SLaSi v0.2 User Guide

Oleksandr Pylypovskyi

February 17, 2016

Contents

1 General Information **3**

2 Command Line Parameters **4**

3 Configuration file **6**

4 Units **7**

5 Geometry **8**

6 Magnetic Parameters **8**

6.1 Exchange 8

6.2 Anisotropies 9

6.3 Surface Anisotropy 10

6.4 Coordinate-Dependent Anisotropy 11

6.5 Dipolar Interaction 12

7 External Magnetic Fields **13**

8 Initial Magnetization Distribution **15**

9 Integration Parameters **16**

10 Miscellaneous **17**

11 Examples **18**

11.1 Uniformly Magnetized Sphere 18

11.2 Walker's limit 18

1 General Information

SLaSi is a C-written spin-lattice simulator, initially based on Fortran source codes of vortex...

SLaSi solves Landau–Lifshitz–Gilbert equation for spin lattice of $\mathcal{N}$ unit magnetic moments

$$\frac{\mathrm{d}\boldsymbol{m}_n}{\mathrm{d}t} = -\gamma \boldsymbol{m}_n \times \boldsymbol{H}_n^{\mathrm{eff}} + \alpha \boldsymbol{m}_n \times \frac{\mathrm{d}\boldsymbol{m}_n}{\mathrm{d}t}, \quad n = \overline{1, \mathcal{N}}, \tag{1}$$

where $\gamma = g\mu_B/\hbar$ is the gyromagnetic ratio, g is the Landé factor, $\mu_B > 0$ is the Bohr magneton, $\hbar$ is the Plank constant (see Sec. 4 for details) and α is the Gilbert damping. The effective field

$$\boldsymbol{H}_n^{\mathrm{eff}} = -\frac{1}{M_S}\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}, \tag{2}$$

where M_S is a saturation magnetization and $\mathcal{H}$ is a Hamiltonian of the given spin lattice, see Sec. 6 and 7 for details.

General features of SLaSi include possibility to calculate interaction between magnetic dipoles using MPI library for parallel computing, single-ion, exchange and surface anisotropy, coordinate-dependent anisotropy, a few types of built-in magnetic fields which can vary in space and/or time and arbitrary geometry in assumption that Heisenberg exchange Hamiltionian is valid for the given lattice.

2 Command Line Parameters

SLaSi supports the following command line keys:

- `-h` or `--help`: print a help message in the console and exit.
- `-i <filename>` or `--input <filename>`: direclty specify the name of configuration file with parameters of simulations. Otherwise it is expected as the first parameter of program name.
- `-n` or `--print-nan`: print word "nan" for nodes with zero magnetic moment in the snapshot files.
- `--no-verbose`: do not make noise in console during calculations.
- `-C` or `--continue <filename>`: continue previous calculations and write the output in log file ¡filename¿ without eraseing previous data in log file. You will require keys `-N`, `-T` and `-o` to change the initial time, snapshot number and directly specify output name.
- `-o` or `--output <filename>`: directly specify the base name of output files. Default: if the *.log file with base equivalent to the base of the configuration file exists, adds "_XX", where XX is increment.
- `-N <integer>` or `--number <integer>`: initial number for the snapshots. Default: numbers start from zero.
- `-T <float>` or `--time <float>`: initial time for snapshots. Default: time starts from zero.
- `--no-first`: do not print snapshot for zero time. Firs snapshot will have number 1.
- `--version`: print program version and exit.
- `-c <parameter> <value>` or `--change <parameter> <value>`: directly change value of parameter, which was specified in the configuration file.

Note that keys cannot be combined like -abcdefg.

In the following listing SLaSi starts on two MPI nodes with `./helix.inp` configuration file. It prints a minimum of information to stdout and prints nans to the output files for the lattice sites with zero magnetic moment.

Listing 1: Typical start

```
user@host $ mpirun -np 2 slasi ./helix.inp -n --no-verbose
```

Usage of SLaSi on computing clusters is dependent on particular settings. It may be the following:

Listing 2: Start on cluster (part of PBS script)

```
$MPI_RUN  -hostfile $PBS_NODEFILE $PBS_O_HOME/bin/slasi
   task.inp --no-verbose -n
```

The following listing demonstrates how to continue "disk" task from the 1574-th snapshot and time of 15740 units.

Listing 3: Continue the task

```
$MPI_RUN  -hostfile $PBS_NODEFILE $PBS_O_HOME/bin/slasi
   disk.inp --no-verbose -n -o disk -C disk.log -N 1574 -T
   15740 --no-first
```

3 Configuration file

All parameters (geomety, material, initial distribution etc) are gived for the SLaSi by a configuration file in format "parameter = value", see examples in the sections below. The SLaSi is called from the command line as

```
user@host $ mpirun -np 4 ./slasi observe.inp
```

Here the program starts from the current directory via MPI on the 4 nodes. The initial parameters are stored in the file observe.inp (really, a file name can be any, the extension is not necessary). The file observe.inp is stored in the same directory as SLaSi in the example above. The configuration file can include comment lines, which start from symbol #. The first symbol in string cannot be space, only letter or #. Strings are given inside the double quotes, for example `"../geometries/mySample.geom"`. The letter case is important: "parameter" and "parAmeter" will be different variables for the SLaSi.

Depending on information inside configuration file, additional files with initial magnetization distribution and geometry of the sample could be needed.

4 Units

Currently only the reduced unit system is supported. SLaSi uses value of the Planck constant $\hbar = 1$, Landé factor $g = 2$ and $\mathcal{S} = 1/2$. The Bohr magneton is defined manually (it gives proportion between exchange and dipolar interaction or allows to rescale values of the saturation fields depending on given interactions).

5 Geometry

Geometry is defined via **`geometry`** parameter. It accepts the following values:

- `circle` — a sample of cylindrical shape in xy plane, diameter **`D`** and height **`h`** are needed;
- `rectangle` — a rectangular parallelepiped, length **`a`**, width **`b`** and height **`h`** are needed;
- `ellipse` — a sample of elliptical shape in xy plane, two axes **`a`** and **`b`** and height **`h`** are needed;
- `file` — indexes of sites of a cubic lattice with nonzero magnetic moments are read from the file **`geomFile`**, length **`a`**, width **`b`** and height **`h`** are needed;
- `curved` — indexes of sites of a deformed lattice with 6 neighbours per inner site are read from the file **`geomFile`** where coordinates of all , length **`a`**, width **`b`** and height **`h`** are needed. *At the moment all indexes in the given rectangle should be given.*

SLaSi generates a cubic lattice where each inner site is connected via exchange interaction with 6 nearest neighbours. For `curved` it is considered as slightly deformed, so Heisenberg Hamiltonian remains valid. Such geometries like `circle` are placed inside this lattice where sites with nonzero magnetic moment form a given shape.

6 Magnetic Parameters

6.1 Exchange

SLaSi calculates dynamics of reduced magnetic moments $\boldsymbol{m}_i = \boldsymbol{M}_i/M_S$ with saturation magnetization

$$M_S = \frac{g\mu_B}{a^3}\mathcal{S}, \tag{3}$$

where $a = 1$ is the lattice constant and $\mathcal{S} = 1/2$ is the length of classical spin.

The exchange Hamiltonian has the following form:

$$\mathcal{H}^{\text{ex}} = -\frac{J_0\mathcal{S}^2}{2}\sum_{n,\delta}\boldsymbol{m}_n\boldsymbol{m}_{n+\delta}, \tag{4}$$

where J_0 is the exchange integral and index δ runs over the nearests neighbours. The corresponding inhomogeneous exchange energy reads

$$E^{\text{ex}} = A \int \sum_{\nu=x,y,z} (\nabla m_\nu)^2 \mathrm{d}\boldsymbol{r}, \quad A = \frac{J_0 \mathcal{S}^2}{2a}. \tag{5}$$

When SLaSi saves value of exchange Hamiltionian (4) the total number of spins, i. e. homogeneous exchange is subtracted.

The lattice exchange constant is named as **`J0`** (default value equals 1). For the dipolar interactions Bohr magneton **`muB`** is the nesessary parameter.

Listing 4: Exchange integral

```
# Value of the exchange integral
J0 = 1.0
# Bohr magneton
muB = 0.1
```

6.2 Anisotropies

SLaSi supports exchange, single-ion and surface anisotropy. Hamiltonians of exchange and single-ion anisotropy read

$$\mathcal{H}^{\text{ex. an.}} = -\frac{a^2 \mathcal{S}^2}{2} \sum_{n,\delta} \left[J_x m_n^x m_{n+\delta}^x + J_y m_n^y m_{n+\delta}^y + J_z m_n^z m_{n+\delta}^z \right], \tag{6}$$

$$\mathcal{H}^{\text{s.-i. an.}} = -\frac{\mathcal{S}^2}{2} \sum_{n} \left[K_x (m_n^x)^2 + K_z (m_n^z)^2 \right]. \tag{7}$$

Single-ion anisotropy by **`Kx`** and **`Kz`** and exchange anisotropy is defined by parameters **`Jx`**, **`Jy`**, **`Jz`** (Default values for both of them equal 0). Exchange interction with exchange anisotropy is considered in the following form:

$$J = J_0 + \frac{a^2}{2} J_i, \quad i = 1, 2, 3, \tag{8}$$

where indexes 1, 2, 3 enumerate axes x, y and z respectively.

The macroscopic anisotropy coefficients equal

$$\mathcal{K}_x = \frac{\mathcal{S}^2}{2a^3} K_x + \frac{a\mathcal{S}^2}{2} \frac{J_x - J_y}{6}, \tag{9}$$

$$\mathcal{K}_z = \frac{\mathcal{S}^2}{2a^3} K_z + \frac{a\mathcal{S}^2}{2} \frac{J_z - J_y}{6}, \tag{10}$$

where the divisor 6 comes from the cubic lattice with the macroscopic energy in the form

$$E^{\text{an}} = K_\nu \int m_\nu^2 \mathrm{d}\boldsymbol{r}, \quad \nu = x, z \tag{11}$$

Therefore, the saturation/anisotropy fields H_s and exchange/magnetic lengths for different case read

- dipolar interaction:

$$H_s = 4\pi M_S = 4\pi g \mu_B \mathcal{S}/a^3, \quad \ell_{\text{ex}}^2 = J_0 a^5/(8\pi g^2 \mu_B^2); \tag{12}$$

- single-ion anisotropy:

$$H_a = 2\mathcal{K}_\nu/M_S = K_\nu \mathcal{S}/(g\mu_B), \quad \ell_{\text{m}}^2 = a^2 J_0/K_\nu; \tag{13}$$

where index $\nu = x, z$.

Listing 5: Exchange and bulk anisotropies

```
# Value of the exchange integral
J0 = 1.0
# Single-ion anisotropy, easy-plane for Kx
# and easy-axis for Kz
Kx = -0.1
Kz = 0.2
# Exchange anisotropies
Jx = 0.01
Jy = 0.001
Jz = 0.0001
```

6.3 Surface Anisotropy

Néel surface anisotropy term is defined in the same way as bulk anisotropy described above with Hamiltionian

$$\mathcal{H}^{\text{s.-i. an. surf.}} = -\frac{\mathcal{S}^2}{2} \sum_{(\boldsymbol{l},\boldsymbol{\delta})} K_{s\boldsymbol{\delta}} (\boldsymbol{m}_{\boldsymbol{l}} \cdot \boldsymbol{u}_{\boldsymbol{l}\boldsymbol{\delta}})^2. \tag{14}$$

where index $\boldsymbol{l}$ runs over the surface sites and the unit vector $\boldsymbol{u}_{\boldsymbol{l}\boldsymbol{\delta}}$ connects the nearest neighbors of the lattice. It can be enabled setting **`surfSIA`** parameter by value `on` (default is `off`) and is added to the bulk anisotropy. Due to invariant $\boldsymbol{m}^2 = 1$ the positive and negative values of $K_{s\boldsymbol{\delta}}$ here correspond to easy-surface and easy-normal anisotropies respectively.

All surface constants should be defined (they may be different for different directions), see listing below where suffix `mx` means negative direction of x axis, `px` means positive direction of x axis etc.

Listing 6: Surface anisotropy

```
# Enable surface anisotropy evaluation
surfSIA = on
# Easy-normal anisotropy
NSAmx = -0.02
NSApx = -0.02
NSAmy = -0.02
NSApy = -0.02
NSAmz = -0.02
NSApz = -0.02
```

6.4 Coordinate-Dependent Anisotropy

Direction of the anisotropy axis (easy-axis or easy-plane) can be defined for the each lattice site separately. In this case data reads from the given file. Each line should start from # if it is commented or satisfy the following format:

```
nodeZ nodeY nodeX nZ nY nX
```

where `nodeX`, `nodeY`, `nodeZ` are integer coordinates of the site in the lattice and `nX`, `nY`, `nZ` are components of the vector defining axis direction. Given vectors will be normalized. Anisotropy constant is given by **`aniAxisCoef`**

Listing 7: Coordinate-dependent anisotropy: part of SLaSi input

```
# Enable coordinate-dependent anisotropy evaluation
# (off by default)
aniAxis = on
aniAxisFile = "../anisotropy.dat"
aniAxisCoef = 0.1
Kz = 0.
Kx = 0.
```

Listing 8: Coordinate-dependent anisotropy: part if input file anisotropy.dat

```
# this is commented line
# Z   Y   X  Nz         Ny                Nx
405   0   0  0.894427  -0.3800745755     0.23567266818
406   1   0  0.894427  -0.376527078386   0.2413127
```

```
407   2   0  0.894427  -0.3728854954    0.246833746
408   3   0  0.894427  -0.36919439      0.2501310141
409   4   0  0.894427  -0.365357284     0.25790203439
```

6.5 Dipolar Interaction

The Hamiltonian of the dipolar interaction reads

$$\mathcal{H}^{\text{dip}} = \frac{g^2\mu_B^2}{2} \sum_{n \neq m} \left[\frac{\boldsymbol{m}_n \boldsymbol{m}_m}{r_{nm}^3} - 3\frac{(\boldsymbol{m}_n \boldsymbol{r}_{nm})(\boldsymbol{m}_m \boldsymbol{r}_{nm})}{r_{nm}^5} \right]. \tag{15}$$

Dipolar interaction can be disabled by parameter **dipInteraction** for which values `on` and `off` allowed.

Note, that in absence of dipolar interaction the Bohr magneton plays a role of an effective scale of magnetic fields. So, for single-ion anisotropy unit numerical value of anisotropy field which saturates sample is obtained when **muB** is four time less than anisotropy constant.

Listing 9: Dipolar interaction

```
J0 = 1.0
# By default, dipolar interaction is enabled
# and the following line is not necessary
dipInteraction = on
muB = 0.003
```

7 External Magnetic Fields

Four types of external magnetic fields can be used in SLaSi. The main ones are constant, linearly changed and harmonically changed along x, y, z axis or in time. The last one is given by direction at the each site from file and cannot be changed with time.

Typical external field record has the following syntax:

```
<fieldDirection> = select <coord cond> <type> <params> <
   increasing time>
```

Here `<fieldDirection>` takes values **`fieldX`**, **`fieldY`** or **`fieldZ`**. Another statements describe how the field should be applied.

Statement `<coord cond>` specifies when or where field acts on the sample. For example: `select q > 0` or `select q < 0`, (here and below x, y, z or t should be used instead of `q`).

`<type>` and `<params>` describe field type: **`const`**, **`lin`** or **`harm`** and its parameters:

- **`const`**: constant field, has only one parameter: amplitude, for example: `const 0.1`.
- **`lin`**: field which is changed linearly along the given coordinate `q`. It is given as linear function $h = h_1 q + h_2$. For example: `lin 0.5 x -0.1`, where $h_1 = 0.5$, $q = x$ and $h_2 = -0.1$.
- **`harm`**: harmonically changed field in form $h = h_1 cos(\omega q + \pi\phi_0)$. For example: `harm 0.5 0.1 t 4.3`, where $h_1 = 0.5$, $\omega = 0.1$, $q = x$ and $\phi = 4.3$ (note, that phase ϕ_0 is given in units of π!).

`<increasing time>` t_0 describes how the given field should increase. SLaSi uses hat function

$$H(t, t_0) \propto \exp \frac{t^2}{t^2 - t_0^2}$$

to smooth applying of the field. SLaSi can apply till 100 external fields per axis of each type described above.

Constant external field with arbitrary spatial distribution is included by setting **`extHeff`** to state `on` (`off` by default). After that SLaSi waits path to file with field description in **`extHeffFile`** and multiplier for given vectors **`extHeffAmp`**.

Listing 10: Examples of different fields

```
# Constant field along x which starts from t > 2
# immediately with amplitude 0.1
fieldX = select t > 2.0 const 0.1 0

# Constant field along z which starts with
# beginning of simulations and is smoothed
# with t_0 = 50
fieldZ = select t > 0.0 const 0.5 50.

# Linearly increasing field applied only
# to the part of the sample
fieldY = select x > 50. lin 0.5 t 0. 0

# Harmonic sine field
fieldZ = select t > 0 harm 0.1 1 t 0.5 0.
```

Listing 11: External field from file

```
extHeff = on
extHeffFile = "../myField.dat"
extHeffAmp = 0.0008
```

Listing 12: myField.dat structure

```
# All lattice sites should be enumerated here
# Z   Y   X  Nz        Ny        Nx
  0   0   0  0.1       0.1       0.1
  0   0   1  0.1       0.1       0.11
  0   1   0  0.1       0.1       0.111
  0   1   1  0.1       0.1       0.1111
  1   0   0  0.1       0.1       0.11111
# ...
```

8 Initial Magnetization Distribution

9 Integration Parameters

10 Miscellaneous

11 Examples

11.1 Uniformly Magnetized Sphere

11.2 Walker's limit

Let us consider a motion of 180° domain wall in a nanowire. We take an 1-dimensional sample of 1000 sites along z axis:

```
geometry = rectangle
a = 1
b = 1
h = 1000
```

There are two anisotropies: easy-axial along z and easy-plane in $y - z$:

```
dipInteraction = off
muB = 0.01
Kz = 0.04
Kx = -0.01
```

Here the dipolar interaction is turned off for simplicity and Bohr magneton is taken 4 times smaller than **Kz** to have field scaled by anisotropy field, see Eq. (13). The domain wall shifted to the 100-th node is relaxed previously. So, now we can apply fields of different static fields along z axis and shift domain wall. Its motion is highly dependent on applied field amplitude. Below the Walker critical field

$$H_W = \eta \frac{2\mathcal{K}_z}{M_S} \left(\sqrt{1 + \frac{1}{Q}} - 1 \right) \sqrt[4]{1 + \frac{1}{Q}}, \quad Q = \frac{\mathcal{K}_z}{|\mathcal{K}_x|} \tag{16}$$

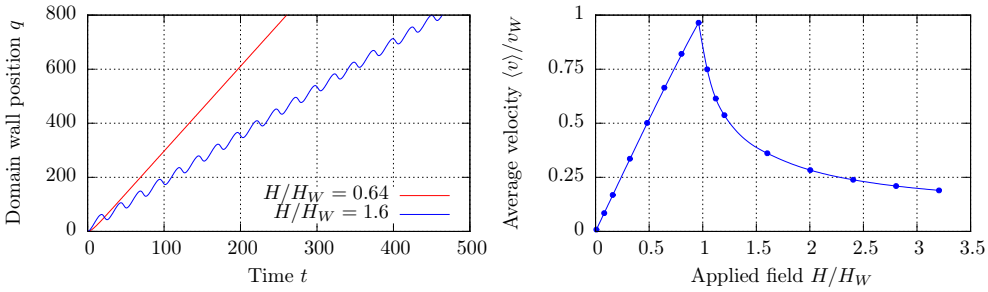
the domain wall moves with constant velocity. H_W corresponds to the maximal velocity

$$v_W = \frac{2\gamma}{M_S} \sqrt{A|\mathcal{K}_x|Q} \left(\sqrt{1 + \frac{1}{Q}} - 1 \right). \tag{17}$$

Above H_W domain wall motion becomes nonlinear, see Fig. 1.

Listing 13: Walker's limit

```
#
# Geometry
#
geometry = rectangle
a = 1
b = 1
h = 1000
```



(a) Domain wall position below and above H_W with coordinates measured in a nanowire.
in lattice constants.

(b) Average velocity of the domain wall

Figure 1: Motion of a domain wall in a nanowire below and above Walker's field.

```
#
################################################
# Magnetic parameters
#
damping = 0.1
J0 = 1.0
dipInteraction = off
muB = 0.01
Kz = 0.04
Kx = -0.01
#
################################################
# Initial distribution
#
init = file
initFile = "../relax/line.sls0050"
#
################################################
# Magnetic field
fieldZ = select t > -1.0 const 0.012 0
#
################################################
# Integration
#
timeStep = 400.0
snapshotCount = 4000
```

```
stopdMdt = -1
volumeModel = 3D
tolerance = 1e-2
intStepCount = 500
scanx1 = 0
scany1 = 0
scanz1 = 150
```