DFT+U theory

The basic idea behind DFT+U is to treat the strong on-site Coulomb interaction of localized electrons, which is not correctly described by LDA or GGA, with an additional Hubbard-like term. The on-site Coulomb interactions are particularly strong for localized d and f electrons, but can be also important for p localized orbitals. The strength of the on-site interactions are usually described by parameters U (on site Coulomb) and J (on site exchange). These parameters U and J can be extracted from ab-initio calculations, but usually are obtained semi-empirically.

The DFT+U corrections can be introduced in ab initio calculations in different ways. The two main branches are the one introduced by Liechtenstein et al. [Liechtenstein], in which U and J enter as independent corrections in the calculations, and the one proposed by Anasimov et al. [Dudarev], where only a single effective $U_{\text{eff}} = U - J$ parameter accounts for the Coulomb interaction, neglecting thereby any higher multi-polar terms. The latter is the one implemented in GPAW. Thus, the DFT+U totally energy in GPAW is:

$$E_{\text{DFT+U}} = E_{\text{DFT}} + \sum_a \frac{U_{\text{eff}}}{2} \text{Tr}(\rho^a - \rho^a \rho^a),$$

where ρ^a is the atomic orbital occupation matrix. This can be understood as adding a penalty functional to the DFT total energy expression that forces the on site occupancy matrix in the direction of idempotency, i.e. to either fully occupied or fully unoccupied levels.

GPAW implementation

Here are some examples about how to introduce the Hubbard correction U_{eff} in the calculator. For instance if one wants to apply a $U_{\text{eff}} = 6$ eV correction to the d orbitals of manganese atoms, one should include the next parameter to the calculator:

```
setups={'Mn': ':d,6.0'}
```

In the case of p electrons (here nitrogen atom is used as expample), one should use:

```
setups={'N': ':p,6.0'}
```

Here is an example of how to apply a $U_{\text{eff}} = 6$ eV to the d orbitals of Ni in NiO:

```
from ase import Atoms
from gpaw import GPAW, PW, FermiDirac

# Setup up bulk NiO in an antiferromagnetic configuration:
a = 4.19  # lattice constants
b = a / 2**0.5
m = 2.0
atoms = Atoms('Ni2O2',
              pbc=True,
              cell=(b, b, a),
              positions=[(0, 0, 0),
                         (b / 2, b / 2, a / 2),
                         (0, 0, a / 2),
                         (b / 2, b / 2, 0)],
              magmoms=[m, -m, 0, 0])

k = 2  # number of k-points
atoms.calc = GPAW(mode=PW(400),
                  occupations=FermiDirac(width=0.05),
                  setups={'Ni': ':d,6.0'},  # U=6 eV for Ni d orbitals
                  txt='nio.txt',
                  kpts=(k, k, k),
                  xc='PBE')
e = atoms.get_potential_energy()
```

Scaling the Hubbard correction

The projection of the orbitals needed to get the atomic orbiatal coccupation matrix is truncated at the augmentation sphere radius (this is beacause GPAW atomic setups tipically have a bound state projector and an unbound one). Due to this truncation the projection of the wavefunctions onto the atomic orbitals is always <1, being at the maximum the integral of the projected atomic orbital within the augmentation sphere. at this point there are two choices: either we normalize the projection to the value of the integral of the projected atomic orbital within the augmentation sphere or we do not do it.

By default in GPAW the Hubbard corrections is normalized. However, in other PAW codes (e.g. VASP) this is not the standard choice and the Hubbard correction is not normalized. Still, GPAW allows to apply the Hubbard correction without normalization. For instance, in the case of the Nitrogen example before one should write:

```
setups={'N': ':p,6.0,0'}
```

The addition of the 0 at the end of the keyword deactivates the normalization.

The normalization does not have a big influence when the Hubbard correction is applied on d or f orbitals (repeat the calculation for NiO, setting the normalization to 0 and check that the band gap is similar in both cases), since more than 90% of their wavefunctions are within the augmentation sphere. However, this is not the case when the Hubbard correction is applied on p orbitals. Thus, one should expect quite different results in a normalized +U correction calculation vs a non-normalized +U correction calculation when the +U correction is applied to p orbitals. One extreme example of this issue can be observed when the +U corrections are applied to the p orbitals of Nitrogen, when one calculates the atomic nitrogen. In this case we have three p orbital fully occupied (spin up) and three p orbitals empty (spin down). In a first approximation, the occupied orbitals should decrease their energy by $U_{\text{eff}}/2$ with respect to a calculation with no +U corrections, whereas the empty orbitals should increase their energy by the same amount. The next example show how this is only true if the +U correction is normalized.

```
from ase import Atoms
from gpaw import GPAW

n = Atoms('N', magmoms=[3])
n.center(vacuum=3.5)

# Calculation with no +U correction:
n.calc = GPAW(mode='lcao',
              basis='dzp',
              txt='no_u.txt',
              xc='PBE')
e1 = n.get_potential_energy()

# Calculation with a correction U=6 eV normalized:
n.calc.set(setups={'N': ':p,6.0'}, txt='normalized_u.txt')
e2 = n.get_potential_energy()

# Calculation with a correction U=6 eV not normalized:
n.calc.set(setups={'N': ':p,6.0,0'}, txt='not_normalized_u.txt')
e3 = n.get_potential_energy()
```

Here are the resulting 2p-splitting:

no U	4.224
normalized U	10.996
not normalized U	4.765

(see also check.py).

References

[Liechtenstein] A. I. Liechtenstein, V. I. Anisimov and J. Zaane, Phys. Rev. B 52, R5467 (1995).

[Dudarev] S. L. Dudarev, G. A. Botton, S. Y. Savrasov, C. J. Humphreys and A. P. Sutton, Phys. Rev. B 57, 1505 (1998).