CANDELA User's Manual

This is the manual for CANDELA, which was developed in Mohan Chen's research group in Peking University.

Outline

In Section 1, we briefly introduce the functions of CANDELA. In Section 2, we introduce how to obtain and install CANDELA. In Section 3, a list of examples is provided. In Section 4, some key parameters are explained in detail. Finally, in section 5, the code structure is introduced.

1. Introduction

CANDELA is the office name of Mohan Chen when he was a postdoc at Princeton University, where the code was created. The code was designed in a way to read in data from either Quantum Espresso, VASP, PROFESS, ABACUS, LAMMPS etc., most of the data are molecular dynamics trajectories. Next, CANDELA is mainly designed for post processing and provides results like radial distribution functions, bond angle distribution functions, diffusion coefficients, and many others. It is convenience for users who use multiple packages and would like to analyze the data in a same manner.

2. Download:

The code is stored in the bitbucket website and can be downloaded from the website with the following address:

https://mohanchen@bitbucket.org/mohanchen/Candela.git

3. Structure of the CANDELA Code

Blue: commonly used. Black: base modules.

Red: analysis for liquid water system. Purple: analysis for mechanical properties.

atoms.cpp, atoms.h	Define the class of atoms
bdf.cpp, bdf.h	Calculate the bond angle distribution functions
bdf_rcut.cpp, bdf_rcut.h	Calculate the bond angle distribution functions within a radius
	cutoff
cell.cpp, cell.h	Cell Information
cellFile.cpp, cellFile.h	Cell Files from various kinds of software
cellABACUS.cpp	Read in cell information from ABACUS
cellLAMMPS.cpp	Read in cell information from LAMMPS
cellPROFESS.cpp	Read in cell information from PROFESS
cellPWmat.cpp	Read in cell information from PWmat
cellQE2.cpp	Read in cell information from Quantum Espresso
cellQE.cpp	Read in cell information from Quantum Espresso

cellRAW.cpp	Read in cell information from RAW format
cellVASP.cpp	Read in cell information from VASP
cellXYZ.cpp	Read in cell information from XYZ format
const.h	Define constants
density2D.cpp, density2D.h	For water, to plot two dimensional figures regarding the
	relation between density and length of covalent bonds.
dielectric.cpp, dielectric.h	Compute dielectric constants for liquid water.
distri2D.cpp,	Compute angle-distance relation for liquid water (2D).
distri2D.h	
distri3D ions.cpp,	Compute 3D distribution of atoms/MLWFs around
distri3D ions.h	ions(hydroxide,Cl), read by VESTA
Doc	Documents
dsf.cpp, dsf.h	Dynamic Structure Factors
ele conductivity.cpp,	Compute the electron conductivity.
ele_conductivity.h	
examples	Include examples.
ext.cpp, ext.h	Extend the cells.
find ion.cpp, find ion.h	Find the ion positions in the interface of water-oil system.
gfun.cpp, gfun.h	Define globally used functions.
HBs.cpp, HBs.h	Define hydrogen bonds in liquid water.
Honeycutt.cpp	Honeycutt analysis (not used yet).
Honeycutt.h	
planarity.cpp,	Compute the hyper structures of OH- ion via the order
planarity.h	parameter named planarity.
ili_3D.cpp, ili_3D.h	Generate 3D data of Instantaneous Liquid Interface (ILI)
ili.cpp, ili.h	Compute Instantaneous Liquid Interface (ILI)
input.cpp, input.h	Input data.
insert.cpp, insert.h	Randomly insert atoms into existing structures.
iprof.cpp, iprof.h	Ionic density profile.
isf2.cpp, isf2.h	Compute intermediate scattering function, new by Qianrui.
isf.cpp, isf.h	Compute intermediate scattering function, old.
main.cpp	Main function of CANDELA package.
Makefile	Makfile of CANDELA package.
math.cpp, math.h	Math functions.
matrix3.cpp, matrix3.h	3-dimensional matrix format.
mdp2.cpp, mdp2.h	Compute mean density profile based on mean liquid interfaces.
mdp3.cpp, mdp3.h	Analysis based on Mean Density Profile and Instantaneous
	Liquid Interface
mdp.cpp, mdp.h	Compute mean density profile based on instantaneous liquid
	interface (ILI).
mj.cpp, mj.h	Provide proton transfer data.
movie_hexane.cpp,	Print movie data for hexane project.
movie hexane.h	

msd_pmtiple.cpp, Compute the mean square displacements. msd_multiple.cpp, Compute the mean square displacements through multiple trajectories. pdf2d.cpp, pdf2d.dh PPF for 2d materials. pdf3c.cpp, pdf3.h Compute PDF for each shell of neighboring atoms. pdf added.cpp, pdf added.h The extra functions used in computing the PDF of systems. pdf.cpp, pdf3h Compute the radial distribution functions. powers.cpp, powers.h Compute the presolvation structure (yes/no). presolvation.cpp, presolvation.h Compute the andom numbers. README.md Change log, important. reorganize.cpp, reorganize.h Read in binary data. ssf.cpp, ssf.h Compute static structure factors ssf selected.cpp, sef selected.h Compute static structure factors for selected q points. tetra_order.cpp, tetra_order.h Velocity autocorrelation functions. vec3.h Vector3 format. vec3.h Vector3 format. vec1.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, waterwire2.h Compute the distribution of velocities. vaterwire.cpp, waterwire1.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire1.h Wave functions, added by Qianrui wffe.cpp, wffel.h Wave functions, added by Qianrui wffe.cpp, wffe.h Wave functions, added by Qianrui wffe.cpp, wffe.h Wave functions, added by Qianrui wre.compress.h xsf.pp, xsf.h Compute the properties from 3D data in XSF format.		·
msd_multiple.h pdf2d.cpp, pdf2d.h pdf3cd.cpp, pdf2d.h pdf3cd.cpp, pdf3ch pdf_added.cpp, pdf_added.h pdf_added.cpp, pdf_added.h pdf_cpp, pdf.h Compute the radial distribution functions. powers.cpp, powers.h Compute the power spectra of system (FFT of VAF). presolvation.cpp, presolvation.h local_pseudopotential.cpp, local_pseudopotential.h random.h Compute the random numbers. README.md Change log. important. reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf_cpp, ssf_h Compute static structure factors ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vecum.cpp, vel.com vecl.cpp, vel.h vel.compute the distribution of velocities. void.cpp, void.h Crate a void in a material. wannierl.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wavefunc.cpp, waterwire.h wavefunc.cpp, waterwire.h wavefunc.cpp, waterwire.h wavefunc.cpp, wffel.h Wave functions, added by Qianrui. wffe.cpp, wffel.h Wave functions, added by Qianrui. wffe.cpp, wffel.h Wave functions, added by Qianrui. wffe.cpp, wffel.h Wave functions, added by Qianrui. ww_compress.h wave_compress.h wavefunc.pp, wiffe.h Wave functions, added by Qianrui. ww_compress.h wave_compress.h	msd.cpp, msd.h	Compute the mean square displacements.
pdf2d.cpp, pdf2d.h pdf5.cpp, pdf5.h pdf added.cpp, pdf added.h pdf.cpp, pdfh pdf.cpp, pdfh Compute the radial distribution functions. Compute the presolvation structure (yes/no). Presolvation.cpp, presolvation.h local pseudopotential.cpp, local pseudopotential.h random.h README.md Compute the random numbers. Compute the input trajectories through nbin, binftream.cpp, sif.h ssf.cpp, ssf.h tetra order.cpp, tetra order.h vacuum.cpp, vacuum.h vec3.h velcort.pp, velcort.h velcort.pp, vold.h vol.cpp, wannier1.h Compute the distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, waterwire2.h Waterwire2.cpp, waterwire3.h Wave functions, added by Qianrui. wffe.cpp, wffe.h Wave functions, added by Qianrui. wffe.cpp, wffe.h Wave compress.h Wave functions, added by Qianrui. www_compress.h	msd_multiple.cpp,	Compute the mean square displacements through multiple
pdf5.epp, pdf5.h pdf_added.epp, pdf_added.h pdf_added.epp, pdf_added.h pdf.epp, pdf.h Compute the radial distribution functions. Compute the power spectra of system (FFT of VAF). Compute the power spectra of system (FFT of VAF). Compute the power spectra of system (FFT of VAF). Compute the presolvation structure (yes/no). Compute the presolvation structure (yes/no). Compute the presolvation structure (yes/no). Compute the random numbers. Change log, important. Compute the random numbers. Change log, important. Compute static structure factors ssf_selected.epp, ssf_selected.h tetra_order.epp, tetra_order.h vacuum.epp, vacuum.h Add vacuum for a structure. vec3.h velcor.pp, velcor.h velcor.pp, vel.h Compute the distribution of velocities. void.epp, void.h vannier1.epp, wannier1.h Compute Infrared Spectra from Wannier functions, including 1D, 2D, and 3D. wannier.epp, waterwire2.h waterwire2.epp, waterwire1.h Waterwire2.epp, waterwire2.h waterwire.epp, waterwire1.h Waterwire.epp, waterwire2.h waterwire.epp, waterwire1.h Waterwire.epp, wffile.h wavefunc.epp, wffile.h Wave functions, added by Qianrui. wffe.epp, wffe.ah Wave functions, added by Qianrui. ww_compress.h Compute the compression of water wires.	msd_multiple.h	trajectories.
pdf_added.cpp, pdf_added.h pdf.cpp, pdf.h pdf.cpp, pdf.h pdf.cpp, pdf.h powers.cpp, powers.h presolvation.cpp, presolvation.cpp local pseudopotential.cpp, local pseudopotential.h random.h Compute the form of the readin pseudopotential. Compute the random numbers. README.md Compute the input trajectories through nbin, binfream.cpp, binfstream.h ssf.cpp, ssf.h Compute static structure factors ssf selected.cpp, ssf selected.h Tetrahedral order parameter computed for each water molecule. Vector3 format. velcor, pv. el.h vel.cpp, vel.h void.cpp, void.h Wannier1.cpp, water.h waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wafe in many data in a material. Compute the distributions of Wannier functions, including 1D, 2D, and 3D. Wanefunctions and double proton transfer. wavefunc.cpp, waterwire.h Wave functions, added by Qianrui wfred.cpp, wffe.h Wave functions, added by Qianrui	pdf2d.cpp, pdf2d.h	PDF for 2d materials.
pdE.cpp, pdE.h powers.cpp, powers.h Compute the radial distribution functions. Compute the power spectra of system (FFT of VAF). Compute the power spectra of system (FFT of VAF). Compute the presolvation structure (yes/no). Change the form of the readin pseudopotential. random.h Compute the random numbers. README.md Change log, important. reorganize cpp, reorganize.h binfiream.cpp, binfstream.h ssf.cpp, ssf.h Ssf.cpp, ssf.h Compute static structure factors ssf_selected.pp, ssf_selected.h tetra_order.pp, tetra_order.h veauum.cpp, vacuum.h Vetor3 format. velcor.cpp, velor.h velcor.pp, vel.h Compute the distribution of velocities. void.cpp, void.h wannier1.epp, wannier1.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h Waterwire.cpp, waterwire.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui write.cpp, wfile.h Wave functions, added by Qianrui write.cpp, write.h Wave functions, added by Qianrui wrecompress.cpp, ww compress.ch	pdf5.cpp, pdf5.h	Compute PDF for each shell of neighboring atoms.
powers.cpp, powers.h presolvation.cpp, presolvation.h local_pseudopotential.cpp, local_pseudopotential.h random.h Change log, important. reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf.cpp, ssf.h Ssf selected.cpp, stf selected.h tetra_order.cpp, tetra_order.h vee3.h Vector3 format. vee03.h Velcopp, vel.h Void.cpp, void.h Wannierl.cpp, wannierl.h Wannierl.cpp, waterwire2.h waterwire2.cpp, waterwire2.h waterwire2.cpp, waterwire2.h waterwire2.cpp, waterwire.h wffe.cpp, wffe.h Wave functions, added by Qianrui wfred.cpp, wffe.h Wave functions, added by Qianrui wfred.cpp, wffe.h Wave functions, added by Qianrui wfred.compp, wffee.h Wave compress.h Wave functions, added by Qianrui wfred.cpp, wffee.h Wave compress.cpp, ww_compress.h	pdf_added.cpp, pdf_added.h	The extra functions used in computing the PDF of systems.
presolvation.cpp, presolvation.b local_pseudopotential.cpp, local_pseudopotential.h random.h Compute the random numbers. Change log, important. reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf.cpp, ssf.h Compute static structure factors ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vec3.h Vector3 format. velcor.cpp, velcor.h vel.cpp, void.h wannierl.cpp, wannierl.h Compute the distribution of velocities. vauum.cpp, vauum.h Compute the distribution of velocities. void.cpp, void.h wannier.cpp, water.h waterwire2.cpp, waterwire2.h waterwire2.cpp, waterwire.h waterwire.cpp, waterwire.h Wave functions, added by Qianrui. wffeile.cpp, wffe.h Wave functions, added by Qianrui. wffeel.cpp, wffe.h Wave functions, added by Qianrui. wffeel.cpp, wffe.h Wave functions, added by Qianrui. wffeucompress.h Compute the compression of water wires.	pdf.cpp, pdf.h	Compute the radial distribution functions.
presolvation.h local_pseudopotential.cpp, local_pseudopotential.h random.h Compute the random numbers. README.md reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf_cpp, ssf.h ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vacuum.cpp, vacuum.h vec3.h velcor.cpp, velcor.h vel.cpp, vel.h void.cpp, void.h wannier1.cpp, wannier1.h wannier2.cpp, waterwire2.h waterwire2.cpp, waterwire2.h waterwire2.cpp, waterwire.h wavefunc.cpp, wavefunc.h wffe.cpp, wffel.h wffe.cpp, wffeed.h wffe.cdpp, wffeed.h wffe.cdpp, wffeed.h wffeed.cpp, wffeed.h wavefunctomp, wffeed.h wffeed.pp, wffeed.h wffeed.dp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wffeed.pp, wffeed.h wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wffeed.pp, wffeed.h wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h wffeed.pp, wffeed.h wffeed.pp, wffeed.h wave functions, added by Qianrui. wffeed.pp, wffeed.h	powers.cpp, powers.h	Compute the power spectra of system (FFT of VAF).
local_pseudopotential.cpp, local_pseudopotential.h random.h README.md Change log, important. reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf.cpp, ssf.h Compute static structure factors ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vacuum.cpp, vacuum.h vec3.h velcor.cpp, velcor.h velcor.pp, velch void.cpp, void.h wannier1.cpp, wannier1.h wannier2.cpp, water.h waterwire2.cpp, waterwire.h waterwire2.cpp, waterwire.h waterwire.cpp, wavefunc.h wavefunc.cpp, wavefunc.h wffe.cpp, wffe.h wffe.cdpp, wffe.ah wffe.cdpp, wffe.ah wffe.cdpp, wffe.ah wffe.cdpp, wffe.ah wffe.cdpp, wffe.ah wavefunctomp, vffes.h wffe.compute. Wave functions, added by Qianrui. wffe.cdpp, wffe.ah wavefunctomp, wffe.sh wffe.compute. Wave functions, added by Qianrui. wffe.comp, wffe.sh wavefunctomp, wffes.h wffe.compress.h Compute the compression of water wires.	presolvation.cpp,	Compute the presolvation structure (yes/no).
local_pseudopotential.h Compute the random numbers.	presolvation.h	
random.h Compute the random numbers. README.md Change log, important. reorganize.cpp, reorganize.h Reorganize the input trajectories through nbin, binftream.cpp, binfstream.h Read in binary data. ssf.cpp, ssf.h Compute static structure factors ssf_selected.cpp, ssf_selected.h Compute static structure factors for selected q points. tetra_order.cpp, tetra_order.h Tetrahedral order parameter computed for each water molecule. vec3.h Vector3 format. velcor.cpp, velcor.h Velocity autocorrelation functions. vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPend.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, wite.h Wave functions, added by Qianrui. wrompress.cpp, w_compress.h	local_pseudopotential.cpp,	Change the form of the readin pseudopotential.
README.md Change log, important. reorganize.cpp, reorganize.h Reorganize the input trajectories through nbin, binftream.cpp, binfstream.h Read in binary data. ssf.cpp, ssf.h Compute static structure factors ssf_selected.cpp, ssf_selected.h Compute static structure factors for selected q points. tetra_order.cpp, tetra_order.h Tetrahedral order parameter computed for each water molecule. vec3.h Vector3 format. velcor.cpp, velcor.h Velocity autocorrelation functions. vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute listributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, wavefunc.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPed.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. wr_compress.h Compute the compression of water wires.	local_pseudopotential.h	
reorganize.cpp, reorganize.h binftream.cpp, binfstream.h ssf.cpp, ssf.h ssf.pp, ssf.h compute static structure factors ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vacuum.cpp, vacuum.h vec3.h velcor.cpp, velcor.h vel.cpp, vel.h compute the distribution of velocities. void.cpp, void.h create a void in a material. wannier1.cpp, wannier1.h compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h compute the free energy map of waterwire involving single and double proton transfer. wavefunc.cpp, wavefunc.h wavefunc.cpp, wafel.h wavefunc.cpp, wfPWmat.h wfqe.cpp, wfRead.h wavefunctions, added by Qianrui wfRead.cpp, wfRead.h wavefunctions, added by Qianrui wfRead.cpp, wfRead.h wavecompress.h Compute the compression of water wires.	random.h	Compute the random numbers.
binftream.cpp, binfstream.h ssf.cpp, ssf.h Compute static structure factors ssf_selected.cpp, ssf_selected.h Compute static structure factors for selected q points. tetra_order.cpp, tetra_order.h vacuum.cpp, vacuum.h vec3.h Vector3 format. velcor.cpp, vel.h Velocity autocorrelation functions. vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui wfQe.cpp, wfRead.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui wrecompress.h Compute the compression of water wires.	README.md	Change log, important.
ssf.cpp, ssf.h ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h vec3.h velcor.cpp, velcor.h velcor.cpp, velcor.h vel.cpp, vel.h void.cpp, void.h wannierl.cpp, wannierl.h water.cpp, water.h water.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, wafe.h wavefunc.cpp, wffel.h wffel.cpp, wffel.h wffel.cpp, wffel.h wffel.cpp, wffel.h wffel.cpp, wffel.h wavefunc.cpp, wffeel.h wavefunctions, added by Qianrui wffee.cpp, wffeel.h Wave functions, added by Qianrui ww_compress.cpp, Compute the compression of water wires.	reorganize.cpp, reorganize.h	Reorganize the input trajectories through nbin,
ssf_selected.cpp, ssf_selected.h tetra_order.cpp, tetra_order.h tetra_order.cpp, tetra_order.h vec3.h velcor.cpp, velcor.h vel.cpp, vel.h void.cpp, void.h wannier.cpp, wannier.h water.cpp, water.h waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wavefunc.cpp, wavefunc.h wfile.cpp, wfile.h wfile.cpp, wfile.h wfile.cpp, wfile.h wfile.cpp, wfile.h wfile.cpp, wfile.h wine compress.cpp, wite.h Compute the distributior functions. Velocity autocorrelation functions. Velocity autocare functions. Velocity autocare	binftream.cpp, binfstream.h	Read in binary data.
tetra_order.cpp, tetra_order.h vecuum.cpp, vacuum.h Vector3 format. veco.cpp, velcor.h vel.cpp, vel.h void.cpp, void.h vannier1.cpp, wannier1.h vater.cpp, water.h vaterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wavefunc.cpp, wffle.h wavefunctions, added by Qianrui. wfflea.cpp, wffead.h wfflea.cpp, wffead.h ware redactive distributions, added by Qianrui. wavefunctions, put redactive distributions, added by Qianrui. wavefunctions, added by Qianrui.	ssf.cpp, ssf.h	Compute static structure factors
vacuum.cpp, vacuum.h Add vacuum for a structure. vec3.h Vector3 format. velcor.cpp, velcor.h Velocity autocorrelation functions. vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. wrie.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires.	ssf_selected.cpp, ssf_selected.h	Compute static structure factors for selected q points.
vec3.h velcor.cpp, velcor.h vel.cpp, vel.h vel.cpp, void.h venic.cpp, void.h venic.cpp, wannier1.h vannier1.cpp, wannier1.h vannier.cpp, wannier.h vater.cpp, water.h vaterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wave functions, added by Qianrui. wfFile.cpp, wffe.h wave functions, added by Qianrui. wfe.cpp, wfe.h Wave functions, added by Qianrui. wfe.cpp, wfread.h wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. wave compress.cpp, wavecompress.h	tetra_order.cpp, tetra_order.h	Tetrahedral order parameter computed for each water molecule.
velcor.cpp, velcor.h Velocity autocorrelation functions. vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire.h Compute the free energy map of waterwire involving single and double proton transfer. wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires.	vacuum.cpp, vacuum.h	Add vacuum for a structure.
vel.cpp, vel.h Compute the distribution of velocities. void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire.h Compute the free energy map of waterwire involving single and double proton transfer. wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfQ-cpp, wfqe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires.	vec3.h	Vector3 format.
void.cpp, void.h Create a void in a material. wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires.	velcor.cpp, velcor.h	Velocity autocorrelation functions.
wannier1.cpp, wannier1.h Compute distributions of Wannier functions, including 1D, 2D, and 3D. Wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. Water.cpp, water.h Construct a water class for each read-in water. Compute the free energy map of waterwire involving single and double proton transfer. Waterwire.cpp, waterwire.h Wave functions, added by Qianrui WfFile.cpp, wfFile.h Wave functions, added by Qianrui. WfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui wfqe.cpp, wfqe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. Compute the compression of water wires.	vel.cpp, vel.h	Compute the distribution of velocities.
wannier.cpp, wannier.h Compute Infrared Spectra from Wannier functions, plot distribution of dipoles. water.cpp, water.h Construct a water class for each read-in water. waterwire2.cpp, waterwire2.h Compute the free energy map of waterwire involving single and double proton transfer. waterwire.cpp, waterwire.h Compute the free energy map of waterwire involving single and double proton transfer. wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfQe.cpp, wfQe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. Wave functions, added by Qianrui. Compute the compression of water wires.	void.cpp, void.h	Create a void in a material.
distribution of dipoles. water.cpp, water.h waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wavefunc.cpp, wavefunc.h wfFile.cpp, wfFile.h wfPWmat.cpp, wfPWmat.h wfqe.cpp, wfqe.h wfRead.cpp, wfRead.h write.cpp, write.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Compute the compression of water wires. ww_compress.h	wannier1.cpp, wannier1.h	
waterwire2.cpp, waterwire2.h waterwire.cpp, waterwire.h waterwire.cpp, waterwire.h wavefunc.cpp, wavefunc.h wfFile.cpp, wfFile.h wfPWmat.cpp, wfPWmat.h wfRead.cpp, wfRead.h wire.cpp, write.h ww_compress.cpp, ww_compress.h Compute the free energy map of waterwire involving single and double proton transfer. Wave functions, added by Qianrui Wave functions, added by Qianrui. Wave functions, added by Qianrui. Wave functions, added by Qianrui Wave functions, added by Qianrui. Wave functions, added by Qianrui. Compute the compression of water wires.	wannier.cpp, wannier.h	
and double proton transfer. Waterwire.cpp, waterwire.h Compute the free energy map of waterwire involving single and double proton transfer. Wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui WfPile.cpp, wfFile.h Wave functions, added by Qianrui. WfPwmat.cpp, wfPwmat.h Wave functions, added by Qianrui. Wfqe.cpp, wfqe.h Wave functions, added by Qianrui WfRead.cpp, wfRead.h Wave functions, added by Qianrui. Write.cpp, write.h Wave functions, added by Qianrui. Wave functions, added by Qianrui. Compute the compression of water wires.	water.cpp, water.h	Construct a water class for each read-in water.
waterwire.cpp, waterwire.h Compute the free energy map of waterwire involving single and double proton transfer. Wavefunc.cpp, wavefunc.h Wave functions, added by Qianrui. WfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. Wfqe.cpp, wfqe.h Wave functions, added by Qianrui. WfRead.cpp, wfRead.h Wave functions, added by Qianrui. Write.cpp, write.h Wave functions, added by Qianrui. Wave functions, added by Qianrui. Wave functions, added by Qianrui. Compute the compression of water wires.	waterwire2.cpp, waterwire2.h	Compute the free energy map of waterwire involving single
and double proton transfer. wavefunc.cpp, wavefunc.h wfFile.cpp, wfFile.h wave functions, added by Qianrui. wfPWmat.cpp, wfPWmat.h wfqe.cpp, wfqe.h wfRead.cpp, wfRead.h write.cpp, write.h ww_compress.cpp, ww_compress.h wavefunctions, added by Qianrui. wavefunctions, added by Qianrui. wavefunctions, added by Qianrui. wavefunctions, added by Qianrui. Compute the compression of water wires.		and double proton transfer.
wavefunc.cpp, wavefunc.h wfFile.cpp, wfFile.h wfPWmat.cpp, wfPWmat.h wfqe.cpp, wfqe.h wfRead.cpp, wfRead.h write.cpp, write.h wave functions, added by Qianrui. Wave functions, added by Qianrui. wfqe.cpp, wfqe.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. write.cpp, write.h Compute the compression of water wires. ww_compress.h	waterwire.cpp, waterwire.h	Compute the free energy map of waterwire involving single
wfFile.cpp, wfFile.h Wave functions, added by Qianrui. wfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. wfqe.cpp, wfqe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires. ww_compress.h		and double proton transfer.
wfPWmat.cpp, wfPWmat.h Wave functions, added by Qianrui. wfqe.cpp, wfqe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires. ww_compress.h Compute the compression of water wires.	wavefunc.cpp, wavefunc.h	Wave functions, added by Qianrui
wfqe.cpp, wfqe.h Wave functions, added by Qianrui wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires. ww_compress.h Compute the compression of water wires.	wfFile.cpp, wfFile.h	Wave functions, added by Qianrui.
wfRead.cpp, wfRead.h Wave functions, added by Qianrui. write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires. ww_compress.h	wfPWmat.cpp, wfPWmat.h	Wave functions, added by Qianrui.
write.cpp, write.h Wave functions, added by Qianrui. ww_compress.cpp, Compute the compression of water wires. ww_compress.h	wfqe.cpp, wfqe.h	Wave functions, added by Qianrui
ww_compress.cpp, Compute the compression of water wires. ww_compress.h	wfRead.cpp, wfRead.h	Wave functions, added by Qianrui.
ww_compress.h	write.cpp, write.h	Wave functions, added by Qianrui.
	ww_compress.cpp,	Compute the compression of water wires.
	ww_compress.h	
	xsf.cpp, xsf.h	Compute the properties from 3D data in XSF format.

4. List of Examples:

- 1) Radial distribution functions for O-O with varying cells
- 2) Bond angle distribution function for O triplets with varying cells
- 3) Bond angle distribution function for O triplets with cutoff in varying cells
- 4) Hydrogen bond analysis
- 5) Mean square displacement
- 6) Mean square displacement computed by multiple sections
- 7) Wannier centers and dipoles
- 8) Tetrahedrality
- 9) Reorganize
- 10) Radial distribution function of O*H in hydroxide solution
- 11) Hydrogen bonds for hydroxide
- 12) Multiple jumps
- 13) Planarity of hydroxide ion and its neighbors
- 14) Movie
- 15) Distance 2D
- 16) XSF 3D
- 17) Distance 2D Wannier
- 18) PDF O(donate)-O(accept)
- 19) Presolvation
- 20) Hydroxide Wannier
- 21) Waterwire2
- 22) Waterwire compress
- 23) Static Structure Factor
- 24) Instantaneous Liquid Interface
- 25) Infrared Spectra
- 27) Dynamic Structure Factor
- 28) Electrical Conductivity

4.1) Radial distribution functions for O-O with varying cells

```
calculation pdf # Pair Distribution Function.
system water
geo in type QE
geo_directory ../SCAN_volume/water.pos
cell file ../SCAN volume/water.cel
geo_1 1
geo 2 122517
geo interval 1
geo ignore 20836 ! first 5 ps
geo_out pdf.txt # output pdf name.
ntype 2 # number of different types of atoms.
          192
                  # total number of atoms.
natom
natom1
          64
natom2
         128
0.01
dr
                 # delta r in real space
rcut 6.22 # real space cutoff
id1 0
id2 H
ele1 0
ele2 O
struf dgx 0.05
struf_ng 480
```

4.2) Bond Angle distribution function for O triplets with varying cells

```
calculation bdf_rcut
system water
geo in type QE #PROFESS/VASP/QE/ABINIT/MESIA/XYZ/PIMD
geo_directory ../SCAN_volume/water.pos
cell file ../SCAN volume/water.cel
geo_1 1
geo 2
          122517
geo interval 1
geo ignore 20836 ! ignore first 5 ps
                    # number of different types of atoms.
ntype
          2
          192
                   # total number of atoms.
natom
natom1
          64
natom2
          128
      0.01 # delta r in real space
dr
id1 0
id2 H
rcut1
      4.0
bdf_rcut 3.154
bdf dtheta 0.5 # d(theta) for degree between (0,180]
ele1 0
ele2 O
func 1
factor 0.9834679045
x0 2.25
у0 40
nx 50
ny 70
dx 0.025
dy 2.0
```

4.3) Bond Angle distribution function for O triplets with cutoff in varying cells

Notes: if func_b==1, do nothing; func_b==2, both bonded; func_b==3, both are not bonded; func_b==4, one bonded, the other not bonded.

```
calculation bdf rcut
system water
               #PROFESS/VASP/QE/ABINIT/MESIA/XYZ/PIMD
geo in type QE
geo_directory ../../SCAN_volume/water.pos
cell file ../../SCAN volume/water.cel
geo 1
            1
geo 2
            122517
geo interval 1
geo_ignore 20836 ! first 5 ps
            2
                      # number of different types of atoms.
ntype
           192
                    # total number of atoms.
natom
natom1
           64
natom2
           128
dr
        0.01
                 # delta r in real space
id1 0
id2 H
rcut1
            4.0
bdf rcut
            3.154
                    # cutoff for O triplets
bdf out
            3.154.dat # output file name
bdf dtheta
            0.5 # d(theta) for degree between (0,180]
ele1 0
ele2 O
func b 2
factor 1.1542012927
x0 2.25
y0 40
nx 50
ny 70
dx 0.02
dy 2.0
```

4.4) Hydrogen bond analysis

```
calculation hbs # hydrogen bond analysis
system water
                  #PROFESS/VASP/QE/ABINIT/ABACUS
geo in type QE
geo_directory ../SCAN_volume/water.pos
cell file ../SCAN volume/water.cel
            PBE0_OH_HBs.dat
geo_out
geo 1
             1
geo 2
             122517
geo interval 1
                  # geometry selected with this interval
geo ignore 20836
                        # number of different types of atoms.
             2
ntype
                        # total number of atoms.
natom
             192
natom1 64 # 0
natom2 128 # H
id1 0
id2 H
rcut oo 3.5
rcut oh 1.24
acut_hoo 30
------ FOR LAMMPS ------
calculation hbs # hydrogen bond analysis
system water
geo_in_type LAMMPS
geo directory ../dump.lammpstrj
geo 1
           1
           100 40001
geo 2
geo interval 1
geo_ignore
          0! first 5 ps
           2
                    # number of different types of atoms.
ntype
           96
                    # total number of atoms.
natom
natom1 32 # O
natom2 64 # H
id1 O
id2 H
rcut oo 3.5
```

rcut_oh 1.24 acut_hoo 30

cartesian 1

4.5) Mean square displacement

```
calculation msd
                  # pair Distribution Function.
system water
geo in type QE #PROFESS/VASP/QE/ABINIT/MESIA
geo_directory ../SCAN_volume/water.pos
cell file ../SCAN volume/water.cel
geo_1
          1
geo 2 122517
geo interval 1
geo_ignore 0
geo_out pdf.txt # output pdf name.
                   # number of different types of atoms.
ntype 2
          192
                  # total number of atoms.
natom
natom1
natom2
          64
          128
      0.01 # delta r in real space
6.22 # real space cutoff
dr
rcut
id1 0
id2 H
ele1 0
ele2 O
struf dgx 0.05
struf_ng 480
func 2
cartesian 1
msd dt 0.005
```

4.6) Mean square displacement computed by multiple sections

Example 1 for QE and Example 2 for LAMMPS

```
calculation msd multiple # mean square displacements
system hydroxide
geo in type QE #PROFESS/VASP/QE/ABINIT/MESIA
geo directory ../SCAN volume/water.pos
cell_file ../SCAN_volume/water.cel
           1
geo 1
geo 2
           122517
geo interval 1
ntype
                    # number of different types of atoms.
                   # total number of atoms.
          192
natom
natom1 64 # 0
natom2 128 # H
id1 0
id2 H
rcut oo 3.5
rcut oh 1.24
acut hoo 30
msd n 5 # number of msd needed
msd t0 5 # starting point of msd
msd t 12.20 # length of msd (in ps)
msd dt0 3 # difference between different different MSD
msd dt .00084661 # delta t between 2 snapshots
msd natom 192
msd stokes 0
system water
```

```
calculation msd_multiple # mean square displacements
system water
geo_in_type LAMMPS
geo_directory ../lmp.nve.64/dump.lammpstrj
geo_1     1
geo_2     200001
geo_interval 1
geo_ignore     1000 ! first 5 ps

ntype 2 # number of different types of atoms.
```

```
natom 192 # total number of atoms.
natom1 64 # 0
natom2 128 # H
id1 0
id2 H
rcut oo 3.5
rcut oh 1.24
acut_hoo 30
msd n 10 # number of msd needed
msd t0 0 # starting point of msd
msd t 100 # length of msd (in ps)
msd dt0 100 # difference between different different MSD
msd_dt 0.005 # delta t between 2 snapshots (ps)
msd natom 192
msd_stokes 0
calculation msd multiple
geo in type PROFESS
geo directory ../out/
geo_1 4000
geo_2 400000
geo interval 4
geo ignore 0
ntype 2
natom 128
natom1 118
natom2 10
id1 L7
id2 L6
msd single 1 #1: open msd for only one type 0: close(by default)
msd type 0  # type id (starts from 0)
msd natom 10
msd_n 10
msd t0 1
{\tt msd} t 10
msd dt0 10
msd dt 0.01
msd stokes 0
```

dt snapshots 0.00025

4.7) Wannier centers and dipoles

```
calculation wannier # pair Distribution Function.
 system water
 geo in type QE # input type of geometry file:
PROFESS/VASP/QE/ABINIT/MESIA
 geo directory ../SCAN volume/water.pos
 cell_file ../SCAN_volume/water.cel
 wannier file ../SCAN volume/water.wfc
            1
 geo 1
 geo 2
            122517
 geo interval 1
 geo_ignore 20836
 ntype
                     # number of different types of atoms.
                   # total number of atoms.
           192
 natom
 natom1
           64
 natom2
           128
 id1 0
 id2 H
 ele1 0
 nbands 256
 dr 0.001
 rcut 1.00
 dz 0.02 # for dipole moment
 rcut1 10.0 # for dipole moment
 nx 70
 ny 70
 x0 0.95
 y0 0.25
 dx 0.0025
 dy 0.005
```

4.8) Tetrahedrality

```
calculation top # hydrogen bond analysis
 system hydroxide
 geo in type QE # input type of geometry file:
PROFESS/VASP/QE/ABINIT/MESIA
 geo_directory ../SCAN_volume/water.pos
 cell_file ../SCAN_volume/water.cel
 geo 1
            1
 geo 2
           122517
 geo interval 1
 geo ignore 20836 ! first 5 ps
                    # number of different types of atoms.
 ntype
           192
 natom
                  # total number of atoms.
           64
 natom1
 natom2 128
 id1 0
 id2 H
 rcut oo 3.5
 rcut_oh 1.24
 acut hoo 30
 bdf rcut 3.15
```

4.9) Reorganize

```
calculation reorganize
system water # system is water
geo in type QE  # PROFESS/VASP/QE/ABINIT/MESIA
geo_directory ../SCAN_volume/water.pos
wannier file ../SCAN volume/water.wfc
cell_file ../SCAN_volume/water.cel
geo 1
          1
geo 2
          122517
geo interval 1
geo_ignore 20836 ! first 5 ps
                  # number of different types of atoms.
ntype
natom
         192
                 # total number of atoms.
          64
natom1
natom2 128
id1 0
id2 H
nbands 256 # number of bands for the system
nbin 10
```

4.10) Radial distribution function of O*H in hydroxide solution

```
calculation pdf
                   # pair Distribution Function.
system hydroxide
geo in type QE # PROFESS/VASP/QE/ABINIT/MESIA
geo_directory OH_PBE0_vdW.pos
geo 1
          1
geo_2
          95306
geo interval 1
geo out    pdf.txt # output pdf name.
                  # number of different types of atoms.
ntype
natom
         191
                  # total number of atoms.
natom1
          64
natom2
dr
          127
         0.002 # delta r in real space
         6.2 # real space cutoff
rcut
id1 0
id2 H
ele1 0
ele2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
```

4.11) Hydrogen bonds for hydroxide

```
calculation hbs # hydrogen bond analysis
system hydroxide
geo in type QE # PROFESS/VASP/QE/ABINIT/MESIA
geo_directory OH_PBE0_vdW.pos
geo out
         PBE0 OH HBs.dat
geo_1
         1
geo 2
         95306
geo\_interval 1  # pick up geometry with this interval
ntype
         2 # number of different types of atoms.
         191 # total number of atoms.
natom
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut oo 3.5
rcut_oh 1.24
acut hoo 30
```

4.12) Multiple jumps

```
calculation mj # multiple jump
system hydroxide
geo_in_type QE #PROFESS/VASP/QE/ABINIT/ABACUS
geo_2
         87
ntype
         2 # number of different types of atoms.
natom 191 # total number of atoms.
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
func 2
```

4.13) Planarity of hydroxide ion and its neighbors

Note: If delta is not set, then the criterion is not used.

Nacc set to 4 suggests that the hyper coordinated structures are selected,

Ndon set to 1 suggests that only those water molecules who donate 1 HB are selected.

```
calculation hyper # instantaneous liquid interfaces
system hydroxide
geo_in_type QE # PROFESS/VASP/QE/ABINIT/MESIA
geo directory OH- PBEO vdW.pos
geo out
          hyper.dat
          1
geo 1
geo 2
          95306 #
geo_interval 1 # pick up geometry with this interval
                    # number of different types of atoms.
           2
ntype
           191
                    # total number of atoms.
natom
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut oo 3.5
rcut oh 1.24
acut hoo 30
rcut 6.0
dr 0.1
nacc 4
delta -0.1
```

4. 14) Movie

```
calculation movie # instantaneous liquid interfaces
geo_in_type QE #PROFESS/VASP/QE/ABINIT/ABACUS
geo_directory OH-_PBE0_vdW.pos
geo_out
          movie.xyz
geo 1
          1
geo_2
          62338
geo interval 62338
          2 # number of different types of atoms.
ntype
          191 # total number of atoms.
natom
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
snatom 5 49 15 16 58 62
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509 \# celldm3 in Angstrom
func 2
```

4.15) Distance 2D

```
calculation dist
 system hydroxide
 geo_in_type QE
 geo_directory OH-_PBE0_vdW.pos
 geo 1
           1
 geo_2
            95306
 geo interval 1
                   # number of different types of atoms.
 ntype
 natom 191 # total number of atoms.
 natom1 64 # 0
 natom2 127 # H
 id1 0
 id2 H
 celldm1 12.444655509
 celldm2 12.444655509
 celldm3 12.444655509
 rcut oo 3.5
 rcut_oh 1.24
 acut hoo 30
 rcut 12.0
 rcut1 4.0
 dr 0.1
 nx 61
 ny 21
 u1 41
 u2 41
 u3 61
 func 1
ele1 O
```

4.16) XSF 3D

Nacc40-plot the *.xsf file (3D pot)

```
calculation dist
system hydroxide
geo_in_type QE
geo directory OH- PBEO vdW.pos
geo_1
          1
geo 2
          95306
geo interval 1
ntype
                 # number of different types of atoms.
          191 # total number of atoms.
natom
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut oo 3.5
rcut_oh 1.24
acut hoo 30
rcut 12.0
rcut1 4.0
dr 0.1
nx 61
ny 41
nacc 40
func 2
u1 41
u2 41
u3 61
ele1 0
```

Add the following header in the front part of dist3D.dat file, change the name to 3d.xsf

```
CRYSTAL
```

PRIMVEC

8.0 0 0

0 8.0 0

0 0 12.0

PRIMCOORD

2 1

0 4.0 4.0 6.0

H 4.0 4.0 7.0

BEGIN_BLOCK_DATAGRID_3D

3D_PWSCF

DATAGRID_3D_UNKNOWN

41 41 61

0.000 0.000 0.000

8 0 0

0 8 0

0 0 12

4.17) Distance 2D wannier

```
calculation dist # instantaneous liquid interfaces
system hydroxide
geo in type QE #PROFESS/VASP/QE/ABINIT/ABACUS
geo_directory OH_PBE0_vdW.pos
wannier file OH PBE0 vdW.wfc
nbands 256
geo 1
          1
geo_2
          95306
geo interval 1
          2 # number of different types of atoms.
ntype
natom 191
                  # total number of atoms.
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut_oo 3.5
rcut oh 1.24
acut hoo 30
rcut 3.0
rcut1 1.0
dr 0.1
nx 60
ny 20
#nacc 30
func 4
```

4.18) PDF O(donate)-O(accept)

```
calculation pdf
system hydroxide
geo in type QE
geo_directory OH-_PBE0_vdW.pos
         1
geo 1
geo_2 10000 95306
geo_interval 1
geo out    pdf.txt # output pdf name.
         2
                 # number of different types of atoms.
ntype
natom 191 # total number of atoms.
natom1
         64
natom2 127
dr 0.02 # delta r in real space
      6.22
rcut
id1 0
id2 H
ele1 0
ele2 O
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
nacc 4
func 2 # g(O_donate, O_accept)
```

4.19) Presolvation

```
calculation pre # hydrogen bond analysis
system hydroxide
geo_in_type QE
geo_directory OH_PBE0_vdW.pos
geo 1
          1
geo_2
         95306
geo interval 1
              # number of different types of atoms.
ntype
natom 191 # total number of atoms.
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut oo 3.5
rcut_oh 1.24
acut hoo 30
rcut 5
dr 0.01
```

4.20) Hydroxide Wannier

Note: add "nacc 5" can select those hydroxide ions with 5 accepted HBs

```
calculation wannier
system hydroxide
geo_in_type QE
geo_directory OH-_PBE0_vdW.pos
wannier_file OH_PBE0_vdW.wfc
geo 1
         1
geo_2
          95306
geo interval 1
ntype 2
natom 191
         2 # number of different types of atoms.
                  # total number of atoms.
nbands 256
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
dr 0.005
rcut 1.00
ele1 O
```

4.21) Waterwire 2

```
calculation waterwire2 # hydrogen bond analysis
 system hydroxide
 geo in type QE
 geo_directory OH_PBE0_vdW.pos
 geo 1
            1
            95306
 geo_2
 geo interval 1
                    # number of different types of atoms.
 ntype
             2
                    # total number of atoms.
            191
 natom
 natom1 64 # 0
 natom2 127 # H
 id1 0
 id2 H
 celldm1 12.444655509
 celldm2 12.444655509
 celldm3 12.444655509
 rcut oo 3.5
 rcut_oh 1.24
 acut hoo 30
 rcut 6.5
 dr 0.01
 nx 60
 ny 40
 x0 -1.5
 y0 4.5
 dx 0.05
 dy 0.05
 func 2
# factor 2.4783510188 #100ps
# factor 1.2391755094 #50ps only if output=10
# factor .6195877547 #50ps only if output=5
# factor .3717526527 #30ps
 factor .4956702036 #40ps
# func b 31
```

Waterwire 2: 41

```
calculation waterwire2
system hydroxide
geo_in_type QE
geo_directory OH_PBE0_vdW.pos
geo 1
          1
geo_2
          95306
geo interval 1
          2
                  # number of different types of atoms.
ntype
          191 # total number of atoms.
natom
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut oo 3.5
rcut_oh 1.24
acut hoo 30
rcut 6.5
dr 0.01
nx 60
ny 40
x0 -1.5
y0 4.5
dx 0.05
dy 0.05
func 2
factor .4956702036 #40ps
func_b 41
```

4.22) Waterwire compression

```
calculation ww_compress # hydrogen bond analysis
system hydroxide
geo_in_type QE
geo_directory OH_PBE0_vdW.pos
geo 1
          1
geo_2
          95306
geo interval 1
         2 # number of different types of atoms.
ntype
natom 191 # total number of atoms.
natom1 64 # 0
natom2 127 # H
id1 0
id2 H
celldm1 12.444655509
celldm2 12.444655509
celldm3 12.444655509
rcut_oo 3.5
rcut oh 1.24
acut_hoo 30
dr 0.05
rcut 8.0
```

3.23) Static structure factor

A. ssf selected

The name of the code for calculating static structure factors is "'ssf_selected.cpp" in the source directory. The code can be parallerized.

1. Prepare the input file

```
calculation
             ssf selected # command of calculating static structure
factor with selected k
geo in type PROFESS #input type of geometry file:
PROFESS/VASP/QE/ABINIT/ABACUS
geo directory ../../md files
geo 1
            0
geo 2
            999
geo interval 2
           Li ssf.txt
                            # output static structure factor name.
ssf out
           1
                       # number of different types of atoms.
ntype
                       # total number of atoms.
natom
            6750
struf dgx
            .1080405600
                          # delta G in G space, 2pi/a, a in the unit of
Angstrom
struf dgy
            .1080405600
                          # delta G in G space
                          # delta G in G space
struf dgz
            .1080405600
struf ng
                         # number of G points with delta G described above
```

- 2. Prepare the "INPUT" file as described above and run CANDELA.exe. The code will generate an output file named "SSF.input0". There number of lines in the file "SSF.input0" depends on the largest G vector, each line represents one particular G vector with four numbers, namely, Gx, Gy, Gz, and |G|.
- 3. Use the following command to sort all G vectors in an ascending order: sort -n -k4 SSF.input0 > SSF.input1

Next, delete some of the G vectors in this SSF.input1 file and only reserve those G vectors that you want to calculate. The principle here is that you can delete G vectors that aretoo close in values, thus the reduced number of G points will save the computational time for calculating the static structure factor.

- 4. Add the number of lines (the number of G vectors you want to compute) as the first line in SSF.input1 and run CANDELA again. The code will generate a new file named "SSF.input". Next, delete both SSF.input0 and SSF.input1 and only keep the "SSF.input" file.
- 5. Add the number of k points in the SSF.input file as the first line and now you can submit the final job to calculate the static structure factor.

B. ssf

The code for calculating the static structure factors (SSF) is ssf.cpp in the source directory.SSF is calculated partly with parallel script.sh and averaged through collect ssf.cpp.

- 1. Modify the script slurm.sh or files with other suffix names (like .bsub) to submit jobs.
- 2. Modify the parallel script.sh
- a. modify some parameters

```
target_dir: directory of geometry
split_to_nfile: how many parts I need
interval: geometry interval
max,min: beginning and ending geometry
addcount: restart from which part
```

b. modify INPUT in parallel script.sh

```
calculation
                         # pair Distribution Function.
             ssf
geo in type
               PROFESS
                                 # input type of geometry file:
PROFESS/VASP/QE/ABINIT/MESIA
geo directory $target dir
                            # IN PROFESS, start from file: ion.*.dat,
geo 1
            $start geo
where * is 'geo 1'
                             # IN PROFESS, end at file : ion.*.dat,
geo 2
            $stop geo
where * is 'geo 2'
geo interval $interval
                          # geometry interval between geo 1 and geo 2
ssf out
            Al ssf.txt
                          # output static structure factor name.
            1
                        # number of different types of atoms.
ntype
                        # total number of atoms.
natom
            864
struf dgx
            .2586375438 # delta G in G space, 2pi/a
             .2586375438
struf_dgy
                          # delta G in G space
             .2586375438 # delta G in G space
struf dgz
```

number of G points

Bold parameters should be modified.

18

- c. Modify slurm.sh in parallel script.sh to the right name.
- 3. Type: sh parallel script and wait till it's over.
- 4. Modify collect ssf.cpp

struf ng

```
split_to_nfile: how many parts in total
line: how many lines of each ssf_out file
```

5. Compile collect_ssf.cpp and use it to get Final_ssf.txt

3.24) Instantaneous Liquid Interface

```
calculation ili # instantaneous liquid interfaces
geo_in_type QE
geo_directory ../60.pos
geo_out
          ili.dat
geo 1
          1
geo_2
          7441
geo interval 1
          3
                  # number of different types of atoms.
ntype
          407
                  # total number of atoms.
natom
natom1 96 # 0
natom2 275 # H
natom3 36 # C
id1 0
id2 H
id3 C
celldm1 12.444655509
celldm2 12.444655509
celldm3 26.9617536455 \# celldm3 in Angstrom
zeta 2.4
d 3
nx 30
ny 30
ntry 20
z0 24
dz -0.5
ele1 0
ref_den 0.016
```

3.25) Infrared Spectra

```
Step1: use 'calculation wannier' to generate all_dipole.dat and
all vdipole.dat files
Step2: obtain the above two files and change calculation to `infrared`
 calculation infrared wannier # step 1: wannier step2: infrared
 system water
                     # system is water
 geo in type QE
 geo directory ./reorganize10/new.pos
 wannier_file ./reorganize10/new.wfc
 cell file ./reorganize10/new.cel
             1
                # starting geometry index
 geo 1
            39411 # ending geometry index
                   # interval between starting and ending geometries
 geo interval 1
 geo ignore 0
                  # ignore the first xxx geometries
           2
                  # number of different types of atoms.
 ntype
           192
                   # total number of atoms.
 natom
 natom1
           64
           128
 natom2
 id1 0
 id2 H
 ele1 0 # please do not change this value
 nbands 256 # number of bands for the system
 factor 0.05 // only for func b=2 in infrared mode
 bdf rcut 0.1 // only for func b=2 in infrared mode
 dr 0.001 # for MLWF distribution
 rcut 1.00 # for MLWF distribution
 dz 0.02 # for dipole moment distribution (in Debye)
 rcut1 10.0 # for dipole moment range (in Deybe)
 tcor 600 # correlation with 'tcor' snapshots for correlation function
 dt snapshots # the delta t of two snapshots, unit is ps
 func b 2
```

3.26) Dynamic Structure Factor

Step 1: Calculate intermediate scattering function.

```
calculation isf2 # Pair Distribution Function.
     geo_in_type PROFESS
     geo directory ../out/
     isf_outfile 2.00864isf.txt #outfile for isf
                   10000 #(make sure: interval*(nt+nT)<=geo 2-geo 1)
     geo 1
     geo 2
                   100000
     geo interval
                   1
                   864
                         # total number of atoms.
     natom
              864
     natom1
     ntype
               1
               10000 #The range of time for ISF.
     isf nt1
     isf nt2 80000 #The number of time for averaging ISF.
     dt snapshots 0.00025 #The time between two continuous snapshots.
     isf target q 2.00 \#  The target q of ISF. (in A^-1)
               20 # The number of dg in x-direction for searching
     isf ngx
     target q
     isf_ngy
                20
     isf ngz
               20
     isf dgx 0.25863754383 \# (in A^-1) The minimal step of q.
     isf dgy 0.25863754383
     isf dgz 0.25863754383
Step 2 : Calculate DSF and get maximam.
     Modify tcut, dw, nwt, interval in onedsf.sh
     tcut : The cut time for ISF. After this cut time, the ISF will decay
exponentially.
     dw : the minimal energy step for DSF (in eV)
     nwt : the maximal number of dw for DSF
     interval : dt snapshots in INPUT
     sh onedsf.sh
```

3.27) Electric Conductivity

```
Step 1: Create directories containing different snapshots. (The order of folders should be 1,2,3...)
      calculation write
      write cartesian 0
      geo in type PWmat
      geo directory MOVEMENT
      geo 11
      geo 2 9000
      geo ignore 7100
      geo interval 100
      headfile
                headqe.txt
      tailfile
               tailqe.txt
                 al.md.in
      geo out
             1
      ntype
      id1 Al
      natom
              64
      natom1 64
Step 2: Calculate electric conductivity.
      calculation
                       ele conductivity
      wf in type
                       PWmat/QE1/QE2/ABACUS
      multi directory path to multi files #the directory of multiple
   snapshots files. (only valid when nscf > 0)
      wfdirectory
                        path to wfs #(if nscf==0, then we read wf in
      path to wfs; else if nscf=Nfolders(Nfolders>0), then we read wf in
      path to multi files/[1,2,...,Nfolders]/path to wfs)
                    #maximal energy (in eV) for conductivity
      wcut
              12
              0.9 #minimla energy step (in eV) for conductivity
      dw
                    #Full Width at Half Maximam of Gauss or Lorentz
      function. Only useful when smear is 1 or 2
 (or
      n fwhm 2
      fwhm
              0.1 0.2 #use it if we want to print results with different
fwhm at the same time.)
      smear
             0
                    #0(default) for linear approx.; 1 for Gauss approx.;
   2 for Lorentz approx.
      nscf
                     #0 (default, only calculate one snapshot) or the number
   of snapshots used
      temperature 10000 # in K
      error con
                  0/1 #if print the std of multi snapshots (only valid
   when nscf > 0)
                 1 #a factor convert student distribution to normal
   distribution
```

Key words

The units of input atomic coordinates depend on the package you used for molecular dynamics simulations. For example, the output files of QE are all in Bohr, and the output files of LAMMPS are typically in Angstroms. We design specific interfaces for different simulation packages, and all data once read into CANDELA, are transformed into data with Angstroms.