OCEAN

The UW-NIST BSE Project
Department of Physics
University of Washington

User's Guide, OCEAN version 0.0
updated Feb 8, 2011

Abstract

OCEAN is an *ab initio* Density Functional Theory (DFT) + Bethe-Salpete Equation (BSE) code for calculations of core-level spectra. Currently the code allows for the calculations of x-ray absorption spectra (XAS) and non-resonant x-ray inelastic x-ray spectra (NRIXS) of periodic systems. The code is written in Fortran 90 with associated shell and Perl scripting.

This work has been supported by $$$.

Contents

Chapter 1

Package Overview

OCEAN provides a package to numerically solve the Bethe-Salpeter equation for core-level excitations. There are several steps in the process.

1. Calculate wave-functions of the ground state Hamiltonian with DFT
2. Calculate atomic orbitals for the core
3. Calculate PAW projectors and functions to transform between the pseudo and all-electron bases
4. Construct an effective Hamiltonian for the BSE, including screening the core-hole
5. Calculate the spectra

The guide is broken up into sections to follow the path the code will take.

Table 1.1: Typographic conventions in this document

font	denotes
SMALL CAPS	names of programs
`typewriter font`	contents of files
‘`quoted typewriter font`’	file names
ROMAN CAPITALS	names of cards in the ‘`input`’ file
`slanted typewriter font`	commands executed at a command line

1.1 Acknowledgments and history

Chapter 2

Tutorial

The purpose of this chapter is to work through a sample OCEAN calculation. Go to the '`EXAMPLES/rutile`' directory and look at the file '`rutile.in`'. This is the main input file. For this example let '`$OCEAN_DIR`' be the location of the installed OCEAN package. Running the command *`$OCEAN_DIR/ocean.pl rutile.in`* will run ocean. Run this now. If everything runs correctly the main output will be the file '`CNBSE/absspct`' which can be plotted against the included '`absspct`' file to check that the calculation worked.

The main input file, '`rutile.in`', contains most of the information about the run such as unit cell parameters and numerical parameters. Other important files are the '`*.fhi`' files which contain the pseudo-potentials, the '`*.fill`' and '`*.opts`' files which contain information about the PAW reconstruction, and the '`jtv1`' file which determines information about the probe photon.

Chapter 3

Input

3.1 Running OCEAN

The OCEAN package is run using the PERL script OCEAN.PL which takes care of reading in the input file and then running the various executables. This chapter will introduce the various control files and explain their formatting and use.

Eventually there will be sample input files describing various calculations and calculation strategies in Chapter **??**.

The input parser for OCEAN is relatively basic. Any text following a comment character, '#,' is ignored up to the line return. If a file has been switching around operating systems it may have non-unix line returns which can cause problems. If the parser returns odd errors try running DOS2UNIX. The parser is case ignoring and converts all inputs to lower case, requiring any filenames in the input file to be lower case. Each card must be on its own line followed by the value or values to be assigned it. All card values can be surrounded by braces ({}) and multi-value cards must be surrounded by braces. For example, the NKPT card must be a list of 3 integers and so must have the following form: "NKPT { 2 2 2 }" whereas the OCCOPT card takes only one integer and may therefore be listed: "OCCOPT 1" or "OCCOPT { 1 }". Line returns may be present within an input.

After parsing the cards are all stored as small, fortran formatted files in the '`Common`' directory. The package then goes through the following steps:

1. ABINIT input files are prepared
2. ABINIT is run to get DFT valence density and wavefunctions. This requires usually 3 runs. One self-consistent run to get the converged density and two additional runs to calculate wavefunctions for the screening calculation and for the final states.
3. The wavefunctions and density are converted from ABINIT output to the form expected by OCEAN.
4. An atomic calculation is done to create projectors following the PAW method.

5. The core electrons are screened self-consistently in the presence of the core-hole in the atomic system

6. Slater F and G integrals are computed at the atomic level

7. The spherical part of the direct interaction is screened using RPA. χ^0 is calculated using real-space integration around the site of interest. The RPA is coupled with a form of the Levine-Louie-Hybersten screening model.

8. The overlap is calculated between the core wavefunction and the conduction band using the specified photon operator, XAS or RXIS.

9. The Bethe-Salpeter equation (BSE) is iteratively solved using a form of the Haydock recursion algorithm.

3.2 Complete list of OCEAN inputs

The input of OCEAN can be divided into several sections

- structural information of the material
- non-structural DFT parameters, eg. convergence
- atomic and screening info
- parameters that control the BSE section, convergence
- parameters that only affect the output spectrum

Structural information ....

Purpose: Specify the structure

Required Cards: ACELL, RPRIM, NTYPAT, ZNUCL, NATOM, TYPAT, XRED, and DIEMAC

Non-Structural DFT parameters ....

Purpose: Control convergence of the calculation

Required Cards: PP_LIST, ECUT, NKPT, NBAND, and PAW.NBAND

Recommended Cards: NGKPT, OCCOPT, and FBAND

Optional Cards: TOLDFE, TOLWFR, PAW.NKPT, and]htmlrefSCRATCHcard:scratch

Atomic and Screening Info ....

Purpose: Controls the PAW and screening sections of the calculation

Required Cards: PAW.FILL, PAW.OPTS, NEDGES, and EDGES

Recommended Cards: PAW.SHELLS, CNBSE.RAD, and SCFAC

BSE Parameters

Purpose: Parameters used in the BSE calculation

Optional Cards: CNBSE.XMESH, METAL, CKSSTRETCH, and CNBSE.MODE

Spectrum Information

Purpose: Controls the spectrum that is written out, not any aspect of the actual calculation

Recommended Cards: CNBSE.BROADENING

Optional Cards: CNBSE.SPECT_RANGE, and CNBSE.NITER

These CARDS are listed below in (almost) the same order as in the table above. Each CARD description is of this form:

♦ **CARD** `required arguments` *type*

or

♦ **CARD** `required argument list[ length of list ]` *type*

The type is one of *Required*, *Recommended*, or *Optional*. The argument list is a brief statement of the valid arguments to the card. Arguments in square brackets are optional. The text description explains the arguments and their uses more fully. Example uses of the card look like this:

```
* brief description of the example
CARD  arguments
```

For arguments that are 2-D arrays the ordering is in normal math / C ordering.

3.3 Structural Information

♦ **ACELL** `acell[ 3 ]` *Required*

ACELL sets the scaling (in Bohr) for the primitive vectors of the unit cell which are in turn set by RPRIM.

```
* For Li metal: BCC structure, a = b = c = 6.595 Bohrs
ACELL { 6.595  6.595  6.595 }
```

♦ **RPRIM** `rprim[ 3 ][ 3 ]` *Required*

RPRIM sets the primitive vectors for defining the unit cell.

```
* For a BCC
RPRIM{ 1.0  0.0  0.0
       0.0  1.0  0.0
       0.5  0.5  0.5  }

* or the more symmetric
RPRIM{ -0.5  0.5  0.5
        0.5 -0.5  0.5
        0.5  0.5 -0.5  }
```

♦ **NTYPAT** `ntypat` *Required*

The card NTYPAT gives the number of different types of atoms in the cell.

♦ **ZNUCL** `znucl[ntypat]` *Required*

`znucl`[] is a list of length `ntypat` that gives the atomic numbers of the types of atoms that are in the cell.

♦ **NATOM** `natom` *Required*

The card NATOM gives the number of total atoms in the cell.

♦ **TYPAT** `typat[natom]` *Required*

The card TYPAT lists each atom by the rank it has in the ZNUCL list. The order they are listed in will correspond to XRED. (This may be fixed soon to be less idiosyncratic.)

♦ **XRED** `xred[natom,3]` *Required*

The XRED card lists the reduced coordinates, (x, y, z), of the all `natom` atoms in the cell.

```
* For a diamond-like structure with 2 atoms per unit cell
XRED{ 0.0  0.0  0.0    # atom 1
      0.25 0.25 0.25 } # atom 2
```

♦ **DIEMAC** `diemac` *Required*

The MACropscopic DIElectric constant `diemac` is required for both the DFT stage and also as a parameter in the screening calculations. It is included here as a physical property of the system of interest.

3.4 DFT Parameters

♦ **PP_LIST** `files[ntypat]` *Required*

The names of the pseudo-potential files need to be listed here in the same order as znucl Currently all these files need to be all lower case.

♦ **NGKPT** `ngkpt[3]` *Recommended*

DFT calculations require a self-consistently determined ground state density for the Kohn-Sham Hamiltonian. As such, in OCEAN, first a ground state calculation is done using only the occupied states and a few unoccupied states for convergence. This density is then used as input to the other DFT calculations to get wavefunctions. NGKPT determines the k-point sampling for this ground state calculation. Care should be taken to ensure enough k-points are used to give a converged density.

♦ **OCCOPT** `occopt` *Recommended*

OCCOPT controls how ABINIT determines occupation. The allowed values of `occopt` are listed in the ABINIT documentation, but the two most important are 1 and 3 as spin-dependent DFT is not yet compatible with OCEAN.

`occopt` **= 1**
: States are all doubly degenerate and either occupied or empty depending on band. Suitable for insulators.

`occopt` **= 3**
: States are all doubly degenerate but can have fractional occupations depending on fermi level. Suitable for metals.

♦ **FBAND** `fband` *Recommended*

FBAND determines the number of unoccupied bands to be included in the SCF calculation of the density. For insulators the default (`fband` = 0.125) is fine, but for metals care must be made to ensure that the highest band in the density calculation has no occupation weight. The number of extra bands is determined by the formula n = `natom` * `fband`.

♦ **ECUT** `ecut` *Required*

The plane-wave basis is truncated according to `ecut` measured in Hartree. The plane-wave cut-off is determined mostly from the pseudo-potentials being used and convergence should be checked with respect to ground state energy.

♦ **TOLDFE** `toldfe` *Optional*

TOLDFE sets the convergence parameter for the density. When the total energy changes by less than `toldfe` in two consecutive SCF runs the density is assumed to be converged. The default (`toldfe` = 10^{-6}) might be a little too permissive, but is adequate for many systems.

♦ **NKPT** `nkpt[3]` *Required*
NKPT sets the number of k-points used to sample the cell for calculation of the final states. The required number of k-points will vary based on system size (inversely with unit cell volume), and in general metals will require larger grids than insulators. Convergence should always be checked against the k-point sampling.

♦ **NBAND** `nband` *Required*
The number of total bands for the final-state wave-functions is set by NBAND. This includes all valence states but not core states.

```
* In diamond there are two C atoms each with 4 valence electrons.
* To include only the bottom 4 conduction bands
nband 8
```

♦ **TOLWFR** `tolwfr` *Optional*
The convergence criterium for the wave-function calculations is set by TOLWFR. The default (`tolwfr` $= 10^{-16}$) should be sufficient.

♦ **PAW.NKPT** `paw.nkpt[3]` *Optional*
The screening calculation is less sensitive to k-point sampling than the final states and is therefore calculated on a smaller grid which is set by PAW.NKPT. The default setting (PAW.NKPT = 2 2 2) is sufficient for a wide variety of systems though very small or very large unit cells may require more or only the gamma point respectively.

♦ **PAW.NBAND** `paw.nband` *Required*
PAW.NBAND sets the number of bands to be calculated for the screening wave-functions. The screening calculation requires a large number of bands as the screened potential only converges in the `paw.nbands` $\to \infty$ limit. In practice including bands around 100 eV above the fermi level should be sufficient, but convergence should be checked.

♦ **SCRATCH** `scratch` *Optional*
Recommended for compute clusters with local scratch space so that ABINIT will write its scratch files somewhere reasonable. The file '`scratch`' should include an absolute path and file name root, otherwise the default '`scratchXX`' will be used, writing scratch files to the '`ABINIT`' directory.

3.5 Atomic and Screening Info

♦ **PAW.FILL** `z z.fill` *Required*
Each type of atom you want spectra of requires a '`*.fill`' file which is explained in 5.0.2.

Eventually the code will handle doing an arbitrary set of edges for a given system so this input could have as many (z z.fill) pairs as you have types of atoms, ntypat. Right now you MUST only put one type here due to bugs. The file 'z.fill' can in principle have any name, but it is recommended a reasonable naming scheme is kept that associates it with a specific pseudo-potential file. The number z is the atomic number of the site of interest.

```
* Looking at Oxygen K-edge
8  o.fill
```

♦ **PAW.OPTS** z z.opts *Required*

Each type of atom you want spectra of requires a '*.opts' file which is explained in 5.0.2. Eventually the code will handle doing an arbitrary set of edges for a given system so this input could have as many (z z.opts) pairs as you have types of atoms, ntypat. Right now you MUST only put one type here due to bugs. The file 'z.opts' can in principle have any name, but it is recommended a reasonable naming scheme is kept that associates it with a specific pseudo-potential file. The number z is the atomic number of the site of interest.

```
* Looking at Oxygen K-edge
8  o.opts
```

♦ **NEDGES** nedges *Required*

The number of edges to run is set by NEDGES. Each site and edge is counted separately. So for a 64 atom amorphous carbon cell you would want nedges = 64 for a complete sample. To run the K and L edges Titanium edges in TiS_2 you would need nedges = 3. See also EDGES.

♦ **EDGES** edges[nedges,3] *Required*

Each edges being run is defined by 3 integers that set which atom in the same order as XRED, and which initial state by n and l. Note, spin-orbit split edges are calculated together as is necessary to include multiplet terms.

For example, running TiS_2 where Ti was listed as the first atom and looking at the K, L_1, and $L_{2,3}$ edges respectively.

```
* atom, n, l
  Ti    1  0
  Ti    2  0
  Ti    2  1
```

♦ **PAW.SHELLS** shells[] *Recommended*

The screening calculation is a hybrid of RPA and a model based off of Levine-Louie-Hybersten see section 6. The corss-over between these two regimes is set by shells in Bohr and several different radii can be chosen to look at the convergence. In the **`shells`** $\rightarrow \infty$ limit the RPA is being used for the entire calculation, but convergence is usually reached around 3 - 4 Bohr. If a large radius is being used than the supercell defined by PAW.NKPT must have dimensions larger than the radius chosen or the math could be odd. The default **`shells`** $= 1.5$ Bohr is not recommended.

♦ **CNBSE.RAD** `cnbse.rad` *Recommended*

One screening radius as defined by PAW.SHELLS is used in the BSE calculation and is specified by CNBSE.RAD.

♦ **SCFAC** `scfac` *Recommended*

The Slater integrals are calculated in an atomic program and are generally scaled by some factor before use. For 3d transition metals a value of 0.8 is often used, while for f-electron atoms a value of 0.6 may be more appropriate. The value is more or less taken to be independent of the chemical environment and only a function of the element in question.

3.6 BSE Parameters

♦ **CNBSE.XMESH** `xmesh[3]` *Optional*

When the wave-functions are converted into the NIST BSE format they are condensed onto a grid controlled by CNBSE.XMESH. The states are then projected onto a localized basis set, using the PAW formalism. The default(**`xmesh`** = 6 6 6) might be insufficient for large unit cells.

♦ **METAL** `metal` *Optional*

The METAL card determines wether the code expects to determine occupation by band or by energy with respect to the fermi energy. The parameter **`metal`** can be set to either *.true.* or *.false.* (default). If **`metal`** is set to *.true.* the code will complain unless OCCOPT is 3 and similarly for **`metal`** = *.false.* and **`occopt`** = 1.

♦ **CKSSTRETCH** `stretch` *Optional*

An approximation to some kind of self-energy is allowed through the use of CKSSTRETCH. All conduction band energies with respect to the fermi level are multiplied by **`stretch`**. By default **`stretch`** = 1.0.

♦ **CNBSE.MODE** `interactions iterations` *Optional*

`interactions`

> The interaction terms of the BSE Hamiltonian can be scaled (or increased) by an arbitrary factor. By default **`interactions`** = 1.

`iterations`

The number of iterations in the Haydock recursion (default is 100). This parameter should scale with the size of the BSE Hamiltonian which is controlled by the number of states, ie. bands and k-points.

3.7 Spectrum Information

♦ **CNBSE.SPECT_RANGE** `emin emax` *Optional*

The energy of the resulting spectra is set by setting the conduction band minimum plus the core-hole binding energy in the absence of spin-orbit splitting to 0. All systems will include a small amount of excitonic binding (depending on l selection rules) and spin-orbit splitting can lower the first peak significantly, eg. $L_{2,3}$ splitting in 4d transition metals. This does not change the BSE calculation, only the output of the spectra at the end. The energy range is given in Hartree and by default `emin` $= -1.5$ and `emax` $= 2.5$.

♦ **CNBSE.NITER** `niter` *Optional*

The card is a mis-nomer, don't use and I'll probably change it.

CNBSE.NITER sets the number of energy points to use when plotting the final spectra. The default is 1200 and should be good for most uses.

♦ **CNBSE.BROADEN** `broadening` *Recommended*

The amount of Lorentzian broadening included in the spectra is set by CNBSE.BROADEN. This is a required convergence parameter for the continued fraction that is our Haydock approximation to the BSE, but setting it to the core-hole intrinsic broadening is recommended since features sharper than that won't be observable. Given in Hartree the default is `broadening` $=$ 0.007 Ha. or about 0.1 eV.

Chapter 4

Ground-State Wavefunctions

In principle any pseudo-potential DFT code can be used. In our implementation we have an interface for using ABINIT[?]. To run with a custom format see the specs listed in Appendix ??.

There are two important differences between what is required for a good core-level calculation with OCEAN and more traditional ground-state DFT calculations. The first is that the wave-functions are projected into a real-space basis around an atomic site, which can lead to faster convergence with respect to k-point sampling. Also, a psuedo-potential that is suitable for ground-state calculations may have poor scattering characteristics at higher energies that will make up the spectrum being calculated. With these brief warnings the rest of this chapter will be spent going over various aspects of the calculation and mention briefly strategies for ensuring accuracy and convergence.

The user is assumed to be familiar with DFT calculations, ie. converging the total energy or unit cell volume.

4.1 Pseudo-potentials

Pseudo-potentials must be carefully constructed to ensure that they adequately mimic the all-electron system being investigated. For general purpose pseudo-potentials this means having good transferability so they can be used in a variety of chemical environments. A large body of literature exists on the topic and the reader is invited to read up on it.[?]

As mentioned above for excited state calculations states as high as 50-100 eV above the fermi level must be adequately represented by the pseudo-potentials used. This often requires a harder pseudo-potential than might otherwise be necessary for other calculations. Additionally the way in which chemical bonding affects the valence and low-lying conduction bands is often influenced by the inclusion of semi-core states in the calculation. Better agreement is often seen with the inclusion of semi-core states, eg. 2s and 2p states in a 3d transition metal. This coincidently has been seen to improve the high-energy scattering.

4.2 K-point sampling

A sum over a few wavefunctions ψ_k is substituted for a integral over the entire Brillouin zone. In principle knowledge of the crystal symmetry of the unit cell could allow for intelligent choices of points to sample at with non-equal weight, but that is not the approach here. Instead a regular mesh of points in reciprocal space is used and the mesh must be dense enough to ensure that the sampling is sufficient.

In practice the required k-point grid scales inversely with unit cell size, but testing runs at different sizes and checking that the spectrum is unchanged should be done for each calculation.

4.3 Beyond DFT

In principle these wave-functions should include self-consistent self-energy corrections, eg. GW, but in practice it has been found that for many systems self-energy corrections only affect the energies and not the wave-functions. Even single-shot self-energy calculations can be computationally expensive and we provide several methods to approximate the GW correction; ad-hoc band stretching and the many-pole self-energy which are both presented later.

LDA+U should work? The author hasn't thought of any reasons why it wouldn't, but ABINIT doesn't currently write out usable wavefunctions when using LDA+U .

Chapter 5

PAW and Atomic

The PAW/Atomic section of the code is material independent. In principle this section can be run once for a given pseudopotential and reused. The important outputs of this section are a collection of all-electron and pseudo projector augmented waves. The all-electron core wavefunction is expressed as coefficients of the all-electron PAW. The psuedo PAW will be used later to express the valence/conduction electron wavefunctions.

The transition matrix elements between core and pseudized conduction or valence states are calculated using the projector augmented wave (PAW) formalism developed by Blöchl [**?**]. In OCEAN the construction of all-electron and pseudized atomic states is taken care of by the program *hfk.x*. The inputs and outputs of this section are unique for each pseudopotential, but not for each system, ie. a library of pseudopotentials and atomic info files can be collected and reused.

5.0.1 scfac

The SCaling FACtor is a real number that modifies the calculated Slater-Condon parameters for the atomic case. For 2nd row elements this should be 1.d0 while for transition metals a factor of 0.8d0 is more appropriate. See DeGroot, some others probably [**?**].

5.0.2 fill

The *fill* file determines the numerical parameters used for the PAW construction. Including the energy range to construct waves over, the maximum number of waves per angular momenta, the cut-off radius to use, and the momentum grid for evaluating the Fourier transform of the projectors.

2	pow
-1.30 2.00 0.0001 0.01 6	Emin, Emax, prec1, prec2, tfac
3.5 1e6	PAW radius, prec
0.05 20	q step, q max

The matrix elements are calculated up to $\langle\phi_i|n^{pow}|\phi_j\rangle$. The PAW reconstruction starts at energy Emin and attempts to span the space to Emax (both in Ha.). The PAW radius is in Bohr and convergence should be checked. The overlaps between the states and the projectors is done in reciprocal space on a grid which is controlled by q step and q max, but there parameters should be sufficient for a range of systems.

5.0.3 opts

The *opts* file contains information about the pseudopotential such as z, core-valence partitioning, and reference configuration. This information should match that used to construct the pseudopotential. In the case of pseudopotentials whose origin is unknown a reasonable guess must be made.

008	Z
1 0 0 0	core states; s p d f
scalar rel	
lda	functional
2.0 3.5 0.0 0.0	valence occupation; s p d f
2.0 3.5 0.0 0.0	repeat of above

For the opt file one specifies the number of levels not included in the pseudo-potential, ie 1s for Oxygen as well as the number of electrons included in the reference configuration of the pseudo-potential, ie 2.0 2s and 3.5 2p for oxygen. For functionals the choice is lda or hf. Options for treatments of spin-orbit are scalar or dirac and rel[ativistic] nonrel[ativistic] and should of course match the psuedopotential construction.

5.1 Diagnostics from the PAW and atomic program

There are a number of output files created by the PAW and atomic program, both inputs for later parts of the code as well as diagnostic files. The diagnostic files are placed in the subdirectory *zdiag_Z_N_L*.

5.1.1 Matrix elements

The *mt##* files contain a comparison of the overlap matrix elements using the all-electron projector augmented waves versus the pseudized partial waves. The first number in the file name is the power of r and the second is the angular momentum quantum number l used. Plotting columns 3 versus 4 should give a straight line for perfect agreement.

5.2 Outputs from the PAW and atomic program

5.2.1 Fourier Coefficients

The *aeft#* and *psft#* files contain the Fourier transform of the projector augmented waves. They are kept in the *zdiag* directory.

5.2.2 Projectors

The *pr#* files contain the projectors on a real-space grid by l. The projectors are, by definition, 0 outside of the PAW cut-off radius. They are kept in the *zdiag* directory.

5.2.3 Info files

The *prjfilez###* file, where # is the three digit Z value, contains info about the Fourier grid and angular momentum range of the projector augmented waves. The first line specifies the minimum angular momentum, maximum angular momentum, number of fourier grid points, and the spacing of the fourier grid. Each following line specifies the number of PAWs for each angular momentum channel.

the *radfilez###* files, where # is the three digit Z value, contains info about the real-space grid. The first line is cut-off radius, rounded from the input cut-off to the nearest grid point. The second is the number of real-space grid points, and the last is the index of the cut-off.

5.2.4 Matrix Elements

The *melfilez###n##l##* file contains the matrix elements between the specified core-hole and the different PAW functions. The first line specifies the maximum x for the operator r^x, always starting at 0. Starting with the fast index the file is then the value of the matrix elements looped over projectors, powers of x, and finally angular momentum.

Chapter 6

Screening Calculation

The screening of the core-hole interaction is done in real space using the random phase approximation up to some radius around the core, 2-5 bohrs, and then using an analytic function of the local density, distance, and static dielectric constant outside of this radius [**?**]. Convergence with respect to this radius should be checked and will change based on both the core and system of interest. The screening is only exact for an infinite number of bands, but should smoothly converge w.r.t. number of bands included. For the average system a 2^3 k-point grid is sufficient for convergence, while a very small system might require a slightly higher sampling and a large unit cell might be ok with just the gamma point.

Chapter 7

CNBSE

The CNBSE section solves for the spectra. The important output file is absspct; column 1 is the energy, column 2 the spectra using a plamon-pole broadening model and column 3 the plain spectra.

Appendix A

Installation

- The code has been tested with both pgi and intel fortran compilers. Would love to hear people’s problems/successes using other vendors. Especially want to add some free (as in $) compiler to the works list.
- Requires fftw3 currently. If you are running on a managed cluster that has nice MODULE support (like NERSC) simply loading the fftw3 module may work fine. Otherwise set the environmental variables FFTWL and FFTWI. For BASH ‘export FFTWL=“-L/myfftw3/lib -lfftw3”’ and ‘export FFTWI=“-I/myfftw3/include”’ and for CSH ‘setenv FFTWL “-L/myfftw3/lib -lfftw3”’ and ‘setenv FFTWI “-I/myfftw3/include”’.
- The code requires perl

The bare bones installation instructions are as follows.

1. Run ‘make FC=ifort OPTIONS=-O2’
2. If this doesn’t fail, type ‘make install INSTDIR=“ ˜/OCEAN_BIN/”’ or wherever you want the installation to go.
3. Create links to abinit and cut3d within the OCEAN_BIN diretory, ie. ‘ln -s /bin/abinit abinit’

Assuming success so far you can now go to the EXAMPLE/rutile directory. Here you will find what you need to do a preliminary calculation of the Ti $L_{2,3}$ edge in rutile. To run OCEAN run the command ‘˜/OCEAN_BIN/ocean.pl rutile.in’ and wait for the code to finish. The code will create several directories and the end results are in the file CNBSE/absspct which can be compared to the absspct file that is distributed in the rutile directory.