Env2Seward Technical Manual

Joshua Horswill

June 24, 2020

Contents

1 Author and Permissions

This code was written by **Joshua Horswill** during an internship with the ILL theory group in June 2020 under supervision of **Marie-Bernadette Lepetit**.

It is permissible to use and diffuse this code provided J. Horswill and MB lepetit are kept informed. Modifications to this code can be made provided that the developer sends modifications to J. Horswill or M.B. Lepetit, so they may be included in the source code.

MB Lepetit
Institut Néel, CNRS UPR 2940
25 rue des Martyrs, BP 166, Bât. K
38042 Grenoble cedex 9
FRANCE

Courriel : Marie-Bernadette.Lepetit@Neel.CNRS.fr

J. Horswill
Institut Laue-Langevin
71 Avenue des Martyrs,
38000 Grenoble,
France

Email : horswill@imap.ill.fr

2 Manual

2.1 Introduction

As you may have guessed from the name of the script, this code processes and formats the output from a program called ENV so it can be used in a calculation performed by a program called SEWARD. These simulation scripts are found in the local tools manual for the 'Fast calculation of the electrostatic potential in ionic crystals by direct summation method' written by MB Lepetit and A. Gellé. This local tools manual contains scripts that provide a "set of charges that allow the calculation of the Madelung potential of an ionic crystal within a predefined accuracy" [1]. This is designed for the purpose of performing an ab-initio quantum calculation on a 'quantum fragment' that is surrounded by a first shell of ionic pseudo-potentials, and a second layer of renormalised point charges [2]. The 'prefix' is a placeholder for the name of the compound being simulated.

2.1.1 How to compile and run the script

A **python 3** distribution such as 'Anaconda' or 'Active Python' must be downloaded, which provides the option of using integrated development environments (IDE) for programming in the python language. For the development of this script, 'Spyder' was used. This is an IDE available in Anaconda. Before running the file, make sure the script is in the same directory as the input files described below. After you have made sure this is the case, open the file with an IDE (typically an 'open file' option in the top left of the window, or you can drag it into the IDE text editor), hit run, and answer the prompts in the terminal. If you want to install python directly, see `https://realpython.com/installing-python/`. Typing '`python3 env2seward.py`' in the terminal running in the same directory as the python file should execute the script, generating the prompts in the command line. The new file will appear in the same directory as the python script.

2.2 Prompt entry

2.2.1 Inputs and outputs

Prefix.env.sew0 is a help file that contains the input data for SEWARD of the MOLCAS chain. It contains the data for the quantum fragment, the TIPS and the renormalised charges. This is one of two input files for the env2seward script. The second is called prefix.psd, and contains the information on the atoms represented by TIPs. The ordering seen in this file is as follows: first, atoms that are irreducable (with symmetry), and second, all atoms (without symmetry). It also requires two groups of basis sets - one for the quantum fragment and one for the ionic pseudopotential. Once these have been prompted and entered through the terminal command line, an output file ready to be fed into SEWARD will be generated. This can be called prefix.sew.in.

In summary the files and data required for this program are:

- `prefix.env.sew0`
- `prefix.env.psd`
- Fragment basis sets and library locations.
- Pseudopotential basis sets and library locations.
- Name of the prefix.

The input files must be in the same directory as the env2seward script for the program to run successfully. The first prompt before basis set and library input will be to ask if you are running the script in a Jupyter Notebook (see relevant section below). If you wish to be prompted for input, enter 'N' or 'n'.

2.2.2 Prefix, Title and Basis Set Formatting

The prefix entered can be anything you want, but it must match the prefix on the input files. For example, if the prefix was 'example', this automatically generates a seward input file called '`example.sew.in`'. However, this will not work unless the env input files have the filenames '`example.env.sew0`' and '`example.env.psd`'. The input on the prompt for the title will be written to the document introduction, and this does not depend on the name of the input files or the prefix.

Basis sets for the fragment potential are unique to the system being evaluated and so cannot be managed by env2seward. These must be entered individually when prompted by the command line, as well as their library location. For example, if oxygen is one of the elements contained within the input files, the command will prompt for it's basis set input after parsing the prefix.env.sew0 file. It will then ask for the specified library location. If the corresponding

library is default for the system, leave the input blank and press enter. If this is not the case, submit the name of the specified library.
It is almost the same for the pseudopotential, but the pseudo basis sets depend upon the charge of the atom. For example, for GdMn2O5 there are three elements but eight basis sets, since there are two types of gadolinium and manganese and four types of oxygen. This results in more prompts for the corresponding basis sets and libraries. The command line will distinguish between the inputs between the fragment and the TIPs. Continue until all basis sets have been entered and the file will be written.

2.3 Reading Filenames, Basis Sets and Libraries from an Input File

There is another option for entering the data required to format the ENV output files, and this is to write the filenames, title and basis set information in a text file. To do this, write the text file in the following format (each bullet point is a new line, although the order does not matter):

- `filename = chosen_filename`
- `title = chosen_title`
- `sew0_file = prefix.env.sew0`
- `psd_file = prefix.env.psd`
- `lib_frag = {"atom_type":{"loc":"specified basis set library","key":"basis set for atom_type"},"atom_type2":...}`
- `lib_pseudo = {"atom_type":{"loc":"specified basis set library","key":"basis set for atom_type"},"atom_type2":...}`

The format of the basis set libraries is in the python dictionary syntax. This means each dictionary key (e.g `"atom_type,"loc"` and `"key"`) corresponds to a piece of data. In this case the data is either a string or another dictionary. Each atom is assigned to it's own dictionary containing a key for the specified basis set library (set to `"loc":""` if default) and a key for the basis set (`"key":"basis set"`). Here is an example for a GdMn2O5 input file, where the fragment dictionary (`lib_frag`) has no specified library but the TIP dictionary (`lib_pseudo`) basis sets are found in PSEUDO:

```
filename = example.sew.in
title = example
sew0_file = GdMn2O5_J1.env.sew0
psd_file = GdMn2O5_J1.env.psd
lib_frag =
    {'Mn':{'loc':'','key':'Mn.ano-rcc.Roos.21s15p10d6f4g2h.6s4p3d1f0g.'},
'O':{'loc':'','key':'O.ano-rcc.Roos.14s9p4d3f2g.4s3p1d0f'}}
```

```
lib_pseudo =
    {'Gd1':{'loc':'PSEUDO','key':'Gd.ECP.Marie.0s.0s.0e-Gd1-GdMn2O5.'},
'Gd2':{'loc':'PSEUDO','key':'Gd.ECP.Marie.0s.0s.0e-Gd2-GdMn2O5.'},
'Mn1':{'loc':'PSEUDO','key':'Mn.ECP.Marie.0s.0s.0e-Mn1-GdMn2O5.'},
'Mn2':{'loc':'PSEUDO','key':'Mn.ECP.Marie.0s.0s.0e-Mn2-GdMn2O5.'},
'O1':{'loc':'PSEUDO','key':'O.ECP.Marie.0s.0s.0e-O1-GdMn2O5.'},
'O2':{'loc':'PSEUDO','key':'O.ECP.Marie.0s.0s.0e-O2-GdMn2O5.'},
'O3':{'loc':'PSEUDO','key':'O.ECP.Marie.0s.0s.0e-O3-GdMn2O5.'},
'O4':{'loc':'PSEUDO','key':'O.ECP.Marie.0s.0s.0e-O4-GdMn2O5.'}}
```

Make sure that all variables are on the **same line** as the string/dictionary they are being assigned to. There also must be spacing between the variable, the = sign and the data, so that the parsing algorithm can identify the data and write it to the output file.

Make sure this file is in the same directory as the python script and the ENV input files. Once you have written this text file with the correct formatting and wish for it to be parsed by env2seward.py, enter the following command:

- `python3 env2seward.py chosen_input_filename`

If the setup has been performed correctly, the script will output 'File has been created' and the sew.in file will appear in the script directory.

2.4 Jupyter Notebook Version

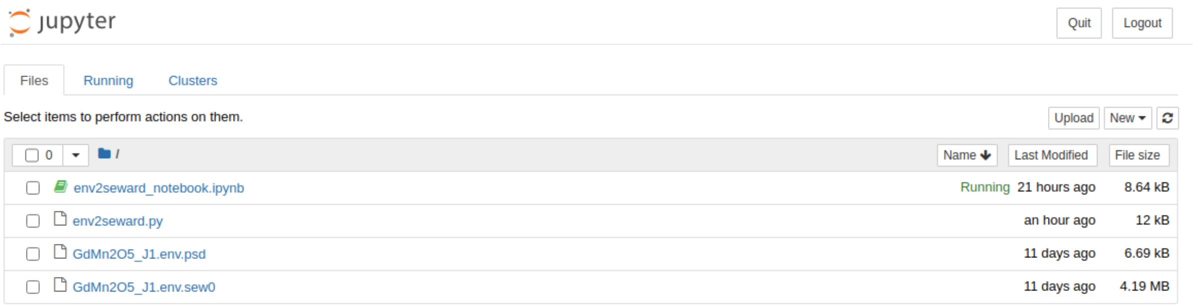
To install Jupyter Notebook, follow this guide: `https://jupyter.org/install`. For the creation of the `env2seward_notebook.ipynb` file, the Anaconda distribution was used to install Jupyter Notebook with the `conda` command. This is the recommended method (see `https://www.anaconda.com/products/individual` for distribution download).

Once you have installed Jupyter notebook you need to download and localise the following files into the same directory/folder:

- `env2seward.py`
- `env2seward_notebook.ipynb`
- `prefix.env.sew0`
- `prefix.env.psd`

Next we need to open the Jupyter Notebook web app. We can do this by:

- Opening up a terminal in the directory containing the four files mentioned above.
- Entering '`jupyter notebook`' to initialise the web app.
- Your browser should open it in a new tab and you should see these four files in your notebook directory.



Select items to perform actions on them.

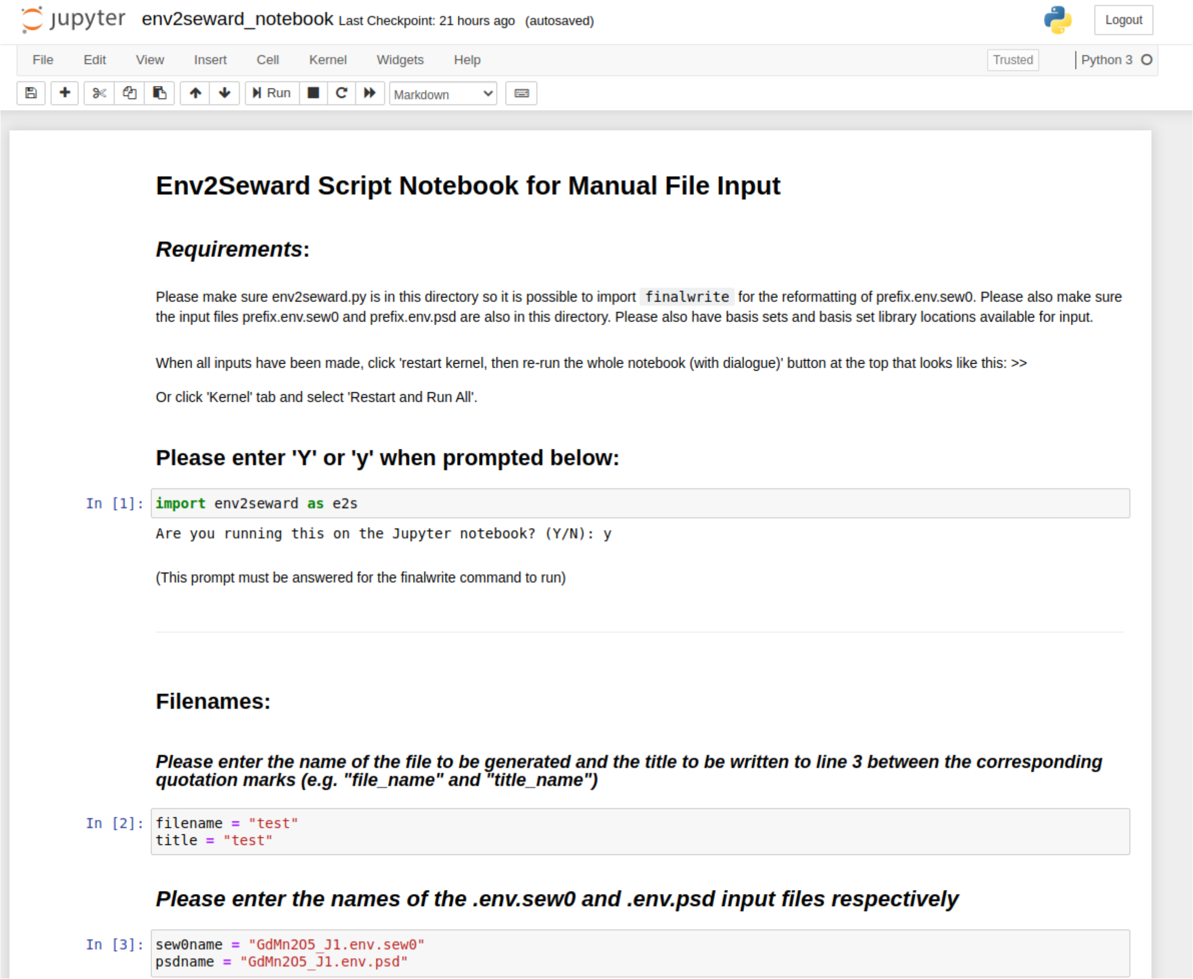
Name ↓		Last Modified	File size
env2seward_notebook.ipynb	Running	21 hours ago	8.64 kB
env2seward.py		an hour ago	12 kB
GdMn2O5_J1.env.psd		11 days ago	6.69 kB
GdMn2O5_J1.env.sew0		11 days ago	4.19 MB

Figure 1: Web app dashboard menu

- Now open the `env2seward_notebook.ipynb` file from the web app dashboard menu.

You should see the menu shown in figure (1) and the notebook shown in figure (2). In the notebook you will see a page of Jupyter cells, some in markdown (a text formatting language) and some to run python3 code. If you want to learn more about how these work, see `https://www.dataquest.io/blog/jupyter-notebook-tutorial/`. In order to run these cells individually, click on them and hit `Ctrl+Enter` or press the run button (below the cell tab in the toolbar). Please follow the instructions in between the code cells to enter the correct inputs. If you would like to see the atom-types that require basis set entry, follow the instructions under the subtitle 'Basis atoms (see instructions below on how to run this cell):'. The next steps are:

- Make sure the inputs have all been entered according to the instructions (filename, title, sew0name, psdname, lib_frag, lib_pseudo)
- Select the button that says 'restart the kernel and re-run the whole notebook' that looks like a fast-forward icon.
- If you cannot find this, click the kernel icon (in English, it may be different in other languages), next to cell and widgets, and select 'Restart and Run All'.
- You will see a prompt in the first code cell (`import env2seward as e2s`). Please enter 'Y'.
- After this the last code cell (`e2s.finalwrite(filename,...)` will output 'File has been created' which means you now have a seward input file ready in the webapp directory. You can download this to a specific directory by going back to the dashboard menu, ticking the box next to '`chosen filename`' and clicking 'Download'.
- The output file will also naturally appear in the same directory as your `env2seward_notebook.ipynb` file.



Env2Seward Script Notebook for Manual File Input

***Requirements*:**

Please make sure env2seward.py is in this directory so it is possible to import `finalwrite` for the reformatting of prefix.env.sew0. Please also make sure the input files prefix.env.sew0 and prefix.env.psd are also in this directory. Please also have basis sets and basis set library locations available for input.

When all inputs have been made, click 'restart kernel, then re-run the whole notebook (with dialogue)' button at the top that looks like this: >>

Or click 'Kernel' tab and select 'Restart and Run All'.

Please enter 'Y' or 'y' when prompted below:

```
In [1]: import env2seward as e2s

        Are you running this on the Jupyter notebook? (Y/N): y
```

(This prompt must be answered for the finalwrite command to run)

Filenames:

Please enter the name of the file to be generated and the title to be written to line 3 between the corresponding quotation marks (e.g. "file_name" and "title_name")

```
In [2]: filename = "test"
        title = "test"
```

Please enter the names of the .env.sew0 and .env.psd input files respectively

```
In [3]: sew0name = "GdMn2O5_J1.env.sew0"
        psdname = "GdMn2O5_J1.env.psd"
```

Figure 2: env2seward_notebook when opened

References

[1] Alain Gellé and Marie-Bernadette Lepetit. Fast calculation of the electrostatic potential in ionic crystals by direct summation method. *The Journal of chemical physics*, 128(24):244716, 2008.

[2] J Varignon, S Petit, A Gellé, and MB Lepetit. An ab initio study of magneto-electric coupling of ymno3. *Journal of Physics: Condensed Matter*, 25(49):496004, 2013.