Fukui Module

PS-TEROS Documentation

Calculating Fukui Functions via Finite Difference Interpolation

Contents

1 Introduction

1.1 What are Fukui Functions?

Fukui functions are conceptual DFT descriptors that measure the local reactivity of a molecular or surface system. They quantify how the electron density at a given point changes when electrons are added or removed from the system.

The Fukui functions are defined as:

$$f^+(\vec{r}) = \frac{\partial \rho(\vec{r})}{\partial N} \mid_{v(\vec{r})} \approx \rho_{N+\delta N}(\vec{r}) - \rho_{N(\vec{r})}$$

$$f^-(\vec{r}) = \frac{\partial \rho(\vec{r})}{\partial N} \mid_{v(\vec{r})} \approx \rho_{N(\vec{r})} - \rho_{N-\delta N}(\vec{r})$$

Where:

- $f^+(\vec{r})$ is the **nucleophilic Fukui function** — indicates where electrons will be added (electrophilic attack sites)
- $f^-(\vec{r})$ is the **electrophilic Fukui function** — indicates where electrons will be removed (nucleophilic attack sites)
- $\rho(\vec{r})$ is the electron density
- N is the number of electrons
- $v(\vec{r})$ is the external potential (kept constant)

1.2 Why Use Fukui Functions?

Fukui functions are powerful tools for:

- **Predicting reactive sites** on surfaces and molecules
- **Understanding adsorption** — where adsorbates will preferentially bind
- **Catalyst design** — identifying active sites for reactions
- **Surface chemistry** — predicting oxidation/reduction behavior

1.3 The Interpolation Method

Instead of using integer electron differences (which can cause convergence issues), we use **fractional charges** with interpolation:

1. Calculate charge densities at multiple fractional charge states:
   - $\delta N = 0.00$ (neutral)
   - $\delta N = 0.05$
   - $\delta N = 0.10$
   - $\delta N = 0.15$
2. Fit a polynomial to the charge density vs. δN data
3. Extract the derivative at $\delta N = 0$ to obtain the Fukui function

This approach provides:

- Better numerical stability
- Smoother Fukui functions
- More accurate results for extended systems (surfaces, slabs)

2 Module Architecture

2.1 Overview

The Fukui module in PS-TEROS automates the calculation of Fukui functions using AiiDA-WorkGraph for workflow management and VASP for DFT calculations.

```
teros/core/fukui/
├── __init__.py      # Public API exports
├── workgraph.py     # Main WorkGraph builder
├── tasks.py         # AiiDA calcfunctions
├── utils.py         # Utility functions
└── docs/            # Documentation
```

Listing 1: Module structure

2.2 Components

2.2.1 `utils.py` — Utility Functions

Function	Description
`make_delta_label(delta_n)`	Converts float (0.05) to valid label ("delta_0_05")
`validate_fukui_inputs(...)`	Validates input parameters before workflow execution
`DEFAULT_DELTA_N_VALUES`	Default: [0.0, 0.05, 0.10, 0.15]

2.2.2 `tasks.py` — AiiDA Calcfunctions

Function	Description
`extract_total_energy(misc)`	Extracts energy from VASP output
`collect_chgcar_files(...)`	Collects CHGCAR files from multiple calculations into single FolderData
`generate_fukui_summary(...)`	Creates summary Dict with calculation metadata

2.2.3 `workgraph.py` — WorkGraph Builder

Function	Description
`FukuiCalculationScatter`	`@task.graph` for parallel VASP calculations
`build_fukui_workgraph(...)`	Main entry point — builds complete workflow
`get_fukui_results(wg)`	Extracts results from completed WorkGraph
`print_fukui_summary(wg)`	Prints formatted summary of results

3 Usage Guide

3.1 Basic Usage

```
from aiida import orm, load_profile
from teros.core.fukui import build_fukui_workgraph, get_fukui_results

load_profile('your_profile')

# Load or create structure
structure = orm.load_node(12345)  # or from file

# Build the workflow
wg = build_fukui_workgraph(
    structure=structure,
    nelect_neutral=192,           # REQUIRED: verify from VASP OUTCAR
    delta_n_values=[0.0, 0.05, 0.10, 0.15],
    code_label='VASP-6.5.1@cluster',
    builder_inputs={
        'parameters': {
            'incar': {
                'encut': 500,
                'ediff': 1e-6,
                'algo': 'All',
                'ispin': 2,
                # ... other settings
            }
        },
        'options': {...},
        'kpoints_spacing': 0.25,
        'potential_family': 'PBE.54',
        'potential_mapping': {'Sn': 'Sn_d', 'O': 'O'},
    },
    fukui_type='plus',            # 'plus' or 'minus'
    max_concurrent_jobs=4,
)

# Submit
wg.submit(wait=False)
print(f"Submitted: PK={wg.pk}")
```

3.2 Parameters

Parameter	Required	Description
`structure`	Yes	Input structure (StructureData or PK)
`nelect_neutral`	Yes	Number of electrons in neutral system
`delta_n_values`	No	List of δN values (default: [0.0, 0.05, 0.10, 0.15])
`code_label`	Yes	VASP code label
`builder_inputs`	Yes	VASP calculation parameters
`relax_first`	No	Relax structure before charge calculations (default: False)

`relax_builder_inputs`	No	Separate inputs for relaxation step
`fukui_type`	No	`'plus'` (nucleophilic) or `'minus'` (electrophilic)
`max_concurrent_jobs`	No	Limit parallel calculations

3.3 Determining NELECT

The `nelect_neutral` parameter is **critical** and must be determined from your pseudopotentials:

```
# Run a test VASP calculation and check OUTCAR:
grep "NELECT" OUTCAR
```

For common pseudopotentials:

- Standard PBE: Sum of valence electrons per element
- Example: SnO_2 with 12 Sn + 24 O
  - Sn (standard): 4 valence electrons
  - Sn_d: 14 valence electrons
  - O: 6 valence electrons
  - NELECT = 12×4 + 24×6 = 192 (standard) or 12×14 + 24×6 = 312 (Sn_d)

3.4 VASP Settings for Fukui Calculations

The module automatically enforces these settings for charge-state calculations:

```
# Automatically set by the module:
'nsw': 0,        # Static calculation
'ibrion': -1,    # No ionic relaxation
'lcharg': True,  # Write CHGCAR
'lwave': False,  # Don't write WAVECAR
'nelect': <calculated>,  # Fractional electron count
```

Recommended user settings for robust SCF convergence:

```
'incar': {
    'prec': 'Accurate',
    'encut': 500,           # Well-converged cutoff
    'algo': 'All',          # Most robust algorithm
    'nelm': 300,            # Allow many SCF steps
    'ediff': 1e-6,          # Tight convergence
    'icharg': 2,            # Start from atomic charges
    'amix': 0.2,            # Conservative mixing
    'bmix': 0.0001,         # Kerker mixing
    'lmaxmix': 4,           # For d-electrons
    'ismear': 0,            # Gaussian smearing
    'sigma': 0.05,
    'ispin': 2,             # Spin-polarized
}
```

3.5 Extracting Results

After the workflow completes:

```
from teros.core.fukui import get_fukui_results, print_fukui_summary

# Get results
results = get_fukui_results(wg.pk)

# Access CHGCAR files
chgcar_folder = results['chgcar_folder']
print(chgcar_folder.list_object_names())
# ['CHGCAR_0.00', 'CHGCAR_0.05', 'CHGCAR_0.10', 'CHGCAR_0.15']

# Access summary
summary = results['summary']
print(summary['fukui_type'])
print(summary['calculations'])

# Print formatted summary
print_fukui_summary(wg.pk)
```

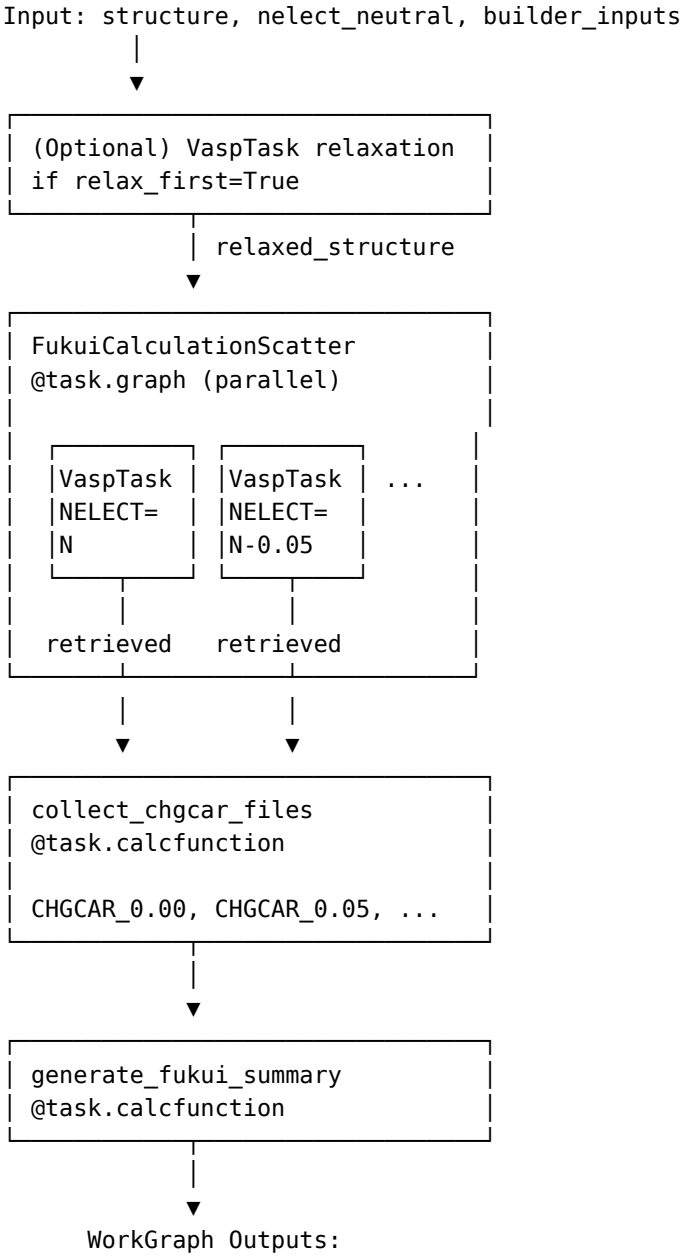
3.6 Exporting CHGCAR Files

To export CHGCAR files for processing with FukuiGrid.py:

```
import os
from pathlib import Path

output_dir = Path('./fukui_chgcars')
output_dir.mkdir(exist_ok=True)

chgcar_folder = results['chgcar_folder']
for fname in chgcar_folder.list_object_names():
    content = chgcar_folder.get_object_content(fname)
    output_path = output_dir / fname

    if isinstance(content, bytes):
        output_path.write_bytes(content)
    else:
        output_path.write_text(content)

print(f"Exported to: {output_dir}")
```

4 Workflow Architecture

4.1 Data Flow Diagram



Listing 2: Fukui workflow data flow

4.2 NELECT Calculation

For **Fukui+** (nucleophilic attack — where electrons are added):

$$\text{NELECT} = N_{\text{neutral}} - \delta N$$

For **Fukui-** (electrophilic attack — where electrons are removed):

$$\text{NELECT} = N_{\text{neutral}} + \delta N$$

δN	**Fukui+ NELECT**	**Fukui- NELECT**
0.00	192.00	192.00
0.05	191.95	192.05
0.10	191.90	192.10
0.15	191.85	192.15

Table 1: Example NELECT values for $N_{\text{neutral}} = 192$

5 Future Development

5.1 Planned Features

The Fukui module will be extended with the following capabilities:

5.1.1 Phase 2: Fukui Function Computation

Integration with FukuiGrid.py for computing the actual Fukui function from CHGCAR files:

- **Polynomial interpolation** of charge densities
- **Derivative extraction** at $\delta N = 0$
- Output: `CHGCAR_FUKUI.vasp` file

5.1.2 Phase 3: Fukui Potential

Calculation of the Fukui potential $v_{f(\vec{r})}$ using different methods:

Method	**Description**
Interpolation	Direct computation from charge densities
Electrodes	Fourier transform-based electrostatic corrections
SCPC	Self-consistent potential correction method

The Fukui potential is useful for:

- Predicting interaction energies with charged adsorbates
- Understanding electrostatic contributions to reactivity

5.1.3 Phase 4: Perturbative Expansion

Implementation of the perturbative expansion model:

$$\Delta U(\vec{r}) = q\Phi(\vec{r}) - q \cdot \Delta N \cdot v_f^{\pm(\vec{r})}$$

Where:

- q is the charge of the active site
- $\Phi(\vec{r})$ is the electrostatic potential
- ΔN is the electron transfer
- $v_f^{\pm}$ is the Fukui potential

This allows prediction of:

- Adsorption site preferences
- Interaction energy landscapes
- Reactivity maps

5.1.4 Phase 5: Workflow Integration

- Integration with main PS-TEROS `build_core_workgraph()`
- Workflow preset: `workflow_preset='fukui_analysis'`
- Automatic Fukui analysis after slab relaxation

6 Troubleshooting

6.1 Common Issues

Issue	Solution
SCF not converging	Use `algo: 'All'`, reduce `amix`, increase `nelm`
CHGCAR not retrieved	Check `ADDITIONAL_RETRIEVE_LIST` includes `'CHGCAR'`
Wrong NELECT	Verify from VASP OUTCAR of a test calculation
Memory issues	Reduce `ncore`, check node memory limits
Workflow stuck	Check `verdi process report <PK>` for sub-process status

6.2 Monitoring Workflows

```
# Check workflow status
verdi process show <PK>

# Detailed task hierarchy
verdi process report <PK>

# Check daemon logs
verdi daemon logshow

# List recent processes
verdi process list -a -p 1
```

6.3 Validating Results

1. **Check convergence**: All calculations should show `electronic_converged: True`
2. **Energy consistency**: Energies should vary smoothly with δN
3. **CHGCAR integrity**: Files should have correct grid dimensions
4. **Physical reasonableness**: Fukui functions should be localized at reactive sites

7 References

1. Parr, R. G., & Yang, W. (1984). Density functional approach to the frontier-electron theory of chemical reactivity. *Journal of the American Chemical Society*, 106(14), 4049-4050.

2. Yang, W., & Mortier, W. J. (1986). The use of global and local molecular parameters for the analysis of the gas-phase basicity of amines. *Journal of the American Chemical Society*, 108(19), 5708-5711.

3. FukuiGrid documentation and examples: https://github.com/rubenfiszel/FukuiGrid

4. PS-TEROS documentation: See `CLAUDE.md` in the repository root.