CASTEP output analysis

Steven R. Schofield (Universtiy College London) May 2025

Load required modules

In [ ]:
```
import sys
from pathlib import Path

# Define candidate paths using Path objects
module_path_list = [
    Path('/Users/steven/academic-iCloud/Python/modules'),
    Path('/hpc/srs/Python/modules')
]

data_path_list = [
    Path('/Users/steven/academic-iCloud/Calculations/castep/Hydrogen_Bridge/
    Path('/hpc/srs/Python-data')
]

# Resolve actual paths
module_path = next((p for p in module_path_list if p.exists()), None)
data_path = next((p for p in data_path_list if p.exists()), None)

# Check and report missing paths
if module_path is None:
    print("Error: Could not locate a valid module path.")
if data_path is None:
    print("Error: Could not locate a valid data path.")

if module_path is None or data_path is None:
    sys.exit(1)

# Add module_path to sys.path if needed
if str(module_path) not in sys.path:
    sys.path.insert(0, str(module_path))

# Print resolved paths
print(f"module_path = {module_path}")
print(f"data_path = {data_path}")
```

In [ ]:
```
# # Ensure modules are reloaded
%load_ext autoreload
%autoreload 2

# Import standard modules
import numpy as np
```

```
import pandas as pd

# Import custom module
import SRSCALCUTILS.castep_tools as ct

from IPython.display import display, Image as StaticImage
```

Programme variable setup

In [ ]:
```
# Parameters for windows creation
# General
job_folder = '1x6x8/1x6x8-821MP-SP-negative'
job_path = data_path / job_folder
verbose = False             # Set this True to print out more information

# Print resolved paths
print(f"job_path = {job_path}")
```

In [ ]:
```
castep_paths = ct.find_all_files_by_extension(job_path,'castep')
#ct.optimisation_summaries(castep_paths)
ct.optimisation_summaries(castep_paths)
```

In [ ]:
```
data_summary = ct.collect_summary_table(job_path)
with pd.option_context('display.max_rows', None, 'display.max_columns', None
    display(data_summary)
```

In [ ]:
```
params = {'task' : 'geomopt',
          'xc_functional' : 'PBE',
          'cut_off_energy' : 750,
          'spin_polarised' : 'true',
          'write_cell_structure' : 'true',
          'charge' : 0}
```

In [ ]:
```
def write_param_file(params):
    width = max(len(k) for k in params)
    for param, value in params.items():
        print(f'{param:<{width+1}} : {value}')
```

In [ ]:
```
!jupyter nbconvert castep_analysis.ipynb --to webpdf
```

In [ ]:
```

```