1. Option 1: Generate Molecular Complex Parameters

1.1 AMBER Configuration

When you select Option 1 from the easyPARM menu, the tool will prompt you to configure your AMBER environment. You will see the following menu:

```
================================
 Amber Configuration Menu
================================
Select your option:
1- Use currently loaded Amber
2- Specify Amber installation path
Enter your choice:
```

- Option 1: Use Currently Loaded AMBER
  If AMBER is already loaded into your environment (e.g., using the module load amber command), select Option 1. The following message will confirm the selection:

  Amber is assumed to be already loaded. Skipping sourcing.

- Option 2: Specify AMBER Installation Path
  if AMBER is not preloaded, select **Option 2** to manually provide the installation path. After selecting this option, the tool will ask you to specify the path:

  Please provide the path for Amber:

  Enter the full directory path where AMBER is installed (e.g., /path/to/amber).

1.2 System Charge.

The code needs the total charge of the system.

Please provide the total charge:

1.3 Providing the Optimized Structure for Seminario Method

To proceed with the Seminario method, you will need to provide the optimized XYZ geometry file. The code will prompt you to enter the name of this file:

Please provide the optimized XYZ geometry file:

1.4 Charge Method

The code supports various types of charge calculation methods. You can choose the method that best fits your needs and system requirements. During execution, the code will prompt you to make a selection from the following menu:

```
================================
 Charge Method Selection Menu
================================
Select the charge calculation method:
1- GAUSSIAN (RESP charges)
2- ORCA (CHELPG charges)
3- ORCA (RESP Charges)
```

```
4- GAMESS (RESP Charges)
5- GAMESS (GAMESS Fit Charges)
Enter your choice:
```

Option 1: Gaussian (RESP charges)

If you select this option, the code will proceed to generate the .mol2 file using **RESP** charges from the Gaussian output. (Refer to the example input in the Appendix for further details).

- **mol2 File Generation**

  After specifying the total charge, the code will prompt you to choose the input format for generating the .mol2 file. You will be presented with the following options:

```
================================
 Charge Output Selection Menu
================================
Please select the input format:
1- Gaussian Output (.log file)
2- Resp (.gesp file)
Enter your choice:
```

  Next, you'll need to select the charge calculation method for the system. The recommended method is **RESP**. The menu will appear as:

```
================================
 Charge Method Selection Menu
================================
Please select the charge method (recommended: RESP):
1- RESP (resp)
2- Mulliken (mul)
3- ESP (esp)
4- AM1-BCC (bcc)
Enter your choice:
```

  You will also be asked to specify the atom type for the system. The options are **GAFF** or **AMBER** or **GAFF2**:

```
================================
 Atom Type Selection Menu
================================
Please select the atom type:
1- Amber Force Field (AMBER)
2- General Amber Force Field (GAFF)
3- General Amber Force Field (GAFF2)
Enter your choice:
```

  Once you have selected the atom type, the code will prompt you to provide the output file containing the charge information. Accepted file formats include .log and .gesp. You will see the following prompt:

```
Please provide the charge output file (e.g., .log, .gesp):
```

Option 2 and 3: Orca (CHELPG and RESP charges)

If you select this option, the code will proceed to generate the .mol2 file using **CHELPG** or **RESP** charges from the ORCA output. (Refer to the example input in the Appendix for further details).

- **mol2 File Generation**

After specifying the total charge, the code will prompt you to choose the atom type for the system. The options are **GAFF** or **AMBER** or **GAFF2**:

```
================================
 Atom Type Selection Menu
================================
Please select the atom type:
1- Amber Force Field (AMBER)
2- General Amber Force Field (GAFF)
3- General Amber Force Field (GAFF2)
Enter your choice:
```

Once you have selected the atom type, the code will prompt you to provide the output file containing the charge information.

For option 2: Accepted file formats include .log and .out. You will see the following prompt:

```
Please provide the charge output file (e.g., .log, .out):
```

For option 3: Accepted file formats include .vpot. You will see the following prompt:

```
Please provide the charge output file (e.g., .vpot ):
```

Option 4 and 5: Gamess (RESP and Gamess charges)

If you select this option, the code will proceed to generate the .mol2 file using **RESP** or **Gamess** charges (i.e. CHELPG, CONNOLLY or GEODESIC) from the Gamess output. (Refer to the example input in the Appendix for further details).

- **mol2 File Generation**

After specifying the total charge, the code will prompt you to choose the atom type for the system. The options are **GAFF** or **AMBER** or **GAFF2**:

```
================================
 Atom Type Selection Menu
================================
Please select the atom type:
1- Amber Force Field (AMBER)
2- General Amber Force Field (GAFF)
3- General Amber Force Field (GAFF2)
Enter your choice:
```

Once you have selected the atom type, the code will prompt you to provide the output file containing the charge information.

For option 4: Accepted file formats include .dat. You will see the following prompt:

Please provide the charge output file (e.g., .dat):

For option 5: Accepted file formats include .log. You will see the following prompt:

Please provide the charge output file (e.g., .log):

1.5 Seminario Method Setup

To run the Seminario method, you must provide one of the required input files. The system will guide you through selecting the appropriate format based on your input.

```
================================
 Input Format Selection Menu
================================
Please select the format you will provide:
1- Orca Output
2- Gaussian Output
3- Gaussian Checkpoint
4- Gaussian Formatted Checkpoint
5- Gamess Output
Enter your choice:
```

1. **Handling an Orca Output:**

   If you select the Orca output option, two files are required to properly run the Seminario method:
   - Orca Output File (.log): You will be prompted to provide the Orca output file. However, if you selected the CHELPG charge method, this file is not required.
   - Orca Hessian File (.hess): You will also need to provide the Orca Hessian file.

   Please provide the Orca output file (.log):
   Please provide the Orca hessian file (.hess):

2. **Handling a Gaussian Output**

   If you select the Gaussian output option, you will need to provide one file to proceed:
   - Gaussian Output File (.log or .out): The system will prompt you for the output file.

   Please provide the Gaussian output file (.log or .out):

3. **Handling a Checkpoint file:**

   If you choose to provide a Gaussian checkpoint (.chk) file, the system will check for Gaussian availability. You'll be asked whether Gaussian is already installed on your system or if you need to manually provide the path to Gaussian's formchk utility.

   Select your option:

1- Gaussian is already loaded (formchk is available)
2- Provide the Gaussian path
Enter your choice (1 or 2):

4. Handling a Formatted Checkpoint file:

If you provide a formatted checkpoint (.fchk) file, the system will automatically proceed to the next step, with no additional input required.

5. Handling a Gamess Output:

If you select the Gamess output option, you will need to provide one file to proceed:

- Gamess Output File (.dat): The system will prompt you for the output file.

Please provide the Gamess output file (.log or .out):

1.6 Metalloprotein structure

If your structure belongs to a metalloproteins, **easyPARM** will prompt you to confirm this. The system will ask:

Does your structure belong to MetalloProtein ? (y/n):

If you answer Yes:

The tool will proceed to request the protein pdb (i.e. metal + protein) file required for metalloprotein processing.

Please provide the metalloprotein pdb file:

If you answer No:

The tool will treat the system as a regular molecular complex (non-metalloprotein). It will proceed to generate the output files without requiring additional files.

1.7 Output Generation

After successfully completing all the preceding steps and ensuring that all required inputs are correctly provided, the software will generate the necessary parameter files for further use. Upon completion, you will receive a set of output files that may include:

A) Standard Output Files

```
================================
                Output Files
================================
Mol2       : COMPLEX.mol2
Frcmod  : COMPLEX.frcmod
PDB       : COMPLEX.pdb
Lib        : COMPLEX.lib
New Atom Type          : Hybridization_Info.dat
```

B) Metalloprotein Output Files

When processing a metalloprotein, **easyPARM** generates several output files essential for simulations and further preparation:

```
================================
                Output Files
================================
Mol2       : All residues and metal mol2
Frcmod  : COMPLEX.frcmod
MetalloProtein pdb       : easyPARM_MetalloProtein.pdb
Bond Information        : Bond_Info.dat
New Atom Type           : Hybridization_Info.dat
```

The **mol2 files** contain molecular structures for the residues and the metal atom involved in the system. For example, if the system includes a zinc (Zn) metal ion coordinated to four residues, the output will include four separate mol2 files for each residue and a METAL.mol2 file for the metal atom. The METAL.mol2 file also incorporates any linked non-standard residues, ensuring that all relevant structural details are captured in a single file.

The **COMPLEX.frcmod file** includes all the derived force field parameters needed for the system.

The **easyPARM_MetalloProtein.pdb file** contains the complete protein structure, including the metal atom and any newly defined residues that are linked to the metal. This file is designed for use in **tleap**, where it serves as the starting point for generating the topology (prmtop) and coordinate (inpcrd) files needed for simulations.

Hybridization_Info.dat file includes the new atom types along with their corresponding atom names and SP3 hybridization. This file should be used to fulfill the requirements of the *addAtomTypes* keyword in tleap.

Finally, the **Bond_Info.dat file** lists the specific bonds between the metal and its coordinating residues. This information must be manually added in **tleap** to define the correct bonding interactions. This step ensures that the resulting topology file (prmtop) accurately represents the metalloprotein's structure and interactions.

1.8 Atomic Charge Restraint (Optional)

The program provides the option to restrain the atomic charge on specific atoms. Upon execution, the following prompt will appear:

```
Would you like to restrain the charge on specific atoms? (Yes or No):
```

- Selecting No will terminate the operation.
- Selecting Yes will allow you to provide details for one or more atoms

1.8.1 Specify the Number of Atoms

If you choose to proceed, the following prompt will appear:

```
How many atoms do you want to restrain?
```

At this point, you will be required to specify the exact number of atoms for which the charge is to be restrained.

1.8.2 Providing Atom-Specific Information

You will see the following prompt:

Please provide the atom number and its charge for atom 1 (e.g., 12 -0.834):

You are required to input both the atom number and its corresponding charge for each selected atom.

1.8.3 Inputting Charges Output

You will see the following prompt:

Please enter the name of output that contains all the ESP charges:

At this point, you must specify the file name containing the Electrostatic Potential (ESP) charges:
For Gaussian: Provide the **standard output file** (e.g., .log or .out) that contains the ESP charges.
For Orca: Provide the output file with the **.vpot extension**, which contains the ESP charges.
For Gamess: Provide the output file with the **.dat extension** which contains the ESP charges

1.8.4 Generating the Output

After processing, easyPARM generates new .mol2 and library (.lib) files containing the updated atomic charges derived from the information provided. These updated files reflect the charge distribution tailored to your system, ensuring compatibility with the AMBER force field for accurate simulations.

Note: At this stage, charge restraints are not supported for metalloprotein systems. This limitation means that the derived charges are computed without applying additional restraints.

2. Option 2: Generate Metalloprotein .XYZ Structure

In this step, **easyPARM** helps generate an initial .xyz structure for further calculations required to derive metalloprotein parameters. The process begins with a prompt asking you to provide a prepared **PDB file**:

Please provide the metalloprotein pdb file:

It is important to ensure the PDB file is properly prepared before using it with easyPARM. You can use tools such as H++ or ProteinPrepare to preprocess the structure. Key considerations include:

1. **Residue Corrections**: Ensure all residues, especially those linked to the metal, are properly assigned. For example:
   - **Cysteine Residues:** CYS residues bound to the metal should be changed to CYM (deprotonated cysteine) etc.
   - **Correct Orientation of Side Chains:** Check the orientation of **ASN**, **GLN**, and **HIS** groups. The process of correcting orientations and adding hydrogens can sometimes shift the positions of residues linked to the metal. Verify that the metal and its coordinating residues are positioned

correctly, as this region will directly influence the .xyz file generated for parameterization.

- **Ligands (Non-standard Residues):** Ensure all ligand atoms have unique names in the PDB file. Tools such as reduce (from AmberTools), Avogadro, or GaussianView can be used to add hydrogens if missing. Example correction for duplicate atom names:

Incorrect:

```
ATOM    528  H   LIG A  31     -12.251  -8.336  -3.849  1.00  9.27           H
ATOM    529  H   LIG A  31     -12.251  -8.336  -3.849  1.00  9.27           H
```

Corrected:

```
ATOM    528  H1  LIG A  31     -12.251  -8.336  -3.849  1.00  9.27           H
ATOM    529  H2  LIG A  31     -12.251  -8.336  -3.849  1.00  9.27           H
```

2. **Handling Multiple Metals**: If the PDB file contains multiple distinct, unconnected metals, each must be parameterized separately. However, if the metals are linked to each other, you can proceed without isolating them.
   - For the first run, retain only one metal while removing others.
   - After parameterizing the structure, switch to the next metal by removing the first and repeating the process.
   - Failure to do this will result in an incorrect initial structure because the code cannot handle multiple metals simultaneously.
3. **Capping Groups**

To facilitate geometry optimization and achieve reasonable atomic charges, the tool automatically adds capping groups to the termini of the amino acid residue linked to the metal :

- Acetyl Group: Added to the N-terminal.
- Amidated Group (NH_2): Added to the C-terminal.

These capping groups ensure the termini are chemically stable for subsequent calculations. Note:- These capping groups will have zeros charges later and will be excluded from the force field later by the code.

4. **Output File**

The generated output will be an initial structure in .xyz format:

XYZ Output: initial_structure.xyz

This file serves as the starting point for QM calculations, including geometry optimization, frequency analysis, and charge derivation, which are necessary for creating accurate metalloprotein parameters. By following this structured approach, **easyPARM** ensures the correctness and readiness of the initial structure for downstream processing.

3. Convert AMBER Parameters To Openmm Or GROMACS Format

At this stage, **easyPARM** allows you to convert AMBER parameters into either OpenMM or GROMACS formats using **ParmEd**. Upon execution, you will be prompted to select your desired conversion option from the following menu:

```
echo "================================"
echo "     AMBER Converter Menu       "
echo "================================"
echo "Select your option:"
echo "1- AMBER to OpenMM"
echo "2- AMBER to GROMACS"
```

After selecting your preferred format, the code will request two essential input files to perform the conversion:

```
Please provide the prmtop file:

Please provide the inpcrd file:
```

- **If OpenMM format is selected**:

  The output will be generated as *system.xml*.

- **If GROMACS format is selected**:

  The output will include *system_gmx.gro* (coordinate file) and *system_gmx.top* (topology file).

4. Appendix

Example for Generating the Required Files

4.1 GAUSSIAN Input

When using Gaussian to generate the required files, note that the choice of the level of theory is up to the user and is not mandatory to match the examples provided. Follow the steps below:

4.1.1 Structure Optimization

First, optimize the molecular structure. Upon completion, you will obtain the optimized structure in .xyz format. Once this step is complete, proceed to the next step.

```
%mem=30GB
%nproc=24
%chk=output.chk
#p B3LYP/def2tzvp symmetry=none opt scf=(MaxCycle=60,xqc)
int(ultrafinegrid) iop(7/33=1)

#Comment

charge multiplicity
coordinates
```

4.1.2 Frequency Calculation

Use the optimized structure to perform a single-point frequency calculation. Upon completion, you will obtain the following files:

- Optimized Structure (.xyz): Contains the final geometry of the optimized structure.
- Checkpoint File (.chk): Stores data from the frequency calculation, including the Hessian matrix and other important information.
- Gaussian output (.log): Contains Hessian matrix and other important information.

```
%mem=30GB
%nproc=24
%chk=output.chk
#p B3LYP/def2tzvp symmetry=none scf=(MaxCycle=60,xqc)
freq(noraman) int(ultrafinegrid) iop(7/33=1)

#Comment

charge multiplicity
coordinates
```

4.1.3 Charge Calculation Using the RESP Method

Next, calculate the molecular charge using the MK method in Gaussian. This step will produce the charge output in the .log file.

```
%mem=30GB
%nproc=24
#p B3LYP/gen pseudo=read SCF=tight Pop=(mkuff) iop(6/33=2)
iop(6/42=6) iop(6/50=1)
#Comment
charge multiplicity
coordinates

C N H 0
6-31G*
****

Ru 0
SDD
****

Ru 0
SDD

output.resp

output.resp
```

4.2 ORCA Input

When using Orca to generate the required files, the choice of the level of theory is flexible and does not need to match the examples provided. Follow the steps below:

4.2.1 Structure Optimization

First, perform a structure optimization of the molecule. Upon successful completion, you will obtain the optimized structure in .xyz format. Once this step is complete, proceed to the next step.

```
!B3LYP 6-31G*  RI-SOMF(1X) defgrid2 KDIIS TIGHTSCF Opt
%pal nprocs 24
end
%basis
NEWGTO  Pt "SDD" END
NewECP Pt "SDD" end
end
%scf MaxIter 1000 end
%maxcore 5000
* xyz 1 1
Coordinates
*
```

4.2.2 Frequency Calculation and Charge Calculation with the CHELPG or RESP Method

1. Run a Single-Point Frequency Calculation using the optimized structure.
2. Output Files: Upon completion of the calculation, you will obtain the following files:
   - Optimized Structure (.xyz): Contains the final geometry of the optimized structure.
   - Charge File (.log): Contains the calculated charges based on the CHELPG method.
   - Hessian File (.hess): Contains the Hessian matrix from the frequency calculation.
   - ESP charge (.vpot): Contains the calculated ESP charges based on the CHELPG method. This output can be used by easyPARM to calculate the RESP charges.

```
!B3LYP 6-31G*  RI-SOMF(1X) defgrid1 KDIIS TIGHTSCF Freq
%pal nprocs 24
end
%basis
NEWGTO  Pt "SDD" END
NewECP Pt "SDD" end
end
%scf MaxIter 1000 end
! CHELPG
%maxcore 5000
* xyz charge multiplicity
Coordinates
*
```

4.3 GAMESS Input

When using Gamess to generate the required files, note that the choice of the level of theory is up to the user and is not mandatory to match the examples provided. Follow the steps below:

4.3.1 Structure Optimization

First, optimize the molecular structure. Upon completion, you will obtain the optimized structure in .xyz format. Once this step is complete, proceed to the next step.

```
$SYSTEM MEMDDI=400 MWORDS=200 $END
$CONTRL DFTTYP=B3LYP RUNTYP=OPTIMIZE ICHARG=charge
MULT=multiplicity $END
$STATPT NSTEP=1000 $END
$BASIS GBASIS=N31 NGAUSS=6 NDFUNC=1 $END
$DATA
Title
C1
coordinates
$END
```

4.3.2 Frequency Calculation

Use the optimized structure to perform a single-point frequency calculation. Upon completion, you will obtain the following files:

- Optimized Structure (.xyz): Contains the final geometry of the optimized structure.
- Gamess output (.log): Contains frequency calculation and other important information.
- Gamess output (.dat): Contains hessian matrix and other important information.

```
$SYSTEM MEMDDI=600 MWORDS=300 $END
$CONTRL DFTTYP=B3LYP RUNTYP=HESSIAN ICHARG=charge
MULT=multiplicity SCFTYP=RHF $END
$BASIS GBASIS=N31 NGAUSS=6 NDFUNC=1 $END
$DATA
Title
C1
coordinates
$END
```

4.1.3 Charge Calculation Using the RESP or GAMESS Methods.

The next step involves calculating the molecular charges using the available methods in **GAMESS**. This process generates charge outputs in either .log or .dat files, depending on the method employed.

- **Using RESP Method**

To calculate RESP charges, the ESP points must be obtained using the **CONNOLLY** option for the **PTSEL** keyword. This method uses a set of points on several fused van der

Waals surfaces, following the algorithm by Michael Connolly, identical to the approach by Kollman and Singh.

The ESP points (output as .dat files) will then be used by **easyPARM** to compute RESP charges. The recommended GAMESS input file for this step is as follows:

```
$SYSTEM MEMDDI=800 MWORDS=200 $END
 $CONTRL DFTTYP=B3LYP ICHARG=charge MULT=multiplicity $END
 $ELPOT IEPOT=1 WHERE=PDC $END
 $PDC PTSEL=CONNOLLY CONSTR=NONE $END
 $BASIS GBASIS=N31 NGAUSS=6 NDFUNC=1 $END
 $DATA Title
C1
Coordinates
$END
```

Note: While the default option is **CONNOLLY**, other methods like **GEODESIC** or **CHELPG** for the **PTSEL** keyword can also be used. In these cases, the ESP points (output as .dat files) are similarly utilized by **easyPARM** to compute RESP charges.

- **Using GAMESS Charge Fitting**

For GAMESS charge fitting, you can select one of the supported methods: **GEODESIC**, **CONNOLLY**, or **CHELPG**. However, you must include the ***CONSTR=CHARGE*** keyword, ensuring the sum of the fitted atomic charges reproduces the total molecular charge.

The fitted charges (output in .log files) will be directly used by **easyPARM**. The recommended GAMESS input for this step is as follows:

```
$SYSTEM MEMDDI=800 MWORDS=200 $END
 $CONTRL DFTTYP=B3LYP ICHARG=charge MULT=multiplicity $END
 $ELPOT IEPOT=1 WHERE=PDC $END
 $PDC PTSEL=CONNOLLY CONSTR=CHARGE $END
 $BASIS GBASIS=N31 NGAUSS=6 NDFUNC=1 $END
 $DATA Title
C1
Coordinates
$END
```

Both methods ensure that the charges are computed accurately, providing the necessary data for generating reliable molecular parameters. **easyPARM** supports ESP points generated through these methods and seamlessly integrates them for RESP charge calculations or GAMESS-fitted charges.

Note: The basis set **6-31G*** is suitable for metals like zinc (Zn). However, for heavier metals such as ruthenium (Ru) or platinum (Pt), this basis set may not be optimal. In these cases, it is recommended to use **SBKJC** or another basis set that is better suited for such metals. It is important to choose a basis set that includes an **Effective Core Potential (ECP)**, which provides more accurate results for metal-containing systems by accounting

for the core electron effects. You should select the best fitting basis set that matches your specific metal and system for improved accuracy.

5. Troubleshooting Metal Issues in Tleap

1. **Using .mol2 and .frcmod Files to Generate Libraries**

   When using mol2 and frcmod files in tleap, metals may not have their connectivity and atomic numbers correctly assigned. To avoid this issue, it is recommended to use the pre-generated .lib and .frcmod files, which include all the necessary data for proper parameterization.

   Your tleap script should include the following commands:

   ```
   loadamberparams COMPLEX.frcmod
   loadoff COMPLEX.lib
   ```

   After generating prmtop file, it is recommended to check the `%FLAG ATOMIC_NUMBER` section to ensure that the atomic numbers of metals are correctly assigned. This is particularly important for metalloproteins, as the atomic number for metal may sometimes be incorrectly recorded as -1.

2. **Handling Metals with High Coordination Numbers**

   If your system includes a metal with a coordination number greater than 8, you may encounter the following warning when generating the **prmtop** file in tleap:

   ```
   Bond: Maximum coordination exceeded on A<Co1 23>
   -- setting atoms pert=true overrides default limits
   ```

   This warning indicates that the metal exceeds the default bond limit of 8 in **AmberTools**, and as a result, not all bonds involving the metal will be included in the generated **prmtop** file. This will lead to incorrect results.

   Solution

   To resolve this issue, you need to increase the maximum bond limit in the AmberTools source code:

   1. Access the AmberTools source directory: `$AMBER_HOME/AmberTools/src/leap/src/leap`
   2. Edit the `atom.h` file to increase the bond limit. By default, the maximum number of bonds is set to 8:

      ```
      /* maximum 8 bonds out of each atom */

      #define MAXBONDS 8
      ```

   3. Change 8 to the desired number of bonds.
   4. Recompile Amber to apply the changes, allowing tleap to recognize the new MAXBONDS limit.

6. Tleap Inputs

The following examples illustrate **tleap** input for preparing metal complexes and metalloprotein systems using **AMBER 22**. These inputs provide a general framework and should be adapted to your specific system.

Metal Complex Input Example

This example corresponds to **complex 7** from the original article describing the code:

```
source leaprc.gaff
source leaprc.water.tip3p
source leaprc.protein.ff19SB
loadamberparams frcmod.ionsjc_tip3p
# Define hybridization
addAtomTypes {
        { "Ru"  "Ru" "sp3" }
        { "C1"  "C" "sp3" }
        { "C2"  "C" "sp3" }
        { "C3"  "C" "sp3" }
        { "C4"  "C" "sp3" }
        { "C5"  "C" "sp3" }
        { "C6"  "C" "sp3" }
        { "N1"  "N" "sp3" }
        { "N2"  "N" "sp3" }
}
# Load force field for metal complex
loadamberparams COMPLEX.frcmod
loadoff COMPLEX.lib
# Load protein + metal complex
mol = loadpdb "HB1.pdb"
# Save vacuum files
savepdb mol HB1_vacuum.pdb
saveamberparm mol HB1_vacuum.prmtop HB1_vacuum.inpcrd
# Solvate the system
solvateoct mol TIP3PBOX 10.0
check mol charge
addions mol Na+ 0.
addions mol Cl- 0
# Save solvated files
savepdb mol HB1_solvated.pdb
saveamberparm mol HB1_solvated.prmtop HB1_solvated.inpcrd
quit
```

Metalloprotein Input Example

This example showcases the setup for a **metalloprotein system**:

```
source leaprc.gaff
source leaprc.protein.ff19SB
```

```
source leaprc.water.tip3p
loadamberparams frcmod.ionsjc_tip3p
# Define hybridization
addAtomTypes {
        { "Zn"  "Zn" "sp3" }
        { "s1"  "S" "sp3" }
        { "s2"  "S" "sp3" }
        { "n3"  "N" "sp3" }
        { "n4"  "N" "sp3" }
}
# Load force field for residues and metal
loadamberparams COMPLEX.frcmod
# Load residues and metal
CY1 = loadmol2 "CY1.mol2"
CY2 = loadmol2 "CY2.mol2"
HI1 = loadmol2 "HI1.mol2"
HI2 = loadmol2 "HI2.mol2"
mol = loadmol2 "METAL.mol2"
PRO1 = loadpdb "easyPARM_MetalloProtein.pdb"
# Define bonds for the metal coordination
bond PRO1.32.Zn1 PRO1.5.SG
bond PRO1.32.Zn1 PRO1.10.SG
bond PRO1.32.Zn1 PRO1.23.NE2
bond PRO1.32.Zn1 PRO1.27.NE2
# Link adjacent residues
bond PRO1.5.N PRO1.4.C
bond PRO1.5.C PRO1.6.N
bond PRO1.10.N PRO1.9.C
bond PRO1.10.C PRO1.11.N
bond PRO1.23.N PRO1.22.C
bond PRO1.23.C PRO1.24.N
bond PRO1.27.N PRO1.26.C
bond PRO1.27.C PRO1.28.N
# Save vacuum files
savepdb mol 1SP2_vacuum.pdb
saveamberparm mol 1SP2_vacuum.prmtop 1SP2_vacumm.inpcrd
# Solvate the system
solvateoct PRO1 TIP3PBOX 10.0
check PRO1 charge
addions PRO1 Cl- 0.
addions PRO1 Na+ 0.
# Save solvated files
saveamberparm PRO1 1SP2_solvated.prmtop 1SP2_solvated.inpcrd
savepdb PRO1 1SP2_solvated.pdb
quit
```

Notes:

1. Ensure the **frcmod** and **mol2** files for your metal-containing residues are correctly generated using easyPARM.
2. Adjust the hybridization section from **Hybridization_Info.dat** file.
3. Adjust the metal coordination bonds from **Bond_Info.dat** file.
4. When working with solvated systems, the box dimensions and ion addition steps may vary depending on your requirements.

7. References

(1) Seminario, J. M. Calculation of Intramolecular Force Fields from Second-Derivative Tensors. *Int J Quantum Chem* **1996**, *60* (7), 1271–1277. https://doi.org/https://doi.org/10.1002/(SICI)1097-461X(1996)60:7<1271::AID-QUA8>3.0.CO;2-W.

(2) Case, D. A.; Aktulga, H. M.; Belfon, K.; Ben-Shalom, I. Y.; Brozell, S. R.; Cerutti, D. S.; Cheatham III, T. E.; Cisneros, G. A.; Cruzeiro, V. W. D.; Darden, T. A.; Duke, R. E.; Giambasu, G.; Gilson, M. K.; Gohlke, H.; Goetz, A. W.; Harris, R.; Izadi, S.; Izmailov, S. A.; Jin, C.; Kasavajhala, K.; Kaymak, M. C.; King, E.; Kovalenko, A.; Kurtzman, T.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Machado, M.; Man, V.; Manathunga, M.; Merz, K. M.; Miao, Y.; Mikhailovskii, O.; Monard, G.; Nguyen, H.; O'Hearn, K. A.; Onufriev, A.; Pan, F.; Pantano, S.; Qi, R.; Rahnamoun, A.; D.R. Roe; Roitberg, A.; Sagui, C.; Schott-Verdugo, S.; Shen, J.; Simmerling, C. L.; Skrynnikov, N. R.; Smith, J.; Swails, J.; Walker, R. C.; Wang, J.; Wei, H.; Wolf, R. M.; Wu, X.; Xue, Y.; York, D. M.; Zhao, S.; Kollman, P. A. Amber 20. *Amber 20* **2021**, San Francisco.

(3) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. a.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. a.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, a. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, a. F.; Sonnenberg, J. L.; Williams; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. a.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. a.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, a. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. G16_C01. 2016, p Gaussian 16, Revision C.01, Gaussian, Inc., Wallin.

(4) Neese, F.; Wennmohs, F.; Becker, U.; Riplinger, C. The ORCA Quantum Chemistry Program Package. *J Chem Phys* **2020**, *152* (22), 224108. https://doi.org/10.1063/5.0004608.

(5) Barca, G. M. J.; Bertoni, C.; Carrington, L.; Datta, D.; De Silva, N.; Deustua, J. E.; Fedorov, D. G.; Gour, J. R.; Gunina, A. O.; Guidez, E.; Harville, T.; Irle, S.;

Ivanic, J.; Kowalski, K.; Leang, S. S.; Li, H.; Li, W.; Lutz, J. J.; Magoulas, I.; Mato, J.; Mironov, V.; Nakata, H.; Pham, B. Q.; Piecuch, P.; Poole, D.; Pruitt, S. R.; Rendell, A. P.; Roskop, L. B.; Ruedenberg, K.; Sattasathuchana, T.; Schmidt, M. W.; Shen, J.; Slipchenko, L.; Sosonkina, M.; Sundriyal, V.; Tiwari, A.; Galvez Vallejo, J. L.; Westheimer, B.; Włoch, M.; Xu, P.; Zahariev, F.; Gordon, M. S. Recent Developments in the General Atomic and Molecular Electronic Structure System. *J Chem Phys* **2020**, *152* (15), 154102. https://doi.org/10.1063/5.0005188.

(6) Shirts, M. R.; Klein, C.; Swails, J. M.; Yin, J.; Gilson, M. K.; Mobley, D. L.; Case, D. A.; Zhong, E. D. Lessons Learned from Comparing Molecular Dynamics Engines on the SAMPL5 Dataset. *J Comput Aided Mol Des* **2017**, *31* (1), 147–161. https://doi.org/10.1007/s10822-016-9977-1.

(7) Comell, W. D.; Cieplak, P.; Bayly, C. I.; Kollman, P. A. Application of RESP Charges. *journal of American Chemical Society* **1993**, *115* (7), 9620–9631.