# User Guide
# for Gilbert Damping

# 1 Introduction

Understanding the behaviour of magnetic moments and their interactions with external perturbations is crucial for the development of efficient and reliable spin-based devices. Among the various parameters characterizing this dynamics, Gilbert damping[4] and magnetic moment of inertia play pivotal roles. The fundamental semi-classical equation describing the magnetisation dynamics using these two crucial parameters is the Landau-Lifshitz-Gilbert (LLG) equation[2, 3], given by:

$$\frac{\partial \boldsymbol{M}}{\partial t} = \boldsymbol{M} \times \left( -\gamma \boldsymbol{H} + \frac{\hat{\boldsymbol{\alpha}}}{M} \frac{\partial \boldsymbol{M}}{\partial t} + \frac{\hat{\mathbf{I}}}{M} \frac{\partial^2 \boldsymbol{M}}{\partial t^2} \right) \tag{1}$$

where $\mathbf{M}$ is the magnetisation, $\mathbf{H}$ is the effective magnetic field including both external and internal fields, $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{I}}$ are the Gilbert damping and moment of inertia tensors with the tensor components defined as $\alpha^{\mu\nu}$ and $\mathrm{I}^{\mu\nu}$, respectively, and $\gamma$ is the gyromagnetic ratio.

The expressions for the damping and inertia involve integration over crystal momentum $\boldsymbol{k}$ in the first Brillouin zone. Accurate evaluation of the integrals involved require a dense $\boldsymbol{k}$-point mesh of the order of $10^6 - 10^8$ points for obtaining converged values. Calculating these quantities using full *ab initio* DFT is hence time-consuming. To overcome this problem, here we propose an alternative. To begin with, the first principles calculations are done on a coarse $\boldsymbol{k}$ mesh instead of dense $\boldsymbol{k}$ mesh. We then utilize the maximally localised Wannier functions (MLWFs)[5] for obtaining the interpolated integrands required for the denser $\boldsymbol{k}$ meshes. In this method, the gauge freedom of Bloch wavefunctions is utilised to transform them into a basis of smooth, highly localised Wannier wavefunctions. The required real space quantities like the Hamiltonian and torque matrix elements are calculated in the Wannier basis using Fourier transforms. The integrands of integrals can then be interpolated on the fine $\boldsymbol{k}$ mesh by an inverse Fourier transform of the maximally localised quantities, thereby enabling the accurate calculations of the damping and inertia.

Within the so-called torque-torque correlation model, the Gilbert damping tensor can be expressed as follows:

$$\alpha^{\mu\nu} = \frac{g}{M_s \pi} \int \int \left( -\frac{df(\epsilon)}{d\epsilon} \right) Tr[\Gamma^{\mu}(\Im G)(\Gamma^{\nu})^{\dagger}(\Im G)] \frac{d^3\boldsymbol{k}}{(2\pi)^3} d\epsilon \tag{2}$$

Here the trace, denoted by $Tr$, goes over band indices, $f(\epsilon)$ is the Fermi function, $(\Im G)$ is the imaginary part of Green's function $G = (\epsilon + \iota\eta - \mathcal{H})^{-1}$ with $\eta$ as a broadening parameter, $M_s$ is the saturation magnetization in units of the Bohr magneton, $\Gamma^{\mu} = [\sigma^{\mu}, \mathcal{H}_{so}]$ is the $\mu^{th}$ component of the torque operator or matrix, $\mu = x, y, z$. $\alpha$ is a dimensionless parameter.

The whole procedure to calculate the torque matrix elements, Green's function is described in Ref. [1]. The key expression in which the python scripts `readwfc.py` and `transformation.py` are used is the expression (17) in the Ref.

[1], which involves calculating transformation between SO and SP basis. The expression is:

$$\begin{aligned}
T_{mn}^{\boldsymbol{RR'}} &= \frac{1}{N^2}\sum_{\boldsymbol{q}} e^{\iota \boldsymbol{q}\cdot(\boldsymbol{R}-\boldsymbol{R'})}[\tilde{\mathcal{U}}^{\boldsymbol{q}\dagger}(N\mathcal{V}^{\boldsymbol{q}})\mathcal{U}^{\boldsymbol{q}}]_{mn} \\
&= \frac{1}{N}\sum_{\boldsymbol{q}} e^{\iota \boldsymbol{q}\cdot(\boldsymbol{R}-\boldsymbol{R'})}[\tilde{\mathcal{U}}^{\boldsymbol{q}\dagger}\mathcal{V}^{\boldsymbol{q}}\mathcal{U}^{\boldsymbol{q}}]_{mn} \qquad (3)
\end{aligned}$$

where $\mathcal{V}_{pl}^{\boldsymbol{q}} = \langle\tilde{\psi}_{p\boldsymbol{q}}|\psi_{l\boldsymbol{q}}\rangle$. After calculating the transformation in pre-processing step, we post process using the code `gilbert.py` to calculate Gilbert Damping. The process is described in the next section.

## 2 Preprocessing

Preprocessing involves first running the SCF, NSCF, and Wannier90 calculations for the spin-orbit and spin-polarised cases. Then, we use the python scripts `readwfc.py` and `transformation.py` find the transformation matrices needed to calculate damping. The steps involved are:

1. For SCF calculations, run the following commands:

```
$ cd <path to GilbertDamping>/examples/Fe/scf/
$ pw.x -i Fe.scf.in
```

2. For NSCF calculation:

```
$ cp -r Fe.save/ Fe.xml ../nscf/
$ cd ../nscf/
$ pw.x -i Fe.nscf.in
```

3. For WANNIER90 calculation:

```
$ cp -r Fe.save/ ../wannier/
$ cd ../wannier/
$ wannier90.x -pp Fe
$ pw2wannier90.x -i Fe.pw2wan.in
$ wannier90.x Fe
```

This generates `Fe.chk` and `Fe.eig` needed for postprocessing.

4. Repeat 1,2 and 3 for the spin-polarised calculations in `/examples/Fe/spinpol/` directory.

```
$ cd ../damping/
$ cp ../scf/Fe.scf.out .
$ cp ../wannier/Fe.chk .
$ cp ../wannier/Fe.spn .
$ cp ../wannier/Fe.eig .
$ cp ../spinpol/wannup/Feup.chk .
$ cp ../spinpol/wannup/Feup.eig .
$ cp ../spinpol/wanndn/Fedn.chk .
$ cp ../spinpol/wanndn/Fedn.eig .
```

5. Reading `wfc.dat`, `wfcup.dat`, and `wfcdw.dat`. Here nk=4096 (number of k-points used in the NSCF calculation).

```
$ cd /examples/Fe/wannier/Fe.save/
$ python ../../../../SRC/readwfc.py wfc nk
$ cd /examples/Fe/spinpol/wannierup/Fe.save/
$ python ../../../../../SRC/readwfc.py wfcup nk
$ python ../../../../../SRC/readwfc.py wfcdw nk
```

This will generate the `.npy` files for the wavefunctions needed in the `transformation.py` script.

6. After reading the wavefunctions, we calculate the tranformations needed in the SO and SP Bloch basis.

```
$ cd /examples/Fe/damping/
$ python ../../../SRC/transformation.py Fe
```

This will create `wanniertrans.npy` with the transformation matrices.

## 3 Input file

The structure of input file `Fedamping.in` needed to run the `gilbert.py` to calculate the Gilbert damping after pre-processing using `pw.x`, `wannier90.x`, `readwfc.py` and `transformation.py` is as follows:

```
seedname: Fe             ### same as Wannier seedname
temperature: 300         ### Temerature used in Fermi function
eta: 0.1                 ### Broadening used in Green's function
grid: 24                 ### k-grid for interpolation (same in all directions)
magnetisation: 2.3812    ### specify from scf output (in Bohr Magnetons)
fermienergy: 17.6254     ### in eV
```

Sample input files are given in the `/examples/Fe/damping/` and `/examples/Fe/Ref/` folders which includes results for the damping reported in Fig. 3 of the Ref. [1]

## 4 Post Processing

Run the following command to calculate the damping for the parameters specified in the input file. The number of cores N to be used specifies the number of energy points between $-\mathrm{Nk_BT}/2$ to $\mathrm{Nk_BT}/2$. The integration is performed using the trapezoidal rule. Use N as large as possible that the derivative of the Fermi function is effectively zero above the range defined by this.

```
cd <path to GilbertDamping>/examples/Fe/damping/
mpiexec -n N python <path to GilbertDamping>/SRC/gilbert.py Fedamping.in
```

# 5 Installation and Prerequisites

The GilbertDamping code is compatible with python3 and it requires numpy, mpi4py, functools, multiprocessing, itertools, and pyfftw libraries. To install these libraries, you can type the following command.

```
$ pip install numpy mpi4py functools multiprocessing itertools pyfftw
```

To install the code, first download the tar zip file and extract:

```
$ tar -xvzf GilbertDamping.tar.gz
```

The source code is in /SRC/ directory and the examples are in the /examples/ directory.

# References

[1] Robin Bajaj, Seung-Cheol Lee, H. R. Krishnamurthy, Satadeep Bhattacharjee, and Manish Jain. Calculation of gilbert damping and magnetic moment of inertia using the torque-torque correlation model within an ab initio wannier framework. *Phys. Rev. B*, 109:214432, Jun 2024.

[2] Satadeep Bhattacharjee, Lars Nordström, and Jonas Fransson. Atomistic spin dynamic method with both damping and moment of inertia effects included from first principles. *Physical Review Letters*, 108(5):057204, 2012.

[3] M-C Ciornei, JM Rubí, and J-E Wegrowe. Magnetization dynamics in the inertial regime: Nutation predicted at short time scales. *Physical Review B*, 83(2):020410, 2011.

[4] Thomas L Gilbert. A phenomenological theory of damping in ferromagnetic materials. *IEEE Transactions on Magnetics*, 40(6):3443–3449, 2004.

[5] Nicola Marzari, Arash A Mostofi, Jonathan R Yates, Ivo Souza, and David Vanderbilt. Maximally localized Wannier functions: Theory and applications. *Reviews of Modern Physics*, 84(4):1419, 2012.