DeltaSpin (Self-Adaptive Spin Constraint in VASP)

Zefeng Cai

July 2021

Contents

1 Terms of Use

PLEASE READ THESE TERMS OF USE CAREFULLY. By using the "DeltaSpin" software package developed by Zefeng Cai and Ben Xu (hereinafter referred to as "The Provider"), you acknowledge that you have read, understood, and agree to abide by these Terms of Use.

1. **License Requirement**

   The Provider's software package, "DeltaSpin", is designed to operate with the Vienna Ab initio Simulation Package ("VASP"). Therefore, a valid and legally obtained license for VASP is required for the usage of "DeltaSpin". Unauthorized use of "DeltaSpin" without a valid VASP license is strictly prohibited.

2. **User Responsibility and Compliance**

   As a user, it is your responsibility to secure and maintain a valid VASP license. If any copyright infringement or unauthorized use of VASP occurs due to non-compliance with the required license terms, you will be solely liable for the consequences.

3. **Consequences of Copyright Violation**

   By agreeing to these Terms of Use, you acknowledge that any legal consequences, penalties, damages, or other adverse effects arising from a copyright violation related to VASP software are solely your responsibility, including violations related to the use of "DeltaSpin" without a valid VASP license.

4. **Indemnification**

   You agree to indemnify and hold harmless the Provider, its affiliates, and their respective officers, directors, employees, agents, and partners from any claims, liabilities, damages, losses, and expenses (including, but not limited to, reasonable legal fees) arising from your violation of these Terms of Use or your use of the "DeltaSpin" software package.

5. **Software Activation and Acceptance of Terms**

   Activating "DeltaSpin" with the provided activation code, or using "DeltaSpin" in any capacity, indicates your explicit acceptance of these Terms of Use. Your agreement to these terms is binding and carries the same legal effect as a physical signature.

These Terms of Use constitute the entire agreement between you and the Provider. If you do not agree to these terms, please refrain from using the "DeltaSpin" software package.

2 Docker Image User Guide

For the following commands, replace the values within the angle brackets (<>) with relevant values for your system:

1. Pull the DeltaSpin image from ghcr.io:

```
docker pull ghcr.io/caizefeng/deltaspin:latest
```

2. Start a container using this image. Please note that you should allocate at least 16 GB of shared memory when starting the container from this image:

```
docker run -d --shm-size=<size of shared memory> \
--name=deltaspin ghcr.io/caizefeng/deltaspin:latest
```

3. Enter the container:

```
docker exec -it deltaspin bash
```

4. Navigate to your calculation directory and place the activation code in the same directory as the `INCAR`, `POSCAR`, `POTCAR`, and `KPOINTS` files:

```
cd /root/DeltaSpin/examples/metal/Fe
cp <path to your activation code file> .
```

5. Run `vasp_deltaspin`:

```
mpirun -np <number of threads> /root/DeltaSpin/bin/vasp_deltaspin
```

3 Getting Started

1. *Determine Wigner-Seitz radius (in Å) for each atom type in the system.*

   Read the `RWIGS` entry presented in `POTCAR` and use that value for INCAR. Note that this value should **remain constant** in every single calculation of this system.

   (OPTIONAL) Though tedious and not recommended, you can also determine the Wigner-Seitz radius through a Bader-type **spin density** analysis.

2. *Determine the magnetic ground state of the system.*

   Set up `INCAR` with the following parameters:

```
RWIGS = <values from the 1st step>
I_CONSTRAINED_M = 2
SCTYPE = 0
LAMBDA = 0
```

   and run `vasp_deltaspin`. When the calculation is finished, the last `MW_current` in `OSZICAR` represents the new ground-state `RWIGS`-defined magnetic moments.

   Note that `MAGMOM` should be close to the ground-state.

   You can add the following two tags to increase precision in ANY non-collinear calculation.

```
LASPH = .TRUE.
GGA_COMPAT = .FALSE.
```

3. *Constrain the system to the desired magnetic configuration.*

   Refer to the `INCAR` template in `DeltaSpin/templates` directory. Set `M_CONSTR` to any value you want to constrain the magnetic moments to.

   Run `vasp_deltaspin`.

   Note that the ground-state magnetic configuration for reference, throughout all DeltaSpin calculations of this specific material system, is what we obtained in the 2nd step instead of any empirical value or `magnetization` in `OUTCAR` [1]. Different definitions of "magnetization" used in VASP will be discussed in another section.

4. *(OPTIONAL) Non-Self-Consistent (NSCF) calculation.*

   Set `INCAR-LAMDA` to the last value of `OSZICAR-lambda` from the same **SCF** calculation that `CHGCAR` is from.

   Also, complete the standard procedure for a typical VASP NSCF calculation, like preparing a `CHGCAR` and setting `INCAR-ICHARG >= 10`.

   Run `vasp_deltaspin`.

[1] This means that if you set `M_CONSTR` to the last `MW_current` in `OSZICAR` from the 2nd step and apply the constraining, the energy will be exactly the same.

4 INCAR Template

```
SYSTEM = NiO
#GGA = PE

#Electronic minimization
PREC = Accurate                 # Accurate is always preferred since non-collinear magnetization is very sensitive to the completeness
      of FFT grid.
ENCUT = 600
EDIFF = 1E-9
LORBIT  = 11
LREAL = .FALSE.
ISTART  = 0
NPAR  = 4
#ICHARG =  1
VOSKOWN = 1
LWAVE = .FALSE.
LCHARG = .FALSE.
NELMIN = 10
NELM = 500

#IBRION = 2 #OPT
#EDIFFG = -1E-3 #OPT
#ISIF = 3 #OPT
#NSW = 500 #OPT
#POTIM = 0.2  #OPT

#DOS
ISMEAR = -5
SIGMA = 0.2
#NEDOS = 3000 #DOS

#Magnetism
# The DEFINITION of atomic spins which are constrained (M_CONSTR in INCAR, MW in constrmag.F/OSZICAR):
# \vec{M}_{I} =
# \int_{\Omega_{I}} \vec{m}(\mathbf{r}) W_{I}(\mathbf{r}) d\mathbf{r}

#External magneitc field (Zeeman field)
Bfield = 0 0 0                  # External field in Cartesian coordinate (X,Y,Z), The unit is Tesla.                !IMPORTANT!

#Non-collinear
ISYM = -1
RWIGS = 1.286 0.820             # Wigner-Seitz radius (in angstrom) for each atom type in the system.                !IMPORTANT!
GGA_COMPAT = .FALSE.            # Restore the full lattice symmetry for gradient corrected functionals. Recommended.
LASPH = .TRUE.                  # Non-spherical contribution to the gradient of the density. Recommended.
ISPIN = 2                       # Does not matter in non-collinear calculation.
LNONCOLLINEAR = .TRUE.
LSORBIT  = .TRUE.

#SASC
I_CONSTRAINED_M = 2
MAGMOM = 0.75914 0.16146 -1.00688 -0.67954 -0.91284 0.56668 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
M_CONSTR = 0.75914 0.16146 -1.00688 -0.67954 -0.91284 0.56668 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000       !IMPORTANT!
SCTYPE = 1                      # Type of SASC calculation   0: noSASC  1: SASC(L)  2: SASC(Q)  3: SASC(QL)          !IMPORTANT!

#SASC(L)
IDECOSC = 0                     # Whether Lagrangian coefficients are coupled   0: coupled  1: decoupled between atoms (deprecated)  2:
      decoupled between every Cartesian component (deprecated)
LAMBDA   = 12*0                 # Initial Lagrangian coefficients for SASC(L)
CONSTRL  = 6*1 6*0              # Whether the component is constrained or not   0: not constrained  1: constrained  !IMPORTANT!
NSC = 100                       # Maximum number of steps for each inner optimization
NSCMIN = 2                      # Minimum number of steps for each inner optimization
SCDIFF = 1E-8                   # Convergence criterion of iteration (RMS)
SCCONVB_GRAD = 1.9 -1           # Break condition for local gradients of spins w.r.t. Lagrangian coefficients (local field)   -1: off
      for the element

IDECAY_GRAD = 2                 # Decay policy of gradient break condition   0: no decay  1: exponential decay  2: step down
SCDECAY_GRAD = 0.9              # Exponetial base of decayed gradient break condition (IDECAY_GRAD = 1)
SCGRADB = 0.1                   # Lower bound of exponetially decayed gradient break condition
NGRAD = 2                       # Number of "steps" the gradient break condition curve includes (IDECAY_GRAD = 2)
NGRAD_STEP = 200 400            # Indices in the outer loop where the gradient break condition discontinuously decay
NGRAD_VALUE = 1.7 -1 \          # SCCONVB_GRAD for corresponding "steps"
              1.5 -1

SCDECAY = -1                    # Exponential decay of convergence criterion   -1: no decay  >0,<1: exponential base
SCDIFFB = 1E-7                  # Lower bound of criterion decay, not gonna work if SCDECAY = -1
SCCONVB = -1                    # Break condition for Lagrangian coefficients difference between two inner steps   -1: this condition
      is off
INISC = 0.01                    # Initial trial step size
SCCUT = 3                       # Restriction of step size
NELMSCI = 1                     # Number of electronic steps BEFORE SASC(L)
NELMSCT = 0                     # Number of INTERMEDIATE normal electronic steps
IALGOSC = 1                     # Inner optimization algorithm   1: CG(F-R) 2: CG(P-R) 3: CG(H-S) 4: CG(D-Y)
IALGOSC_DIAG = 1                # Inner diagonalization algorithm   1: sub-space rotation  2: Blocked-Davidson  3: sub-space rotation
      with B-D for the last step
LCUTSC_TRIAL = .TRUE.           # Whether trial step size is updated acoording to last-step optimal value   TRUE: update  FALSE: do not
      update
LDESC = .FALSE.                 # Debug mode   TRUE: on  FALSE: off

#SASC(Q)
```

```
LAMBDA_Q   = 10 10 10 10 10 10 # Initial Lagrangian coefficients for SASC(Q)
CONSTRL_Q  = 1 1 1 1 1 1       # Whether the component is constrained or not   0: not constrained  1: constrained
NSC_Q = 500                    # Maximum number of steps for SASC(Q) iteration
SCDIFF_Q = 3E-4                # Convergence criterion of iteration (+Inf-Norm)
EDIFF_Q = 1E-6                 # Global break condition (EDIFF) for SASC(Q)
INISC_Q = 0.1                  # Initial trial step size
LDESC_Q = .FALSE.              # Debug mode   TRUE: on  FALSE: off

#Orbital momemts
#LORBMOM = .TRUE.

#Mixer
AMIX       = 0.2
BMIX       = 0.0001
AMIX_MAG   = 0.8
BMIX_MAG   = 0.0001

#LSDA+U
LDAU       = .TRUE.
LDAUTYPE   = 2
LDAUL      = 2 -1
LDAUU      = 5.3 0.0
LDAUJ      = 0.0 0.0
LDAUPRINT = 2
LMAXMIX    = 4
```

5 Selected Multi-step Use Cases

5.1 Calculate the effective field of the original ground-state configuration under Zeeman field

1. Relax the system to reach the ground-state magnetic configuration without an external field.

```
Bfield = 0 0 0
I_CONSTRAINED_M = 2
SCTYPE = 0
LAMBDA = 0
```

2. Apply a finite external field `B`. Constrain the size and direct of the magnetic moments using DeltaSpin.

```
Bfield = <B_x> <B_y> <B_z>
I_CONSTRAINED_M = 2
SCTYPE = 1
LAMBDA = <number of atoms>*0
M_CONSTR = <the last "M_current" from step 1>
```

5.2 Calculate the effective field of a direction-constrained magnetic configuration

1. Constrain the direction of the magnetic moments using the original VASP algorithm that is compatible with DeltaSpin.

```
I_CONSTRAINED_M = 1
SCTYPE = 0
LAMBDA = <some positive real number>
```

2. Constrain the size and direct of the magnetic moments using DeltaSpin.

```
I_CONSTRAINED_M = 2
SCTYPE = 1
LAMBDA = <number of atoms>*0
M_CONSTR = <the last "M_current" from step 1>
```

5.3 Calculate the electronic structure of a fully-constrained configuration

1. Self-consistently constrain the size and direct of the magnetic moments using DeltaSpin.

```
ISTART = 0
ICHARG = 2
LAMBDA = <number of atoms>*0
LWAVE = .TRUE.
LCHARG = .TRUE.
```

2. Change to a desired k-point grid/path. Run DeltaSpin non-self-consistently.

```
ISTART = 1
ICHARG = 11
LAMBDA = <the last "lambda" from step 1>
```

6 Compatibility

Using `SCTYPE=0`, *DeltaSpin* is now compatible with almost all functionalities from the original VASP, including

1. Constrained local moments approach (direction, `I_CONSTRAINED_M = 1`)
2. Constrained local moments approach (size and direction, `I_CONSTRAINED_M = 2`)
3. Spin spiral (`LSPIRAL=.TRUE.`)

7 Strategy

When the energy difference (dE) keeps fluctuating around a value above the normal EDIFF (approximately 1E-00 ⇔ 1E-02 eV), consider the following steps:

1. Decrease the values of `AMIX` and `AMIX_MAG`, and usually, it is necessary to increase `NELM` at the same time. To avoid out-of-memory issues caused by the increasing number of electronic steps, you may also need to allocate more memory when specifying the computing resources.
2. Test for a suitable `SCCONVB_GRAD`. Modifying this value might have a significant impact on convergence. Some values of `SCCONVB_GRAD` may result in a complete corruption of the diagonalization process.

8 Magnetization in VASP

`OUTCAR` *magnetization* (`magnetization (x,y,z)` in `OUTCAR`), is the integral of spin density in an area $\Omega_{\mathbf{R}}^{\mathrm{AE}}$, where AE partial wave has integer occupancy, which can be slightly different from the augmentation sphere $\Omega_{\mathbf{R}}$.

$$\mathbf{M}_{\mathrm{OUTCAR}} = \int_{\Omega_{\mathbf{R}}^{\mathrm{AE}}} \rho^1(r) \tag{1}$$

$$\rho^1(r) = \sum_{n,(i,j)} f_n \langle \tilde{\Psi}_n | \tilde{p}_i \rangle \langle \phi_i | r \rangle \sigma \langle r | \phi_j \rangle \langle \tilde{p}_j | \tilde{\Psi}_n \rangle \tag{2}$$

where

$$\int_{\Omega_{\mathbf{R}}^{\mathrm{AE}}} \langle \phi_i | r \rangle \langle r | \phi_j \rangle = \delta_{ij} \tag{3}$$

Because

$$\begin{aligned} \rho(r) &= \tilde{\rho}(r) + \rho^1(r) - \tilde{\rho}^1(r) \\ &= \sum_n f_n \langle \tilde{\Psi}_n | r \rangle \sigma \langle r | \tilde{\Psi}_n \rangle + \rho^1(r) - \sum_{n,(i,j)} f_n \langle \tilde{\Psi}_n | \tilde{p}_i \rangle \langle \tilde{\phi}_i | r \rangle \langle r | \tilde{\phi}_j \rangle \sigma \langle \tilde{p}_j | \tilde{\Psi}_n \rangle \end{aligned} \tag{4}$$

and

$$\sum_i |\tilde{\phi}_i\rangle\langle\tilde{p}_i| = 1 \text{ within } \Omega_{\mathbf{R}} \tag{5}$$

If we assume $\Omega_{\mathbf{R}}^{\mathrm{AE}} \approx \Omega_{\mathbf{R}}$, we have

$$\mathbf{M}_{\mathrm{OUTCAR}} = \int_{\Omega_{\mathbf{R}}^{\mathrm{AE}}} \rho^1(r) \approx \int_{\Omega_{\mathbf{R}}} \rho(r) \tag{6}$$

`OSZICAR` *magnetization* (`M_current` in `OSZICAR`), is the integral of spin density in a user-defined real-space sphere.

$$\mathbf{M}_{\mathrm{OSZICAR}} = \int_{\Omega'_{\mathbf{R}}} \rho(r) \tag{7}$$

Weighted `OSZICAR` *magnetization* (`MW_current` in `OSZICAR`), is the weighted integral of spin density in a user-defined real-space sphere. Prevalent weight functions are Bessel functions.

$$\mathbf{M}_{\mathrm{OSZICAR}}^{\mathrm{W}} = \int_{\Omega'_{\mathbf{R}}} W(r)\, \rho(r) \tag{8}$$

If we define $\Omega'_{\mathbf{R}}$ using WS radii (`RWIGS`) in `POTCAR`, that is $\Omega'_{\mathbf{R}} \approx \Omega_{\mathbf{R}}$,"*OUTCAR magnetization*" and "*OSZICAR magnetization*" will be closer (closer than partial core radius `RPCOR` and outmost cutoff radius `RCORE`), but still have a notable difference. That's because these two have totally different origins.

Most importantly, both of them are based on the muffin-tin approximation and the summation of all atomic magnetic moments won't be equal to total magnetization. It's because there always exist some gaps in between those spheres.

The only well-defined quantity in DFT is "total magnetization", the all-space integral of charge density, which is `mag` in `OSZICAR`.

To get access to a new definition of atomic magnetization, the summation of which is exactly the all-space integral, one could apply Bader analysis on spin density.

9 Disclaimer

VASP is a proprietary software, ensure that you possess an appropriate license to use it.