

Tutorial on Simulating EXAFS Using *Demeter*

Mikhail Askerka, Benjamin Rudshiteyn, Ke R. Yang, and Victor S. Batista*

victor.batista@yale.edu

Yale University, Department of Chemistry

225 Prospect Street, New Haven, CT 06520

Written October 19, 2016

At least we don't have to wait for beam time.

Table of Contents

Section 1. Introduction.	2
Section 2. Setting up Your Computer.	3
Section 3. Things To Be Done.	5
Section 4. Python Scripts: genexafs.py and fitting.py.	6
Section 5. The PSII Example.	10
Section 6. Monte Carlo Refinement Code: Optmin.	15
Section 7. References.	16

Section 1. Introduction.

Extended X-ray Absorption Edge Fine Structure (EXAFS)¹⁻⁹ is a powerful method to study the coordination environments around heavy atom centers, particularly for metal centers relevant to water oxidation catalysts (WOC) e.g. in the solar energy field. Our group has applied it to the Mn-containing Photosystem II (PSII)¹⁰⁻¹⁷ and to the Ir “blue solution” dimer.¹⁸ However, EXAFS in principle can be applied to all elements and various samples (amorphous materials, liquids, solutions, etc.). Its inherently local probe nature prevents EXAFS from uniquely determining structures, as X-ray diffraction (XRD) can, resulting in extremely convoluted data, whose interpretation often relies on the simulation. However, EXAFS is much less prone to radiation damage¹² and is most helpful when crystal structures are not available.¹⁸ The basic principles, strengths, and weaknesses of EXAFS as well as an example from a recent research problem¹⁸ have been discussed already in the accompanying file, **Dropbox/Tutorials/EXAFS/Ir_Blue_Solution_for_Ben/160729EXAFS_Solar_Energy_v1.pptx**, which uses files in the **all_files** subdirectory. Some parts are already included in this tutorial, while may be included within a future version.

Section 2. Setting up Your Computer.

This tutorial assumes you are running the Mac operating system. The main program, *Demeter*,¹⁹ <http://bruceravel.github.io/demeter/>, is available for Windows, but we have not tested it there. The commands with “sudo” will require the user’s administrator password. For Linux, please replace all the “sudo port” commands with “sudo apt-get” install (skip the downloads of Macports and X11).

3. Download Macports at <https://www.macports.org/install.php>.
 - a. It may do it on its own at some point, but the following is needed for MacPorts to work properly.

export PATH=/opt/local/bin:/opt/local/sbin:\$PATH

4. Download X11 (aka xquartz) to visualize terminal applications at <https://www.xquartz.org>.
5. Use the **sudo port install** command in the Terminal to install the following Macports (others that are recommended are given at our Wiki: <http://wikidchem.wikidot.com/wiki:software>) *in this order*:
 - a. **xorg-libXt +flat_namespace**
 - b. **py27-pip**
 - c. **py34-pip**
 - d. **grace** (if this does not work, try **xmgr**)
 - e. **gcc47**
 - f. **gnuplot**
 - g. **xorg-server Demeter**

Note: If gnuplot keeps upgrading from version 4 to 5 without you wanting it, you can undo it this way with the full name of whatever gnuplot version you are using instead of the one below:

sudo port activate gnuplot

@4.6.5_2+aquaterm+luaterm+pangocairo+wxwidgets+x11

6. We recommend installing python this way by using these commands:
 - a. **sudo port select --set pip pip27** (or **sudo port select --set pip pip34** if you want to run Python 3)
 - b. **pip install --upgrade ase**
 - c. **pip install --upgrade numpy scipy matplotlib**
 - d. **sudo pip freeze --local | grep -v '^-\e' | cut -d = -f 1 | xargs -n1 sudo pip install -U**
7. To get these python commands to work, adding these export statements to your `~/.bash_profile` would help (do not include forced line breaks):

export

PYTHONPATH=/opt/local/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-packages:\$PYTHONPATH

export

PATH=/opt/local/Library/Frameworks/Python.framework/Versions/2.7/bin:\$PATH

8. If you can enter **athena** into the terminal and see the GUI, you have been successful. We will avoid the GUI to just use the other terminal commands *Demeter* has provided, particularly **feff6** and **ifffit**, which activates the IFEFFIT²⁰ part of the program, which generates EXAFS from an XYZ structure.

You may check your installation of python to make sure that when the which python command is used, something like this kind of folder is printed out (or whatever the commands in 6 print out):

/opt/local/Library/Frameworks/Python.framework/Versions/2.7/bin/python

Section 3. Things To Be Done.

In addition to generating the theoretical data, the raw experimental data have to be treated as well. The procedure is as follows (referring to files in Table 1):

- I. Convert measured intensities (I) to absorption coefficient $\mu(E)$
- II. Remove smooth background $\mu_0(E)$ from $\mu(E)$
- III. Determine the absorption edge E_0 .
- IV. Convert absorption coefficient $\mu(E)$ to fine structure function $\chi(E)$.
- V. Convert $\chi(E)$ to $\chi(k)$.
- VI. K -weight the EXAFS $\chi(k)$ [the python scripts assume all previous steps were already done]
- VII. Fourier transform into R -space.

On the theoretical side, we have to do the following:

- A. Generate an Input Structure as an .xyz file, typically from a DFT or QM/MM optimization.
- B. Determine the list of the absorbing atoms [done in **gen_exafs.py**]
- C. Feed it into the FEFF program [through the bash script **compile_xyz_to_feff**, that creates an input file for FEFF using **headfeff.inp**, constants.f90, **xyz_to_feff.f90**].
- D. Calculate scattering amplitude $f(k)$, phase-shifts $\delta(k)$, and mean-free path $\lambda(k)$ [It is executed for each of the absorbing atoms independently using bash script **exafs_fit**].
- E. Feed it into the IFEFFIT program [This is where one, if needed, can change the global k -range, interpolation k -grid, global Debye-Waller factor and other parameters (refer to the EXAFS documentation)].
- F. Refine edge energy E_0 to reproduce experimental $\chi(k)$ [fitting routine in **EXAFS_routines.py** refines the edge].
- G. (Optional): Refine the structure using the Monte Carlo refinement code, *optmin*.
- H. (Optional) Reoptimize this structure with the same parameters as the initial calculation and go back to Step A.

Section 4. Python Scripts: genexafs.py and fitting.py.

1. First, open the **Build_EXAFS_Oct2016.tar.gz** in the Dropbox in a new folder (or double-click in the Finder).

a. tar -xvzf Build_EXAFS_Oct2016.tar.gz

2. You should have the files listed in Table 1 in the folder.

Table 1. The list of files with their purpose that come out of the .tar file in alphabetical order.

Filename	Purpose
calc2.iff	Input file needed for IFEFFIT program. Does interpolation and Fourier transform of the $\chi^2(k)$ data.
calc2_exp.iff	Input file needed for IFEFFIT program. Does interpolation and Fourier transform of the $\chi^2(k)$ data for the experimental data.
compile_xyzfeff	Bash script to compile the Fortran files to create individual input files for each of the absorbing centers for FEFF.
constants.f90	Contains physical constants needed for Fortran codes.
coords_S3.xyz	Portion of the PSII S3 geometry used for this tutorial.
exafs_fit	Bash script to operate the fitting part of the program.
EXAFS_routines.py	Contains the EXAFS routines used by the main python code.
fe_chi_k3_exp.DAT	Experimental data in form of k vs. $\chi^2(k)$; obtain from colleagues or the Demeter GUI (this will be added to tutorial in the future).
gen_exafs.py	Main python script.
headfeff.inp	Part of a FEFF6 input file needed for xyz_to_feff.f90.
input_EXAFS.txt	Input file for main python script.
make_calc	Bash script that creates the calc.iff input file for IFEFFIT.
process.iff	IFEFFIT commands needed to build the eventual input file.
xyz_to_feff.f90	A Fortran program to extract the coordinates from the xyz file to build a FEFF input.

3. Try this command:

python gen_exafs.py

It will print to the screen the proper usage of the program and what should go into the input_EXAFS.txt program. Do not remove any lines from this input file and do not enter any nonsensical options (such as a word where a number should be). The output to the screen is as follows below. Recall that 0 = false and 1 = true for the first four “Action” options. This output should explain what each part of the input file is for.

gen_exafs.py - A Python script that constructs EXAFS spectra in the momentum and reduced distance coordinates and fits it to experimental data

```
#####  
##### Usage: python gen_exafs.py [inputfile] #####  
#####
```

Input file keywords:

```
##### Action options #####  
build_exafs (0 or 1) Build EXAFS spectra using FEFF  
package  
fitting (0 or 1) Fit the spectra to the experiment  
by shifting the EXAFS edge  
build_plot (0 or 1) Build and save high resolution  
plots  
do_home_made_FT (0 or 1) Do the home made Fourier  
transform or the Ifeffit Fourier Transform  
##### EXAFS options #####  
abs_atom (str) Absorbing atom in the EXAFS  
experiment  
atomic_number (int) Atomic number of that atom  
edge_shift (float) Initial guess for the shift edge  
n_points (int) Grid density for the k-spectrum  
interpolation  
abs_atoms_line (str) Always "" not sure why I made this  
an input  
##### Fitting options #####  
E_shift_min (float) Starting edge shift for the scan  
E_shift_max (float) Final edge shift for the scan  
scanning_step (float) Step for the scan  
grid_density (int) Grid density for the k-spectrum  
interpolation  
##### Ifeffit FT options #####  
kmin (float) Lower boundary for the FT  
window function  
kmax (float) Upper boundary for the FT  
window function  
##### Home made FT options #####  
kmin_FT (float) Lower boundary for the FT  
window function  
kmax_FT (float) Upper boundary for the FT
```

window function
dk (float) Window width. The greater the value, the slower the window decays
T (float) Step size for the new axis in the Fourier Transform
N (int) Number of points on the new axis in the Fourier Transform
name_FT_input (array of str) The names of the files with Chi*k^3 vs k data you want to do the FT on
name_FT_output (array of str) The names of the output files for the FT on
File name options
file_names (array of str) bash files that need to be altered to use FFFF
file_names_ifeffit (array of str) bash files that need to be altered to use Iffffit
coords_file_name (str) the name of the file with input coordinates
exp_file_name (str) the name of the file with data in Chi*k^3 vs k
Plotting options
k_lower (float) Lower boundary for k in the Chi*k^3 vs k plot
k_upper (float) Upper boundary for k in the Chi*k^3 vs k plot
chi_lower (float) Lower boundary for Chi in the Chi*k^3 vs k plot
chi_upper (float) Upper boundary for k in the Chi*k^3 vs k plot
FT_lower (float) Lower boundary for the FT magnitude in the FT plot
FT_upper (float) Upper boundary for the FT magnitude in the FT plot
rd_lower (float) Lower boundary for the reduced distance in the FT plot
rd_upper (float) Upper boundary for the reduced distance in the FT plot

#####
Input File Example
#####

Action options
build_exafs 1

```

fitting                1
build_plot             1
do_home_made_FT       1
##### EXAFS options #####
abs_atom              Mn
atomic_number         25
edge_shift            20
n_points              1000
abs_atoms_line
##### Fitting options #####
E_shift_min           11.
E_shift_max           12.5
scanning_step         0.2
grid_density          1000
##### Ifeffit FT options #####
kmin                   4.0
kmax                   10.5
##### Home made FT options #####
kmin_FT               4.0
kmax_FT               10.5
dk                     3.
T                      0.1
N                      1000
name_FT_input          fe_chi_k3.DAT fe_chi_k3_exp.DAT
name_FT_output         fe_FT.out fe_FT_exp.out
##### File name options #####
file_names             compile_xyzfeff exafs_fit make_calc
file_names_ifeffit     calc.iff calc2.iff
coords_file_name       coords_S3.xyz
exp_file_name          fe_chi_k3_exp.DAT
##### Plotting options #####
k_lower                2
k_upper                12.7
chi_lower              -8.
chi_upper              8.
FT_lower               0.
FT_upper              8.
rd_lower               0.
rd_upper               6.

```

Note:

1. If one indicates 1 for do_home_made_FT, but 0 for BUILD_EXAFS, the program

will crash. The vice versa options are acceptable.

Section 5. The PSII Example.

This example uses the relevant portion of our group's S_3 model of the oxygen-evolving complex (OEC) of PSII,¹⁴ which contains several Mn ions: 6 Å about the center. You will create a spectrum resembling that paper's second figure. This distance is the limitation of the method, besides a 500 atom hard limit. The structure is given in Figure 1.

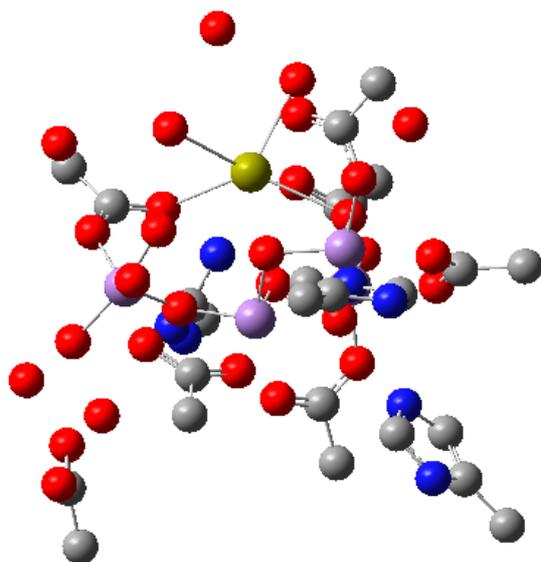


Figure 1. Structure contained in `coords_S3.gjf` with hydrogens omitted. Color code: N = blue, O = red, C = gray, Ca = yellow, and Mn = purple.

Within the folder containing the files in Table 1, launch the program as follows:

```
python gen_exafs.py input_EXAFS.txt
```

The first thing printed to the screen are the numerical labels for the relevant absorbing atoms, followed by the atomic coordinates, a stream of digits, identification of the species, and the output for FEFF. The code will have to run for each absorbing atom separately scattering off all the others. Then the final spectrum would be an average.

Eventually the output will start to look like this:

```
Ifeffit 1.2.12 Copyright (c) 2005 Matt Newville, Univ of Chicago
```

command-line shell version 1.1 with GNU Readline

Iffeffit 1.2.12 Copyright (c) 2005 Matt Newville, Univ of Chicago

command-line shell version 1.1 with GNU Readline

11.0 3395.81994922

The important are the bottom two numbers. The one on the left is the zero-edge shift, E_0 , the value of which is being decided upon. The one on the right is the RMSD indicating how well the resulting spectrum agrees with the experiment: the lower the better.

In the course of events, Figure 2 should appear on the screen plotting various spectra with different E_0 values vs. experiment, followed by Figures 3 and 4, where the numbers are all in eV. The optimal choice of E_0 should be clear from the minimum in Figure 3 and will appear in Figure 4. If no minimum is found, a new range should be selected for the “Fitting options” in **input_EXAFS.txt**. Every time a Figure appears, it must be manually closed for the program to continue. This example should take just a minute overall, but for larger systems, it can take several minutes.

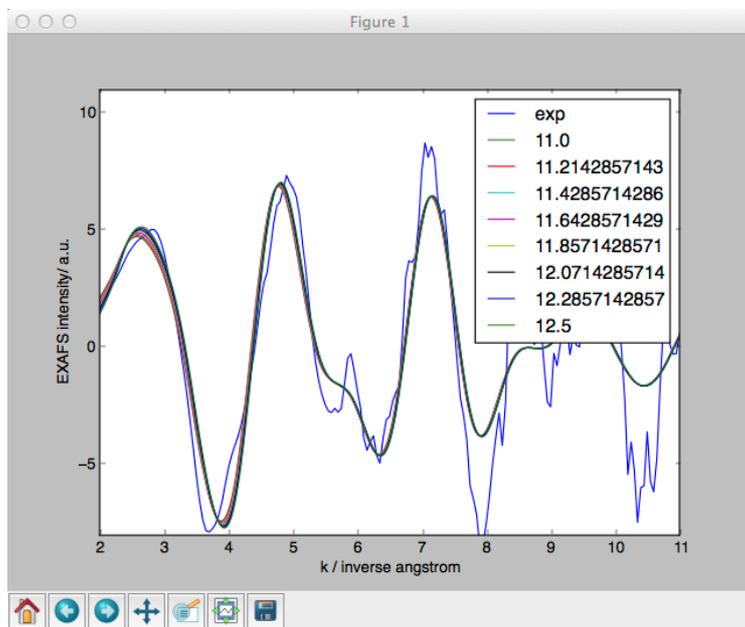


Figure 2. Plot of the tested spectra with different E_0 values vs. experiment in black.

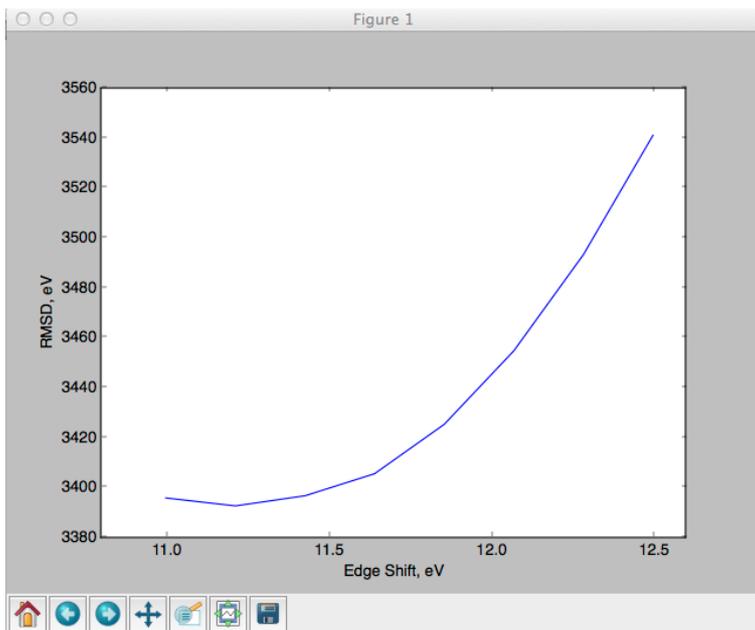


Figure 3. Plot of the RMSD as a function of the E_0 edge shift.

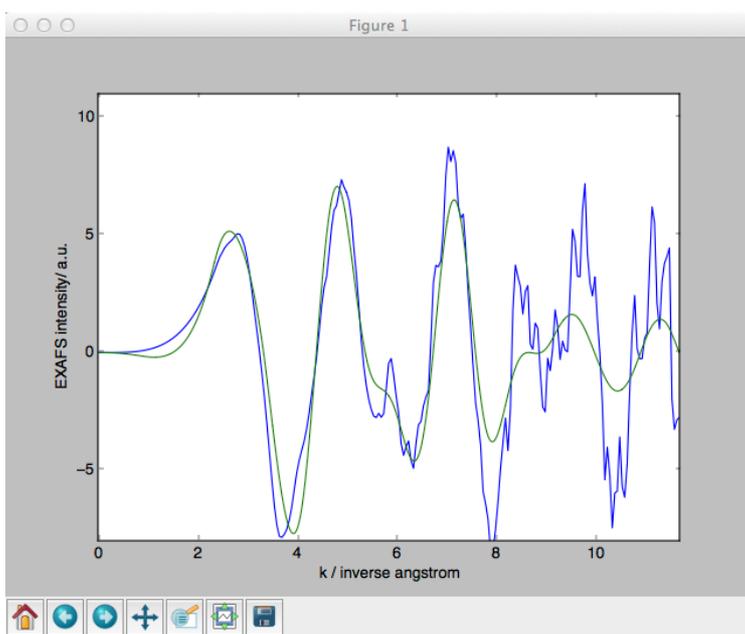


Figure 4. Plot of the spectrum with the best match.

Once the program is done printing things out, Figure 5 will appear.

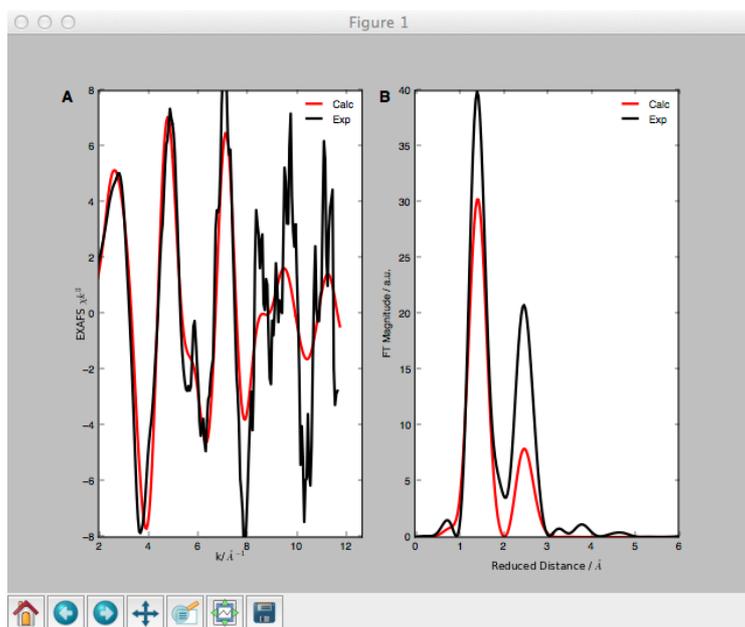


Figure 5. Plot of the k-space spectra (left) and the R-space spectra (right) with the experiment in black and the calculated in red.

Once the program is done, a file called **EXAFS.png** containing Figure 6 will be produced. The options for this plot may be edited in `EXAFS_routines.py` under the **build_plots** function, if the options in the input file called “Plotting options” are not enough. It utilized the following important files as indicated by this excerpt of `EXAFS_routines.py` (lines 162 to 170). Note these files if you would like to graph the spectra in another program.

```

k_exp_x = np.loadtxt('{0}'.format(exp_file_name),usecols=range(0,1))
k_exp_y = np.loadtxt('{0}'.format(exp_file_name),usecols=range(1,2))
FT_exp_x = np.loadtxt('fe_FT_exp.out',usecols=range(0,1))
FT_exp_y = np.loadtxt('fe_FT_exp.out',usecols=range(1,2))
k_x = np.loadtxt('fe_chi_k3.DAT',usecols=range(0,1))
k_y = np.loadtxt('fe_chi_k3.DAT',usecols=range(1,2))
FT_x = np.loadtxt('fe_FT.out',usecols=range(0,1))
FT_y = np.loadtxt('fe_FT.out',usecols=range(1,2))

```

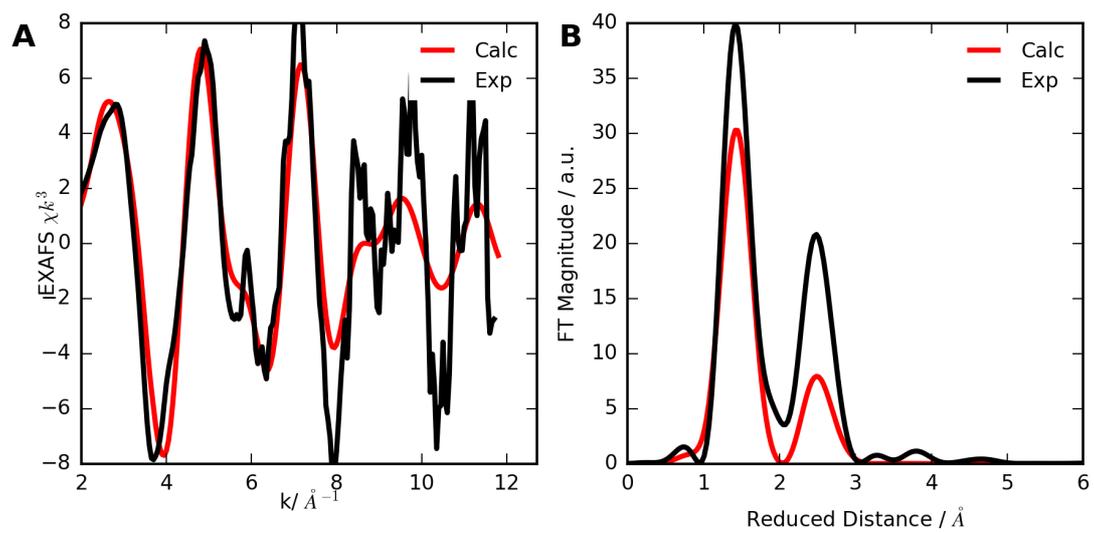


Figure 6. Publication quality version of Figure 5.

Section 6. Monte Carlo Refinement Code: Optmin.

This section will come out in the near future.

Section 7. References.

1. Ankudinov, A.; Bouldin, C.; Rehr, J.; Sims, J.; Hung, H., Parallel Calculation of Electron Multiple Scattering Using Lanczos Algorithms. *Phys. Rev. B* **2002**, *65* (10), 104107.
2. Ankudinov, A. L.; Ravel, B.; Rehr, J. J.; Conradson, S. D., Real-Space Multiple-Scattering Calculation and Interpretation of X-ray-Absorption Near-Edge Structure. *Phys. Rev. B* **1998**, *58* (12), 7565.
3. Ashley, C. A.; Doniach, S., Theory of Extended X-ray Absorption Edge Fine Structure (EXAFS) in Crystalline Solids. *Phys. Rev. B* **1975**, *11* (4), 1279.
4. Bouldin, C.; Sims, J.; Hung, H.; Rehr, J. J.; Ankudinov, A. L., Rapid Calculation of X-ray Absorption near Edge Structure Using Parallel Computation. *X-Ray Spectrom.* **2001**, *30* (6), 431-434.
5. Kronig, R. d. L., The Quantum Theory of Dispersion in Metallic Conductors. II. *Proc. R. Soc. London, Ser. A* **1931**, 255-265.
6. Kronig, R. d. L.; Penney, W. G. In *Quantum Mechanics of Electrons in Crystal Lattices*, Proc. R. Soc. London, Ser. A, The Royal Society: 1931; pp 499-513.
7. Lee, P. A.; Pendry, J. B., Theory of the Extended X-ray Absorption Fine Structure. *Phys. Rev. B* **1975**, *11* (8), 2795.
8. Sayers, D. E.; Stern, E. A.; Lytle, F. W., New Technique for Investigating Noncrystalline Structures: Fourier Analysis of the Extended X-ray Absorption Fine Structure. *Phys. Rev. Lett.* **1971**, *27* (18), 1204.
9. Stern, E. A., Theory of the Extended X-ray Absorption Fine Structure. *Phys. Rev. B* **1974**, *10* (8), 3027.
10. Askerka, M.; Ho, J.; Batista, E.; Gascón, J.; Batista, V., Chapter Seventeen-The MOD-QM/MM Method: Applications to Studies of Photosystem II and DNA G-Quadruplexes. *Methods Enzymol.* **2016**, *577*, 443-481.
11. Askerka, M.; Vinyard, D. J.; Brudvig, G. W.; Batista, V. S., NH₃ Binding to the S₂ State of the O₂-Evolving Complex of Photosystem II: Analogue to H₂O Binding during the S₂ → S₃ Transition. *Biochemistry* **2015**, *54* (38), 5783-5786.
12. Askerka, M.; Vinyard, D. J.; Wang, J.; Brudvig, G. W.; Batista, V. S., Analysis of the Radiation-Damage-Free X-Ray Structure of Photosystem II in Light of EXAFS and QM/MM Data. *Biochemistry* **2015**, *54* (9), 1713-1716.
13. Askerka, M.; Wang, J.; Brudvig, G. W.; Batista, V. S., Structural Changes in the Oxygen-Evolving Complex of Photosystem II Induced by the S₁ to S₂ Transition: A Combined XRD and QM/MM Study. *Biochemistry* **2014**, *53* (44), 6860-6862.
14. Askerka, M.; Wang, J.; Vinyard, D. J.; Brudvig, G. W.; Batista, V. S., S₃ State of the O₂-Evolving Complex of Photosystem II: Insights from QM/MM, EXAFS, and Femtosecond X-ray Diffraction. *Biochemistry* **2016**, *55* (7), 981-984.
15. Pal, R.; Negre, C. F.; Vogt, L.; Pokhrel, R.; Ertem, M. Z.; Brudvig, G. W.; Batista, V. S., S₀-State Model of the Oxygen-Evolving Complex of Photosystem II. *Biochemistry* **2013**, *52* (44), 7703-7706.
16. Sproviero, E. M.; Gascón, J. A.; McEvoy, J. P.; Brudvig, G. W.; Batista, V. S., A Model of the Oxygen-Evolving Center of Photosystem II Predicted by Structural Refinement Based on EXAFS Simulations. *J. Am. Chem. Soc.* **2008**, *130* (21),

6728-6730.

17. Vinyard, D. J.; Askerka, M.; Debus, R. J.; Batista, V. S.; Brudvig, G. W., Ammonia Binding in the Second Coordination Sphere of the Oxygen-Evolving Complex of Photosystem II. *Biochemistry* **2016**, *55* (31), 4432-4436.

18. Yang, K. R.; Matula, A. J.; Kwon, G.; Hong, J.; Sheehan, S. W.; Thomsen, J. M.; Brudvig, G. W.; Crabtree, R. H.; Tiede, D. M.; Chen, L. X.; Batista, V. S., Solution Structures of Highly Active Molecular Ir Water-Oxidation Catalysts from Density Functional Theory Combined with High-Energy X-ray Scattering and EXAFS Spectroscopy. *J. Am. Chem. Soc.* **2016**, *138* (17), 5511-5514.

19. Ravel, B.; Newville, M., ATHENA, ARTEMIS, HEPHAESTUS: Data Analysis for X-ray Absorption Spectroscopy using IFEFFIT. *J. Synchrotron. Radiat.* **2005**, *12* (4), 537-541.

20. Newville, M., IFEFFIT: interactive XAFS Analysis and FEFF Fitting. *J. Synchrotron. Radiat.* **2001**, *8* (2), 322-324.