

μ -PRO[®] MANUAL

Ferroelectric module

Xiaoxing Cheng
Bo Wang

November 25, 2019

Contents

1 Installation and setup	4
1.1 Prerequisite	4
1.2 Installation	4
1.3 Activate your usage	4
2 μ-Ferro: an introduction	6
2.1 μ -Pro [®] and μ -Ferro	6
2.2 A quick tutorial	6
2.3 Structure of μ -Ferro package	8
3 Files of μ-Ferro	10
3.1 Input files	10
3.1.1 input.in	10
3.1.2 pot.in	11
3.1.3 Polar.in	12
3.1.4 OctaTilt.in	12
3.1.5 AFMtip.in	12
3.2 Output files	13
3.3 Pre-processing files	14
3.3.1 PBS file for submitting tasks	14
3.3.2 scripts for batch tasks	14
3.4 Post-processing files	15
3.4.1 GNUPlot & bash scripts	15
4 The input.in File	17
4.1 List of most important parameters	17
4.2 CDER	18
4.3 CELASBC	19
4.4 CELECBC	19
4.5 CPOLARBC	19
4.6 DIELECON	20
4.7 ELECFIELD	20
4.8 FILMTHICK	20
4.9 FLEXOCON	20
4.10FRACTION	21
4.11GRADPCON	21
4.12GRADQCON	21
4.13LAFMTIP	22
4.14LBULK	22
4.15LELAS	22
4.16LELEC	22
4.17LFLEXO	22
4.18LINHOM	23

4.19 LOCTILT	23
4.20 LOUTELAS	23
4.21 LOUTELEC	23
4.22 LOUTFLEX	24
4.23 LOUTFORC	24
4.24 LOUTGRAD	24
4.25 LOUTLAND	24
4.26 LPNOISE	24
4.27 LQNOISE	24
4.28 MISFIT	25
4.29 PNOISEED	25
4.30 PNOISMAG	25
4.31 QNOISEED	25
4.32 QNOISMAG	25
4.33 REALDIM	26
4.34 ROTANGLE	26
4.35 SCREENBOT	26
4.36 SCREENTOP	26
4.37 STRAIN	26
4.38 STRESS	27
4.39 SUBTHICK	27
4.40 SYSDIM	27
4.41 TDELTA	27
4.42 TEM	28
4.43 TIPGAMMA	28
4.44 TIPRAD	28
4.45 TOUTPUT	28
4.46 TSTART	28
4.47 TTOTAL	29
5 Examples	30
5.1 Equilibrium domain structure of BaTiO ₃ single crystal	30
6 Recommended references	35
6.1 Potential related	35
6.2 Elastic related	36
6.3 Electric related	36
6.4 Tip related	36

Preface

This manual is for the μ -Ferro module of phase field package μ -Pro[®]. μ -Ferro is a phase-field method based program to simulate the electric polarization evolution in ferroelectric materials under external fields. Users are recommended to have basic knowledge about ferroelectric materials, programming experiences are not necessary. However, basic knowledge of LINUX operations are necessary, as currently only the linux executable is released.

This manual starts with a brief introduction and a simple example to get the reader started. Followed by a detailed introduction of different input files, output data and pre/post processing scripts. Next, all of the parameters that can be set using the input.in file are listed and explained in detail.

The manual is mainly a reference guide and explains most files and control flags implemented in the code. A complete description of the underlying algorithms and theories can be found in the recommended reference section [6](#).

Chapter 1

Installation and setup

1.1 Prerequisite

μ -Ferro is built with Intel parallel studio cluster edition, and Intel mpi is required to run the program. Make sure the `$LD_LIBRARY_PATH` is correctly configured that the folder containing shared library `libmpi.so.12` and `libmpifort.so.12` must be presented. These two library should be found in the Intel mpi distribution folder, for example, on the Penn State ACI-ICS server, it is located at `/opt/intel/compilers_and_libraries_2016.3.210/linux/mpi/intel64/lib`. Another dependency is the mpi version of fftw library, we need the `libfftw3_mpi.so.3` and `libfftw3.so.3` shared library. Make sure the fftw library is compiled with intel compiler and intel mpi instead of gcc compiler.

To add an additional path to `$LD_LIBRARY_PATH`, you would use the `export` command.

```
1 export LD_LIBRARY_PATH=/YOUR/OWN/PATH:$LD_LIBRARY_PATH
```

1.2 Installation

To install and use the program is simple, all you need to do is copy the `Ferroelectric.exe` file to your server and change the permission of the file to be executable. Notice here the file extension `exe` does not mean it is a Window's program. The `Ferroelectric.exe` is a linux executable, the extension is to differentiate it from other text files.

1.3 Activate your usage

In order to register your copy of the program, we need the following information:

1. **hostname**. Be careful that this may not be the host name you used to get access to your linux server, but the host name of the login node. For example, I can connect to the Penn State server through `ssh xuc116@aci-b.aci.ics.psu.edu`, but there are multiple login nodes, I'm actually connected to a node called `aci-004.aci.production.int.aci.ics.psu.edu`. This is the host name you should provide to us, rather than the `aci-b.aci.ics.psu.edu` one. If there are multiple host names, a list should be provided. You can easily obtain the hostname of your server or computer by execute `echo $HOSTNAME` in the linux terminal.
2. **username**. The user name you want to grant access to use μ -Ferro. You can find the user name by typing `echo $USER` in the terminal. Note if you apply for the group license rather than individual one, you should provide the group name instead of user name.

3. **group name.** The group of users that you want to grant access to use μ -Ferro. You can find all of the group you belong to by executing the **id** command in terminal.
4. **ip address.** You can obtain your server's ip address by typing **curl ipinfo.io/ip** in terminal. Same as hostname, if there are more than one ip address that you may get connected to, provide us with a list of them.

Chapter 2

μ -Ferro: an introduction

2.1 μ -Pro[®] and μ -Ferro

μ -Pro[®] represents Microstructure-Property, which is a bundle of several phase-field based microstructure modeling programs, ranging from alloy solidification and precipitation to ferroelectric and ferromagnetic domain evolution. Different from most simulation programs, all of the programs in μ -Pro[®] package is solved through spectral method rather than finite difference or finite element methods. MPI and FFTW3 boost our code efficiency, and empowers us to perform 3D simulations at relatively low cost.

μ -Ferro is one of the modules in μ -Pro[®] package, which can evolve ferroelectric domains structures by solving the time dependent Ginzburg Landau equation as well as taking elastic and electric contribution into consideration. The equation is solved using a semi-implicit spectral scheme ?? in a parallel fashion. This ensures the coupled equations to be solved accurately and efficiently.

μ -Ferro simulates bulk single crystals and thin films clamped on substrates, to study the electric polarization response to external mechanical and electrical stimuli. Some interesting features includes the ability to different film orientations into account, inhomogeneous structure (such as superlattices and islands), flexoelectric effect, dislocations, charged defects and secondary order parameters (such as oxygen-octahedral rotation and tilt).

2.2 A quick tutorial

The files needed for this tutorial can be found in the μ -Ferro example folder (usually located at `mupro/install/path/MUPRO_version_number/Ferroelectric/Example`).

1. Copy the following three files from the example folder to your work directory.

input.in This file is the main input file of μ -Ferro. Tags in each line gives predefined keywords which tells the program what to perform. It is followed by an equation sign '=' and one or several values associated with the keyword. For each keyword, a default will be used if it is not been specified in the file.

Polar.in This file describes the initial polarization distribution of the simulated system. The program still works even if without **Polar.in**, and it will use the default random noise as initial structure. This file contains the polarization at each grid points in a columnar fashion. An example of the file is shown in Table 2.1. The first line specify the x, y and z ranges of the simulation system. Starting from the second line, the first three columns represents the grid position. The 3rd to 6th columns represents the polarization components projected to the x, y, and z axis of the Cartesian coordinate for calculation (global coordinate), Px, Py, Pz. The 7th to 9th

Table 2.1: Example of Polar.in file

10	10	10							
1	1	1	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	1	2	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	1	3	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
...									
1	1	10	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	2	1	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	2	2	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
...									
1	10	10	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	10	10	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	10	10	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
...									
10	10	10	0.00E+00	0.00E	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

columns represents the polarization components projected to the 1, 2, and 3 axis of the crystallographic coordinates (local coordinate).

pot.in This file stores the necessary thermodynamic parameters and other physical properties for the ferroelectric material you want to simulate, such as of Landau potential coefficients, elastic constants, and electrostrictive coefficients. Note that for the same material, a few different versions of thermodynamic parameters may exist. For this quick tutorial, *BaTiO₃* from reference ?? is used. The content of this pot.in is shown in Listing ??.

2. Run the μ -Ferro program.

If you have successfully installed the package through the **install.sh** we provided, then you should be able to run it just by typing **mupro-Ferroelectric** in terminal. Otherwise, you need to find the **Ferroelectric.exe** located in the Ferroelectric folder of our μ -Ferro distribution and then either execute it by give the whole path to the binary, or copy it to your current working directory. You need to make sure you have all the three input files copied to the current directory where you're trying to run the μ -Ferro program.

After the program started, some information will be printed to the screen as shown in Figure 2.1.

After the program finished, you will get some output data files, all of which are in columnar pattern similar to the **Polar.in** file, that first three column define the position for data point, and then rest of the columns are data at each grid point. You may process the data or visualize it use the tools you're most familiar with.

We have provided several examples in our μ -Ferro distributions including this quick tutorial files, but in the future when you are supposed to prepare these files yourself. It may be a daunting task for beginners, that's why we have this manual and some other tools to help you out.

For the creating the **input.in** file, you should refer to the Chapter 4 and find the keyword you need for your specific case. For the **Polar.in** file, we prepared a small program called "Structure-Generator" that will help you create some commonly used or geometrically simple domain structures. And lastly, for the **pot.in** file, we are building an online database for phenomenological thermodynamic parameters, in the future you may want to download potential files from the website directly. For now, we put the thermodynamic parameters for some common materials in the "potentials" folder in the ferroelectric directory of μ -Ferro distribution. But of course, you can use your own parameters for the potential file. The **pot.in** file is in the JSON format, you are allowed to change the value (number or string after the colon), but not the keys (the string before the colon, which is the name for the parameter we used in our


```

-----
Valid to use, thanks for using MUPRO
-----
*****Ferroelectric main program*****
*****using*****
*****mu-pro package*****
-----
-----Simulation system general setup-----
-----Input parameters-----
Temperature (K) = 298.0000
Real dimension lx (nm) = 8.0000
Real dimension ly (nm) = 8.0000
Real dimension lz (nm) = 8.0000
Total simulation timesteps = 101
Initial simulation timesteps = 0
Output for every # timesteps = 50
dt for TDGL = 0.010000
Consider elastic effect = T
Elastic BC is 2
Use bulk total stress as elastic BC-----
bulk total stress sxx = 0.0000
bulk total stress syy = 0.0000
bulk total stress szz = 0.0000
bulk total stress syz = 0.0000
bulk total stress sxz = 0.0000
bulk total stress sxy = 0.0000
Add noise to polarization = T
Consider inhomogeneous system= T
Consider electric effect : T
Electric BC is 2
Export stress/strain = T
-----Input done-----
-----Finished reading
Broadcasting parameters to all cores-----
-----Finished broadcasting
-----

```

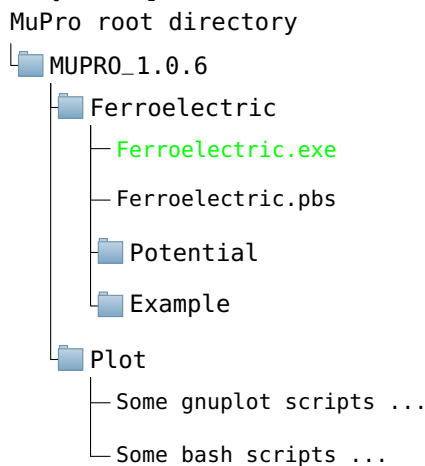
Figure 2.1: Example screen output

simulation). Once the keys are changed, we can no longer find all of the value needed for simulation, and the program may crash.

The post processing can also be quite challenging sometimes. We also prepare some gnuplot scripts, as well as a visualization GUI based on VTK+Qt to help you get started with the process. When you have more sophisticated visualization needs, you may find the right one that suits you the most.

2.3 Structure of μ -Ferro package

μ -Ferro is a Fortran 90 program. It allows for dynamic memory allocation and a single executable which can be used for any type of calculation. Generally the executables and the thermodynamic potentials should reside in the following directories:



The μ -Pro[®] root directory contains the license.lic file and the μ -Pro[®] distribution of specific version, in this example is **MUPRO_1.0.6**. In the **MUPRO_1.0.6** folder there are

three directories,

Ferroelectric contains the main program for μ -Ferro phase-field simulation, an example PBS script for you to submit jobs on systems using PBS job queue system, a **Potentials** folder which contains several commonly used thermodynamic potentials and physical properties that the program needs, and an **Example** folder that contains some example projects for you to try out and gain experience with using the program.

Plot contains three pairs of gnuplot script and bash script, one for 1D plot, one for 2D scalar heat plot, and the third for 2D vector glyph plot.

Chapter 3

Files of μ -Ferro

3.1 Input files

The following input files need to be presented in your working directory.

1. input.in
2. pot.in
3. Polar.in

Some other optional input files are

- OctaTilt.in
- AFMtip.in

3.1.1 input.in

input.in file is in a free-format structure, each line of the file is made up of several parts

identifier The string that appears before "=" in each line. Identifiers are keywords that μ -Ferro used to link the corresponding parameters in the program with values that are set in the input file. According to the values to be set, the identifiers can be classified into four types: 1. logical switch 2. choice options 3. single value 4. array values.

values The values that appears after "=" and before "#" in each line. These are numbers, booleans or string values related to the identifier.

comments The string after "#".

Let's take a look at some examples of the four types of identifiers and their values.

logical switch All logical switch type of identifiers start with "L" representing logical type, and the acceptable values are either True or False.

Example: LELAST = True # Consider the elastic effect on ferroelectric system

choice options All choice option type of identifiers start with "C" representing choice, and the acceptable values are integers.

Example: CELASBC = 2 # Set the elastic boundary condition to be stress free for bulk simulations.

single value This type of identifiers only accept one value, and it can be either real or integer number depending on the specific identifier.

Example: TIMESTEP = 10000 # Set the maximum simulation steps to be 10000.

array value This type of identifiers accept more than one values, and it can be either real or integer number depending on the specific identifier.

Example: SYSDIM = 128 128 40 # Set the simulation system size in x, y, z direction

Note that although identifiers in input.in are designed to be case insensitive, it is highly recommended to use all capitalized letters for IDENTIFIERS for clarity.

3.1.2 pot.in

Like most materials simulation methods, phase field simulation for ferroelectrics require an potential, i.e., the Landau potentials of ferroelectric materials. The standard potential files are provided in the potentials folder of μ -Ferro distribution. If missing, one can easily obtained the potentials files from online database.

```
1 {
2   "name"      : "BiTiO3",
3   "Comment"   : "Fitted by Jian Jun Wang, the landau coefficient needs
4   to be updated according to stress for every time step",
5   "Reference" : "JAP 108. 114105 (2010)",
6   "CurieT"   : 390,
7   "a0"        : 42769496,
8   "p0"        : 0.26,
9   "Landau"    :
10    { "a1"      : "5*10^5*160*(cosh(160/tem)/sinh(160/tem)-cosh(160/390)/
11    sinh(160/390))",
12    "a11"     : -1.154E8,
13    "a12"     : 6.530E8,
14    "a111"    : -2.105E9,
15    "a112"    : 4.091E9,
16    "a123"    : -6.688E9,
17    "a1111"   : 7.590E10,
18    "a1112"   : -2.193E10,
19    "a1122"   : -2.221E10,
20    "a1123"   : 2.416E10
21  },
22  "Electrostrictive" :
23  {
24    "Q11" : 0.11 ,
25    "Q12" : -0.045 ,
26    "Q44" : 0.029
27  },
28  "ElasticModulus" :
29  {
30    "C11" : 1.7794E11,
31    "C12" : 0.9635E11,
32    "C44" : 1.2199E11
33  },
34  "ElasticCompliance" :
35  {
36    "S11" : 9.07E-12,
37    "S12" : -3.186E-12,
```

```

36         "S44"   : 8.197E-12
37     },
38     "Gradient" :
39     {
40         "G11"   : 5.10E-10,
41         "G12"   : 0.0,
42         "G44"   : 1.0E-11,
43         "GM44"  : 1.0E-11
44     }
45 }

```

3.1.3 Polar.in

Polar.in is also a mandatory input file (or μ -Ferro will start from a random noise of polarization). It stores the initial spatial distribution of polarization for calculation. It has a header line, specifying the total grid size of the system. The rest of the lines are made up of 9 columns, first three are data point positions, 4- to 6-th columns are polarization under global coordinate (the lab coordinate), 7- to 9-th columns are polarization in local coordinate (the crystal lattice coordinate). Be careful that the sequence of data point is z,y,x rather than x,y,z, that is to say, the data position gets update along z direction first, then y direction and in the last x direction.

Table 3.1: Example of Polar.in file

```

10 10 10
1 1 1 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
1 1 2 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
1 1 3 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
...
1 1 10 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
1 2 1 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
1 2 2 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
...
1 10 10 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
2 10 10 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
3 10 10 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00
...
10 10 10 0.00E+00 0.00E+00 0.26E+00 0.00E+00 0.00E+00 0.26E+00

```

3.1.4 OctaTilt.in

This file is similar to Polar.in, but stores the oxygen octahedral rotation around the three crystal axis. Please read this reference [1] for more information about how we consider oxygen octahedral rotation in phase-field simulation.

3.1.5 AFMtip.in

This file is used to set the elastic/electric boundary conditions of ferroelectric films by an AFM tip as a mechanical indenter or as a bias electrode. The first line of this file contains three values, the first one is the times of changes of AFMP tip for the whole simulation, the second value is the choice of electric boundary condition, and the third value is the choice of elastic boundary condition, the meaning of different value for boundary condition can be found the in Chapter 4 for identifier of **CTIPFORC** and **CTIPBIAS**.

The rest of **AFMtip.in** file are made up of 7 columns,

timestep timestep that the tip parameters get changed.

ntipx grid position of the tip in x direction.

ntipy grid position of the tip in y direction.

bias the electric potential applied to the tip.

force the force applied by the tip to the film.

fricx friction factor in x direction.

fricy friction factor in y direction.

As an example, to simulate a moving AFM tip with 1V bias which writes along the path of the letter "P", one can prepare the AFMtip.in as

```

1      25      1      0
2 # timestep ntipx ntipy bias force fricx fricy
3      1 30.00 90.00 1.00 0.00 0.0 0.0
4     2001 30.00 84.00 1.00 0.00 0.0 0.0
5     4001 30.00 78.00 1.00 0.00 0.0 0.0
6     6001 30.00 72.00 1.00 0.00 0.0 0.0
7     8001 30.00 66.00 1.00 0.00 0.0 0.0
8    10001 30.00 60.00 1.00 0.00 0.0 0.0
9    12001 30.00 54.00 1.00 0.00 0.0 0.0
10   14001 30.00 48.00 1.00 0.00 0.0 0.0
11   16001 30.00 42.00 1.00 0.00 0.0 0.0
12   18001 30.00 36.00 1.00 0.00 0.0 0.0
13   20001 30.00 90.00 1.00 0.00 0.0 0.0
14   22001 36.00 90.00 1.00 0.00 0.0 0.0
15   24001 42.00 90.00 1.00 0.00 0.0 0.0
16   26001 48.00 90.00 1.00 0.00 0.0 0.0
17   28001 54.00 90.00 1.00 0.00 0.0 0.0
18   30001 60.00 90.00 1.00 0.00 0.0 0.0
19   32001 60.00 84.00 1.00 0.00 0.0 0.0
20   34001 60.00 78.00 1.00 0.00 0.0 0.0
21   36001 60.00 72.00 1.00 0.00 0.0 0.0
22   38001 60.00 66.00 1.00 0.00 0.0 0.0
23   40001 60.00 60.00 1.00 0.00 0.0 0.0
24   42001 54.00 60.00 1.00 0.00 0.0 0.0
25   44001 48.00 60.00 1.00 0.00 0.0 0.0
26   46001 42.00 60.00 1.00 0.00 0.0 0.0
27   48001 36.00 60.00 1.00 0.00 0.0 0.0

```

Listing 3.1: PBS script

Notice the potential and force applied by tip is actually a distribution rather than uniform value. The value specified here is just one parameter used to get the distribution, you should always check the electric potential distribution and stress distribution to confirm that you get what you're expecting.

3.2 Output files

All output files generated by μ -Ferro are listed in Table 3.2. They can be controlled by corresponding tags. Most of the output data files are time dependent, that is, the spatial distribution of some value at a given timestep. The file name for these time dependent data have one thing in common, they are all made up of three parts separated by ".", the first one is the file name, such as "Polar", "Strain", the second one is the time step this data get output, which has a length of 8 digits, filled with 0 if the number length is less than 8, the third one is the file extension "dat". All possible data files are listed and explained in the Table 3.2.

Table 3.2: Output file list

File name	control	Columns	Explanation
Polar	-	6	Polarization in global coordinate and local coordinate
Grade_En	LOUTGRAD	1	Gradient energy distribution
Grad_For	LOUTGRAD	3	Gradient driving force for polarization x, y, z
Eigen_Str	LOUTELAS	6	Eigen strain distribution in voigt notation
Strain	LOUTELAS	6	Total strain distribution in voigt notation
Elast_Str	LOUTELAS	6	Elastic strain distribution in voigt notation
Stress	LOUTELAS	6	Stress distribution in voigt notation
Displace	LOUTELAS	3	Displacement x, y, z
Elast_En	LOUTELAS	1	Elastic energy
Elas_For	LOUTELAS	3	Elastic driving force for polarization x, y, z
Elect_En	LOUTELEC	1	Electric energy
Elefield	LOUTELEC	3	Electric field x, y, z
Elec_Phi	LOUTELEC	1	Electric potential
Elec_For	LOUTELEC	3	Electric driving force x, y, z
Charges	LOUTELEC	3	Bound charge, free charge, total charge
LandP_En	LOUTLAND	1	Landau energy for polarization as order parameter
LandPFor	LOUTLAND	3	Landau driving force for polarization x, y, z
Eflexo	LOUTFLEX	3	Flexoelectric driving force, as known as the effective electric field, x, y, z
Gen_ForX	LOUTFORC	6	Elastic, electric, flexoelectric, landau, gradient and total driving force in x direction
Gen_ForY	LOUTFORC	6	Elastic, electric, flexoelectric, landau, gradient and total driving force in y direction
Gen_ForZ	LOUTFORC	6	Elastic, electric, flexoelectric, landau, gradient and total driving force in z direction
OctaTilt	-	6	Oxygen octahedral tilt around x, y, z axis
ElasQFor	LOUTELAS	3	Elastic driving force for oxygen octahedral tilt
GradQ_En	LOUTGRAD	1	Gradient energy of oxygen octahedral tilt
GradQFor	LOUTGRAD	3	Gradient driving force for oxygen octahedral tilt x, y, z component
LandQ_En	LOUTLAND	1	Landau energy of oxygen octahedral tilt as order parameter
LandQFor	LOUTLAND	3	Landau driving force for oxygen octahedral tilt
GenQForX	LOUTFORC	4	Elastic, landau, gradient and total driving force for oxygen octahedral tilt in x axis
GenQForY	LOUTFORC	4	Elastic, landau, gradient and total driving force for oxygen octahedral tilt in y axis
GenQForZ	LOUTFORC	4	Elastic, landau, gradient and total driving force for oxygen octahedral tilt in z axis

3.3 Pre-processing files

3.3.1 PBS file for submitting tasks

Many of the linux high performance computing system are using "Portable Batch System (PBS)" to manage the queueing and allocation of jobs on the cluster. For these types of systems, a PBS script is needed for job submission.

```

1 #PBS -l nodes=1:ppn=2
2 #PBS -l walltime=24:00:00
3 #PBS -N Ferroelectric
4 #PBS -j oe
5 #PBS -A allocation_to_use
6 #PBS -M myemail@school.edu
7 #PBS -m ae
8
9 cd $PBS_O_WORKDIR
10 echo `date`
11 mpirun MUPRO -ferroelectric.exe
12 echo `date`

```

Listing 3.2: PBS script

3.3.2 scripts for batch tasks

From time to time you may want to run a series of jobs to see the trend of how things change. The following script gives an example that replace custom keyword in the input.in file, **nx_size** and **lx_size**, with different numbers ranging from 40 to 120, before submitting the job with PBS command. Note that the keyword used here for batch substitution should not be the same as the identifiers (left side of "=") specified in input.in.

```

1 #!/bin/bash
2
3 length=( 40 60 80 100 120 )
4
5 for interval in "${length[@]}"
6 do
7     let nx=$interval
8     let lx=($nx)
9     mkdir $interval
10    cd $interval
11    cp ../input.in .
12    sed -i "s/nx_size/${nx}/g" input.in
13    sed -i "s/lx_size/${lx}/g" input.in
14    qsub ferro.pbs
15    cd ..
16 done

```

Listing 3.3: Batch submission of jobs

3.4 Post-processing files

3.4.1 GNUPlot & bash scripts

In the **Plot** folder we provide several script that enables you to quickly draw some 1D plot, 2D scalar plot and 2D vector plot for the output data μ -Ferro program generated.

plot_1d.sh

This script helps you to draw 1D profile of a one column vs another in the data file. It takes at least 9 arguments: 1. minimum value in x direction, 2. maximum value in x direction, 3. minimum value in y direction, 4. maximum value in y direction, 5. minimum value in z direction, 6. maximum value in z direction, 7. column number of the data used for the horizontal axis of the 1D plot, 8. column number of the data used for the vertical axis of the 1D plot, 9. name of the data file.

You can actually pass more data file names as additional arguments, and the script will draw the same 1D profile for all of the data files on one figure.

An example usage of the script is shown in Listing [3.4](#). It means slice the **Polar.00001000.dat** file into one line, $1 \leq x \leq 128$, and $y = 40, z = 20$, then plot the figure using the first column and the fourth column of the data file, which correspond to grid position in x direction and the x component of polarization in global coordinates. The output figure has a default name of **fig1d.png**.

```

1 plot_1d.sh 1 128 40 40 20 20 1 4 Polar.00001000.dat

```

Listing 3.4: Example of 1D plot usage.

plot_2ds.sh

This script helps you to draw 2D heat plot for one column in the data file. It takes at 7 arguments: 1. minimum value in x direction, 2. maximum value in x direction, 3. minimum value in y direction, 4. maximum value in y direction, 5. minimum value in z direction, 6. maximum value in z direction, 7. column number of the data you want to use for the heat plot.

An example usage of the script is shown in Listing [3.5](#). It means slice the **Polar.00001000.dat** file into one plane, $1 \leq x \leq 128, 1 \leq y \leq 128$, and $z = 20$, then plot the figure using the fourth column of the data file, which correspond to the x component of polarization in global coordinates. The output figure has a default name of **fig2ds_Polar.00001000.png**.


```
1 plot_2ds.sh 1 128 1 128 20 20 4 Polar.00001000.dat
```

Listing 3.5: Example of 2D scalar plot usage.

plot_2dv.sh

This script helps you to draw 2D vector plot for two column in the data file. The 3D vectors are projected to the 2D slice you specify. It takes at least 8 arguments: 1. minimum value in x direction, 2. maximum value in x direction, 3. minimum value in y direction, 4. maximum value in y direction, 5. minimum value in z direction, 6. maximum value in z direction, 7. choice between global and local coordinates (1-global, 2-local) of the vector to be drawn, 8. column number according to which the vectors are colored.

An example usage of the script is shown in Listing [3.6](#). It means slice the **Polar.00001000.dat** file into one plane, $1 \leq x \leq 128, 1 \leq y \leq 128$, and $z = 20$, and global coordinates, then plot the figure using the fourth and fifth columns of the data file, which correspond to the x and y component of polarization in global coordinates, px is used as the x component of the 2D vector and py as the y component. And the vector will be colored according to the 5-th column, which is the y-components of vectors. The output figure has a default name of **fig2dv_Polar.00001000.png**.

```
1 plot_2dv.sh 1 128 1 128 20 20 1 4 Polar.00001000.dat
```

Listing 3.6: Example of 2D vector plot usage.

Chapter 4

The input.in File

4.1 List of most important parameters

Table 4.1: Most important tags used for μ -Ferro

	Tags	Class	Default	Description
System	SYSDIM	Vector	10, 10, 10	Grid points in x, y, z
	REALDIM	Vector	Same as SYSDIM	Real length along x, y, z
	SUBTHICK	Scalar	0	Grid points for substrate thickness
	FILMTHICK	Vector	0	Grid points for film thickness
	ROTANGLE	Vector	0.0, 0.0, 0.0	Euler rotation angle around z, x', z''
Time	TTOTAL	Scalar	1000	Total timesteps
	TSTART	Scalar	0	Initial timestep
	TOUT	Scalar	200	Time interval for output
	TDELTA	Scalar	0.01	Delta t of TDGL
Solvers	LINHOM	Flag	FALSE	Flag for inhomogeneous solver
	CDER	Choice	0- step-corrected FFT 1- FDM 2- normal FFT	Choices of derivative solver
Thermal	TEM	Scalar	298.0	Temperature in Kelvin
Polarization	CPOLARBC	Choice	0- natural BC 1- free BC 2- blocked BC 3- small extrapolation length 4- zero bound charge 5- large extrapolation length	Choices of boundary condition, for Px, Py, Pz
	ETRLP	Vector	0.0, 0.0, 0.0	Extrapolation length of Px, Py, Pz in nm
	LPNOISE	Flag	FALSE	Flag of polarization noise
	PNOISMAG	Scalar	0.1	Magnitude of polarization noise
	PNOISEED	Scalar	10	Seed of polarization noise
Elastic	LELAS	Flag	TRUE	Flag of elastic energy
	CELASBC	Choice	0- film 1- bulk strain 2- bulk stress	Choices of elastic boundary conditions
	MISFIT	Vector	0.0, 0.0, 0.0	Misfit strain exx, eyy, exy

	STRAIN	Vector	0.0, 0.0, 0.0, 0.0, 0.0, 0.0	Strain BC for bulk, exx, eyy, ezz, eyz, ezx, exy
	STRESS	Vector	0.0, 0.0, 0.0, 0.0, 0.0, 0.0	Stress BC for bulk, sxx, syy, szz, syz, szx, sxy
	TIPRAD	Scalar	50.0	Tip radius in nm
	LELEC	Flag	TRUE	Flag of electric energy
Electric	CELECBC	Choice	1- open circuit 2- short circuit 3- top open bottom short 4- top short bottom open 5- bulk	Choices of electric boundary conditions
	TIPGAMMA	Scalar	10.0	Lortenz <i>gamma</i> for tip in nm
	SCRBOT	Scalar	0.0	Screening factor of bottom electrode
	SCRTOP	Scalar	0.0	Screening factor of top electrode
	Gradient	GRADCON	Vector	0.6, 0.0, 0.3
Flexoelectric	LFLEXO	Flag	FALSE	Flag of flexoelectric energy
	FLEXOCON	Vector	5.1, 3.3, 0.045	Flexoelectric tensor component f11, f12, and f44
Oxygen Octahedral Rotation	LOCTILT	Flag	FALSE	Flag of oxygen octahedral tilt
	GRADQCON	Vector	1.0 0.0 0.5	Gradient energy tensor of OOT component v11, v12, v44
	LQNOISE	Flag	FALSE	Flag of OOT noise
	QNOISMAG	Scalar	0.1	Magnitude of OOT noise
	QNOISEED	Scalar	10	Seed of OOT noise
Outputs	LOUTLAND	Flag	FALSE	Flag of output Landau energy of polarization
	LOUTELAS	Flag	FALSE	Flag of output elastic energy
	LOUTELEC	Flag	FALSE	Flag of output elastic energy
	LOUTGRAD	Flag	FALSE	Flag of output gradient energy
	LOUTFLEX	Flag	FALSE	Flag of output flexoelectric fields
	LOUTFORC	Flag	FALSE	Flag of output driving forces

4.2 CDER

Full name: choice derivative

Type: choice

Options: **0** FFT method with step correction
1 finite difference method
2 FFT method without step correction

Description: Choose the numerical methods for taking derivatives. The places may be affected by this option include bound charges (divergence of polarization), electric field (gradient of electric potential), and flexoelectric field/flexoelectric driving force(derivatives of stress tensors)

Notes: Do not change CDER unless following situations:

1. Flexofield calculations

4.3 CELASBC

Full name: choice elastic boundary conditions

Type: Choice

- Options:**
- 0** film boundary condition, i.e., stress-free at film surface and clamped at film bottom. (for detailed treatment please refer to micromechanics [Li 2002][Khachaturyan Chap 8?])
 - 1** strained boundary condition for bulk, i.e., clamped boundary condition. Assign a fixed strain tensor as average strain value of bulk system. Use this combined with **STRAIN**-tag [Ref].
 - 2** stressed boundary condition for bulk. Assign a fixed stress tensor as average stress value of bulk system. Use this combined with **STRESS**-tag [Ref].

Description: Choose the elastic boundary condition when solving for mechanical equilibrium.

Notes:

- 1. CELASBC = 1 and 2 only work for bulk system while CELASBC = 0 only works for film system. Please be consistent with the **LBULK** tag.
- 2. CELASBC = 1 is related to **STRAIN**
CELASBC = 2 is related to **STRESS**
CELASBC = 0 is related to **MISFIT**

4.4 CELECBC

Full name: Choice electric boundary conditions

Type: Choice

- Options:**
- 0** bulk boundary condition, i.e., given a fixed electric field to bulk system. Specify applied electric field by **ELECFIELD**.
 - 1** open-circuit boundary conditions at film surface and bottom.
 - 2** short-circuit boundary condition at film surface and bottom.
 - 3** open-circuit boundary condition at film surface while short-circuit boundary condition at film bottom.
 - 4** open-circuit boundary condition at film bottom while short-circuit boundary condition at film surface.

Description: Choose the electric boundary condition of electrostatic equation

Notes:

- 1. CELECBC = 0 only works for bulk systems while other options only work for filmsystems.
- 2. CELECBC = 0 is paired with **ELECFIELD**.
- 3. In experiment, even without electrode, there are still plenty of mechanisms that can compensate charges at the interface or surface, so be careful when to use the open circuit condition.

4.5 CPOLARBC

Full name: Choice polarization boundary conditions

Type: Choice

- Options:**
- 0** natural boundary condition
 - 1** Neumann boundary condition, $\frac{dP}{dz} = 0$ at film surface and bottom.
 - 2** Dirichlet boundary condition, $P = 0$ at film surface and bottom.

Description: Choose the boundary condition for polarization when evolving the time dependent Ginzburg Landau equation. The polarization boundary condition is related to extrapolation length in journal articles.

Notes: The boundary condition for polarization is a value that is difficult to determine, for normal calculation that focus on domain structures, you can go with the default Neumann boundary condition, which set derivative to zero. Only when you're simulating some special distribution near the interface or surface, you may want to try the other two boundary conditions.

4.6 DIELECON

Full name: Dielectric constants (background)

Type: Vector, Real

Description: To set background dielectric constant which will override the dielectric constants assigned in the pot.in file.

Notes:

1. The background dielectric constant denotes the dielectric response to applied electric field apart from the spontaneous polarization. It also represents the dielectric constant at infinite high temperature in the plot of dielectric permittivity as a function of temperature. For detailed description, please refer to [Ref. Tagantsev, Zheng & Woo]

4.7 ELECFIELD

Full name: Electric Field

Type: Vector, Real

Description: This tag sets up the electric boundary condition of bulk system using an applied electric field. It can be used to obtain the PE loop of a bulk system. Three real numbers must be specified for this tag, electric field along x, y and z direction, separated by space.

Notes:

1. The unit of assigned electric field is in V/m.

4.8 FILMTHICK

Full name: Film thickness

Type: Integer

Description: Film thickness is the number of simulation grid points for the film. Related value is **SUB-THICK**, which is the number of simulation grid for the substrate. When the film thickness is set to 0, then the program will treat the system as a bulk simulation with periodic boundary condition in all three directions rather than a thin film simulation which is only periodic in the in-plane (x,y) direction.

Notes:

1. Set to 0 for bulk system, non 0 for thin film system.

4.9 FLEXOCON

Full name: Flexoelectric constants

Type: Vector,real

Description: This vector tag sets the three independent flexoelectric constants of cubic symmetry material, i.e., longitudinal f1111, transverse f1122, shear f1221.

Notes:

1. The unit of assigned flexoelectric constant is in V.
2. The theoretical estimation of flexoelectric constant for perovskites is within 0-20 V. The sign is usually not determined.

4.10 FRACTION

Full name: Material composition fraction**Type:** Scalar,real**Description:** This tag set the fraction for materials such as PZT, PMNPT and KNN.**Notes:**

1. For $\text{Pb}(\text{Zr}_{1-x}\text{Ti}_x)\text{O}_3$, this value set the Ti fraction.
2. For PMN-xPT , this value set the fraction of PT.
3. For $\text{K}_{1-x}\text{N}_x\text{NbO}_3$, this value set the Na fraction.

4.11 GRADPCON

Full name: Gradient energy constant of polarization**Type:** Vector, real**Description:** Gradient coefficient for polarization. Specifying this value will override the default one obtained from pot.in file. Different from the value used in pot.in, which is in SI unit, the value set by GRADPCON is in reduced unit normalized by $a_0 l_0^2$, in which a_0 is chosen to be the value of a_1 , the first coefficient for landau potential, at 298k, and l_0 is the denominator for length, for ferroelectric case we use $10^{-9}m$ per grid as its value. Three values need to be specified for tag, coefficient G11, G12 and G44, as in the formular for gradient energy $\frac{G_{11}}{2}(P_{1,1}^2 + P_{2,2}^2 + P_{3,3}^2) + G_{12}(P_{1,1}P_{2,2} + P_{1,1}P_{3,3} + P_{3,3}P_{2,2}) + \frac{G_{44}}{2}[(P_{1,2} + P_{2,1})^2 + (P_{1,3} + P_{3,1})^2 + (P_{3,2} + P_{2,3})^2]$ [Phenomenological model of a 90° domain wall in BaTiO3-type ferroelectrics]PHYSICAL REVIEW B 74, 104104 2006. The default value for gradient energy coefficient is $G_{11} = 0.6$, $G_{12} = -0.6$ and $G_{44} = 0.6$. It is recommended to test the gradient energy coefficient value for your specific system, that the domain wall width should be more than 1.5 grid thick.**Notes:**

1. Default value for gradient energy coefficient is $G_{11} = 0.6$, $G_{12} = -0.6$ and $G_{44} = 0.6$.

4.12 GRADQCON

Full name: Gradient energy constant of oxygen octahedral rotation**Type:** Vector, real**Description:** Gradient coefficient for oxygen octahedral rotation. Specifying this value will override the default one obtained from pot.in file. Different from the value used in pot.in, which is in SI unit, the value set by GRADQCON is in reduced unit normalized by $b_0 l_0^2$, in which b_0 is chosen to be the value of b_1 , the first coefficient for landau potential, at 298k, and l_0 is the denominator for length, for ferroelectric case we use $10^{-9}m$ per grid as its value. Three values need to be specified for tag, coefficient G11, G12 and G44, as in the formular for gradient energy $\frac{v_{11}}{2}(q_{1,1}^2 + q_{2,2}^2 + q_{3,3}^2) + v_{12}(q_{1,1}q_{2,2} + q_{1,1}q_{3,3} + q_{3,3}q_{2,2}) + \frac{v_{44}}{2}[(q_{1,2} + q_{2,1})^2 + (q_{1,3} + q_{3,1})^2 + (q_{3,2} + q_{2,3})^2]$. The default value for gradient energy coefficient is $v_{11} = 1$, $v_{12} = -1$ and $v_{44} = 1$.**Notes:**

1. Default value for gradient energy coefficient is $v_{11} = 1$, $v_{12} = -1$ and $v_{44} = 1$.

4.13 LAFMTIP

Full name: Flag of AFM tip

Type: logic

Description: This flag controls whether to include an AFM tip (or time-variant electric bias for film systems) into the system or not. When turning on, it requires an extra input file, **AFMtip.in**, which specifies the moving paths, electric bias and mechanical load of an AFM tip. For the

Notes: format of **AFMtip.in**, please refer to Section [3.1.5](#).

4.14 LBULK

Full name: Flag for bulk or thin film system

Type: logic

Description: This flag controls whether the simulated system is treated as a bulk or thin film one. This will affect several other tags, such electric and elastic boundary conditions.

Notes: For thin film system, you need to specify the **SUBTHICK** and **FILMTHICK**

4.15 LELAS

Full name: Flag of elasticity

Type: logic

Description: This flag controls whether to consider the elastic effect or not. It is recommended to turn it on for normal ferroelectric simulations, since all ferroelectrics are ferroelastic materials, elastic effect plays a key role in determining the equilibrium domain structure. When LELAS is turned on, a mechanical equilibrium equation $\sigma_{ij,i} = 0$ is solved using spectral method, and the elastic driving force for polarization can be obtained based on the calculated stress distribution. In the thin film cases, the boundary condition is periodic in the x-y (in-plane) direction, traction (stress components that are related to the z direction, σ_{33} , σ_{13} , σ_{23}) free for the film surface, and displacement free at the some distance into the substrate, and such distance is given by the **SUBTHICK**. In the bulk cases, average stress or strain can be specified.[yulan li 2002]

Notes:

4.16 LELEC

Full name: Flag of electricity

Type: Logic

Description: This flag controls whether to consider the electric effect or not. It is recommended to turn it on for all kinds of ferroelectric simulaitions, because electric field can have a significant influence on polarization. When turned on, a Poisson equation is solved using the spectral method. In the thin film case, there is periodic boundary condition in the in-plane (x-y) direction, and short or open circuit boundary condition for the film normal direction. In the bulk cases, the average electric field can be specified.[yulan li 2002]

Notes:

4.17 LFLEXO

Full name: Flag of flexoelectricity

Type: Logic

Description: This flag controls whether to consider the flexoelectric effect or not.

Notes:

4.18 LINHOM

Full name: Flag of inhomogeneous system

Type: Logic

Description: This flag controls whether you want your simulated system to be considered as homogeneous or not. The homogeneous is not in the sense of different phases but in terms of material properties, such as elastic constant or dielectric constant. Inhomogeneous system require material properties to be set pointwise in the form of an array rather than a single constant, which will lead to significantly more memory usage. So only turn this flag on when necessary.

Notes:

1. Only use this flag when necessary, it will significantly increase both the calculation time and memory usage.

4.19 LOCTILT

Full name: Flag of oxygen octahedral rotation

Type: Logic

Description: This flag controls whether you want to evolve the oxygen octahedral tilt/rotation in the simulation system. Oxygen octahedral tilt is a second set of order parameter, and it is coupled with elasticity and polarization. [Fei paper]

Notes:

4.20 LOUDELAS

Full name: Flag of output elasticity

Type: Logic

Description: This flag controls whether output of elastic related data files are allowed. Files include **Displace**, **Eigen_St**, **stress**, **strain**, **ElastEn**, **Elas_For**, for more details, please see Section [3.2](#)

Notes:

4.21 LOUTELEC

Full name: Flag of output electricity

Type: Logic

Description: This flag controls whether output of electric related data files are allowed. Files include **Elefield**, **Elec_Phi**, **Charges**, **Elec_For**, for more details, please see Section [3.2](#).

Notes:

4.22 LOUTFLEX

Full name: Flag of output flexoelectric field

Type: Logic

Description: This flag controls whether output of flexoelectric are allowed. Files include **Eflexo**, for more details, please see Section [3.2](#).

Notes:

4.23 LOUTFORC

Full name: Flag of output all driving forces

Type: Logic

Description: This flag controls whether output of driving force is allowed. Files include **Gen_ForX**, **Gen_ForY**, **Gen_ForZ**, for more details, please see Section [3.2](#).

Notes:

4.24 LOUTGRAD

Full name: Flag of output gradient energy

Type: Logic

Description: This flag controls whether output of gradient energy is allowed. Files include **Grade_En**, **Grad_For**, for more details, please see Section [3.2](#).

Notes:

4.25 LOUTLAND

Full name: Flag of output Landau energy

Type: Logic

Description: This flag controls whether output of landau energy is allowed. Files include **LandP_En**, **LandPFor**, for more details, please see Section [3.2](#).

Notes:

4.26 LPNOISE

Full name: Flag of initial noise of polarization

Type: Logic

Description: This flag controls whether initial noise for polarization is added or not. The noise is a uniform distribution generated using the fortran subroutine RANDOM. The random seed is set with **PNOISEED**.

Notes:

4.27 LQNOISE

Full name: Flag of initial noise of oxygen octahedral rotation

Type: Logic

Description: This flag control whether initial noise for oxygen octahedral tilt is added or not. The noise is a uniform distribution generated using the fortran subroutine RANDOM. The random seed is set with **QNOISEED**.

Notes:

4.28 MISFIT

Full name: Misfit strain

Type: Vector,real

Description: Misfit strain that is due to the lattice mismatch between film and substrate, calculated by $\frac{a_s - a_f}{a_s}$, in which a_s is the lattice parameter for substrate and a_f is the lattice parameter for thin film. Three values need to set for this tag, they are (ϵ_{11} ϵ_{22} ϵ_{12}).

Notes:

4.29 PNOISEED

Full name: Seed for polarization noise

Type: Integer

Description: Set the seed value for the polarization noise generated with RANDOM subroutine. This is the value used in fortran subroutine RANDOM_SEED

Notes:

4.30 PNOISMAG

Full name: Noise magnitude for polarization

Type: Real

Description: Set the magnitude for polarization noise.

Notes:

4.31 QNOISEED

Full name: Seed for oxygen octahedral rotation

Type: Integer

Description: Set the seed value for the oxygen octahedral rotation noise generated with random subroutine. This is the value used in fortran subroutine RANDOM_SEED.

Notes: :

4.32 QNOISMAG

Full name: Noise magnitude for oxygen octahedral rotation

Type: Real

Description: Set the magnitude for oxygen octahedral rotation noise.

Notes: :

4.33 REALDIM

Full name: Real dimension

Type: Vector, real

Description: REALDIM is a 3 dimension vector specifying the real size of the simulated system in x, y and z direction with an unit of nanometer. Different from the SYSDIM value, REALDIM is real number rather than integer. In order to get a more reasonable simulation result, it is recommended that the width of domain walls no thinner than 1.5 simulation grid point. In other word, if you know the domain wall thickness for your simulated system is 1.5 nm, then each grid point (set with SYSDIM) in your system should be fine representing 1 nm, leading to $REALDIM=1\text{nm}\cdot SYSDIM$.

Notes:

4.34 ROTANGLE

Full name: Rotation Euler angles

Type: Vector

Description: Rotation angle determines the crystallography orientation of the material, it is a 3 dimensional vector of real number that correspond the the three euler angle. We used a very common proper euler angle, z-x-z, which means first angle is the rotation around z axis, second angle is the rotation around y axis in the new coordinate system, and the third value is the rotation angle around z axis in the new system.

Notes:

4.35 SCREENBOT

Full name: Screening factor at film bottom

Type: Real

Description: This value is used for setting the screening percentage of film/substrate interface charges under open circuit electric boundary conditions. It can only be used for the thin film type of system, and for open circuit conditions, for which the electric displacement is fixed.

Notes:

4.36 SCREENTOP

Full name: Screening factor at film top

Type: Real

Description: This value is used for setting the screening percentage of top surface charges under open circuit electric boundary conditions. It can only be used for the thin film type of system, and for open circuit conditions, for which the electric displacement is fixed.

Notes:

4.37 STRAIN

Full name: Applied strain

Type: Vector, Real

Description: This value is a 6 dimension vector, $(\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{23}, \epsilon_{13}, \epsilon_{12})$, specifying the applied strain for bulk system.

Notes:

4.38 STRESS

Full name: Applied stress

Type: Vector, Real

Description: This value is a 6 dimension vector, $(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{13}, \sigma_{12})$, specifying the applied stress for bulk system.

Notes:

4.39 SUBTHICK

Full name: Substrate thickness

Type: Scalar, integer

Description: The integer value set the thickness of the substrate, which is only meaningful for the thin film case. The **SUBTHICK** is in terms of simulation grid point rather than in real world unit, such as nanometer. This value actually means the thickness of substrate lattice that is been distorted due to the mismatch between film and substrate material. And at such substrate thickness away from the interface, the no displacement, or fully clamped elastic boundary condition is applied. For example, if you set **SUBTHICK** to be 10, and in your simulation system 1 grid equals 1nm, then its physical meaning is you consider the substrate lattice not affected by the thin film, if deeper than 10 nm away from the interface.

Notes:

4.40 SYSDIM

Full name: System dimension

Type: Vector(3), real

Description: This 3 dimensional vector holds the dimension in x, y and z direction of the simulation system in real world unit of nanometer.

Notes:

4.41 TDELTA

Full name: Increment time step

Type: Real

Description: This value controls the size of marching step when we evolve the TDGL equation to minimize total free energy. Larger timestep can leads to shorter simulation time, but may also easily leads to inconvergency. Recommended values are 0.01 or 0.02 .

Notes:

4.42 TEM

Full name: Temperature

Type: Real

Description: The value set the current temperature of the simulation system. Default value is room temperature of 298k, since most experiments are carried out under such conditions.

Notes:

4.43 TIPGAMMA

Full name: AFM tip gamma

Type: Real

Description: This value defines the gamma coefficient γ in a Lorentz distribution to describe the electric potential distribution on top of the film surface, i.e.,

$$\phi(x, y) = \frac{\gamma^2}{(x - x_{tip})^2 + (y - y_{tip})^2 + \gamma^2} \quad (4.1)$$

Notes:

1. The unit of assigned gamma is in nm.

4.44 TIPRAD

Full name: AFM tip radius

Type: Real

Description: This value defines the AFM tip radius (assuming spherical). Then the contact radius will be calculated according to Hertz contact model of a spherical indenter.[\[2\]](#)

Notes:

1. The unit of assigned tip radius is in nm.
2. The common value of a tip radius is around 30 - 100 nm.

4.45 TOUTPUT

Full name: Interval timestep for output

Type: Integer

Description: This value set the interval for output all output data files in Section [3.2](#).

Notes:

4.46 TSTART

Full name: Start timestep

Type: Integer

Description: This value set the starting timestep of the simulation. Default value is 0, which is the case if you're running a new simulation. But when you are restarting from a previous output, then you may want to set this TSTART value to be where you stopped before, and then it won't create confusion in the naming of output data.

Notes:

4.47 TTOTAL

Full name: Total timesteps

Type: Integer

Description: This value set the total amount of timesteps that you want to run. For example, if the TSTART value is 10000, and the TTOTAL is 10000, then as long as you have enough wall time to finish the simulation, the program will stop at step 20000.

Notes:

Chapter 5

Examples

5.1 Equilibrium domain structure of BaTiO₃ single crystal

This example is about calculating the equilibrium domain structure for bulk single crystal BaTiO₃, at room temperature, under stress free boundary conditions.

The **input.in** file is listed in Listing [5.1](#), with meaning of each line explained in the comment.

```
1 # BTO bulk
2
3 SYSDIM = 64 64 64           # Set the system's grid size to be nx=64, ny=64, nz=64
4 REALDIM = 64 64 64        # Set the system's real size to be lx=64nm, ly=64nm, lz=64nm
5 LBULK = True               # Set the simulation type to be bulk, rather than film
6
7 TEM = 298                  # Set the system temperature to be 298k,
8
9 LPNOISE = T                # Choose to add noise to polarization for initial structure
10 PNOISMAG = 0.1            # Set the magnitude for polarization noise to be 0.1
11 GRADPCON = 0.6 -0.6 0.6  # Set normalized gradient constant, G11=0.6, G12=-0.6, G44=0.6
12
13 TTOTAL = 50000            # Set the maximum time step to be 50000
14 TSTART = 0                # Set the initial time step to be 0
15 TOUTPUT = 10000          # Set the output interval to 10000 steps
16
17 LELAS = t                  # Choose to consider the elastic contribution
18 CELASBC = 2               # Use stress boundary condition
19 STRESS = 0.0 0.0 0.0 0.0 0.0 0.0 # Stress 11,22,33,23,13,12 components
20
21 LELEC = t                  # Choose to consider the electric contribution
22 CELECBC = 0               # Choose the electric boundary condition to be bulk
23 DIELECON = 45 45 45      # Set the background dielectric constant epsilon 11,22,33
24 ELECFIELD = 0 0 0        # Set the bulk electric field boundary condition x,y,z
25
26 LOUTELAS = T              # Choose to enable elastic related output
27 LOUTELEC = T              # Choose to enable electric related output
```

Listing 5.1: input.in file

The **pot.in** file is listed as in Listing [5.2](#)

```
1 {
2   "name": "BaTiO3",
3   "Comment": "Ignore the change of Landau coefficients upon hydrostatic
4     pressure",
5   "Reference": "Wang J.J. et al., JAP 2010 108(11)",
6   "Prerequisite": {
```

```

6   "CurieT": "390"
7   },
8   "P0": "0.26",
9   "Landau": {
10  "a1": "5E5*160*(cosh(160/TEM)/sinh(160/TEM)-cosh(160/390)/sinh(160/390))",
11  "a11": "-115400000",
12  "a12": "653000000",
13  "a111": "-2105000000",
14  "a112": "4091000000",
15  "a123": "-6688000000",
16  "a1111": "75900000000",
17  "a1112": "-21930000000",
18  "a1122": "-22210000000",
19  "a1123": "24160000000"
20  },
21  "Electrostrictive": {
22  "Q11": "0.11",
23  "Q12": "-0.045",
24  "Q44": "0.029"
25  },
26  "ElasticStiffness": {
27  "C11": "177940000000",
28  "C12": "96350000000",
29  "C44": "121990000000"
30  }
31 }

```

Listing 5.2: pot.in file

Always remember to check the **const.dat** file whether the values for flags, choices and parameters are the same as your expectation.

```

1  -----
2  *****Flags and choices for the simulation*****
3  -----
4  | choice_material      :          BaTiO3
5  | flag_noise_p        :          T | output_landau      :          F |
6  | flag_elastic        :          T | output_elastic     :          T |
7  | flag_electric       :          T | output_electric    :          T |
8  | flag_AFMtip        :          F | output_gradient    :          F |
9  | flag_flexo         :          F | output_flexo       :          F |
10 | output_driving_force :          F | output_time_depend :          |
11 | output_driving_force :          F | output_time_depend :          |
12 | flag_bulk           :          T | flag_inhomo        :          F |
13 | choice_elas_BC      :          2 | choice_elec_BC     :          2 |
14  -----
15  -----
16  -----
17  *****The real constants in SI units*****
18  -----
19 | a0  =  4.27695E+07 | p0  =  0.26000 | g110 =  4.27695E-11 |
20 | xf  =  1.00000 | tem =  298.00000 |
21  -----
22 | a1   =  -4.27695E+07 | a11  =  -1.15400E+08 | a12  =  6.53000E+08 |
23 | a111 =  -2.10600E+09 | a112 =  4.09100E+09 | a123 =  -6.68800E+09 |
24 | a1111 =  7.59000E+10 | a1112 =  -2.19300E+10 | a1122 =  -2.22100E+10 |
25 | a1123 =  2.41600E+10 | | |
26 | c11  =  1.77940E+11 | c12  =  9.63500E+10 | c44  =  1.21990E+11 |
27 | s11  =  9.07028E-12 | s12  =  -3.18612E-12 | s44  =  8.19739E-12 |
28 | Q11  =  1.10000E-01 | Q12  =  -4.50000E-02 | Q44  =  2.90000E-02 |
29 | eps1 =  4.50000E+01 | eps2 =  4.50000E+01 | eps3 =  4.50000E+01 |

```



```

30 | g11 = 2.56617E-11 | g12 = -2.56617E-11 | g44 = 2.56617E-11 |
31
32
33
34 *****Dimensionless material constants*****
35
36 | a1 = -1.00000 | a11 = -0.18240 | a12 = 1.03211 |
37 | a111 = -0.22502 | a112 = 0.43711 | a123 = -0.71459 |
38 | a1111 = 0.54821 | a1112 = -0.15840 | a1122 = -0.16042 |
39 | a1123 = 0.17450 | | |
40 | c11 = 61544.99741 | c12 = 33325.05620 | c44 = 42193.29119 |
41 | s11 = 0.00003 | s12 = -0.00001 | s44 = 0.00002 |
42 | Q11 = 0.00744 | Q12 = -0.00304 | Q44 = 0.00196 |
43 | eps1 = 0.01704 | eps2 = 0.01704 | eps3 = 0.01704 |
44 | g11 = 0.60000 | g12 = -0.60000 | g44(m) = 0.60000 |
45
46
47
48 *****Some other parameters*****
49
50 | Lx = 64.000 | Ly = 64.000 | Lz = 64.000 |
51 | nx = 64 | ny = 64 | nz = 64 |
52 | flim_grid = 0 | sub_grid = 0 |
53 | dx = 1.000 | dy = 1.000 | dz = 1.000 |
54

```

Listing 5.3: const.dat file

If the parameters looks fine to you, usually you should take a quick look at the **energy_out.dat** file, which contains energy summation for all system grid point of each time step. The beginning of the file is shown in Listing 5.4.

```

1 step Elastic Energy Electric Energy Landau Energy Gradient P Energy Total Energy
2 kt: 1 energy: 0.1032848345E+04 0.9398782304E+05 -0.3734208495E+05 0.2823160738E+05 0.8591019382E+05
3 kt: 2 energy: 0.8158606020E+03 0.1585332232E+05 -0.3530906729E+05 0.1840756288E+05 -0.2323214885E+03
4 kt: 3 energy: 0.7559950332E+03 0.2787724738E+04 -0.3547078752E+05 0.1542340774E+05 -0.1650366000E+05
5 kt: 4 energy: 0.7218784418E+03 0.5508686643E+03 -0.3606537342E+05 0.1371553289E+05 -0.2107709343E+05
6 kt: 5 energy: 0.6943022941E+03 0.1439256013E+03 -0.3679505762E+05 0.1236344560E+05 -0.2359338413E+05
7 kt: 6 energy: 0.6702223019E+03 0.5705124401E+02 -0.3760617678E+05 0.1119703888E+05 -0.2568186435E+05
8 kt: 7 energy: 0.6489871593E+03 0.3148713822E+02 -0.3848642736E+05 0.1017104049E+05 -0.2763491257E+05
9 kt: 8 energy: 0.6302753569E+03 0.2045260631E+02 -0.3943099878E+05 0.9263263637E+04 -0.2951700718E+05
10 kt: 9 energy: 0.6138202820E+03 0.1428600937E+02 -0.4043680075E+05 0.8457551773E+04 -0.3135114269E+05
11 kt: 10 energy: 0.5993845920E+03 0.1038491475E+02 -0.4150140340E+05 0.7740588714E+04 -0.3315104517E+05

```

Listing 5.4: Energy summation for each time step

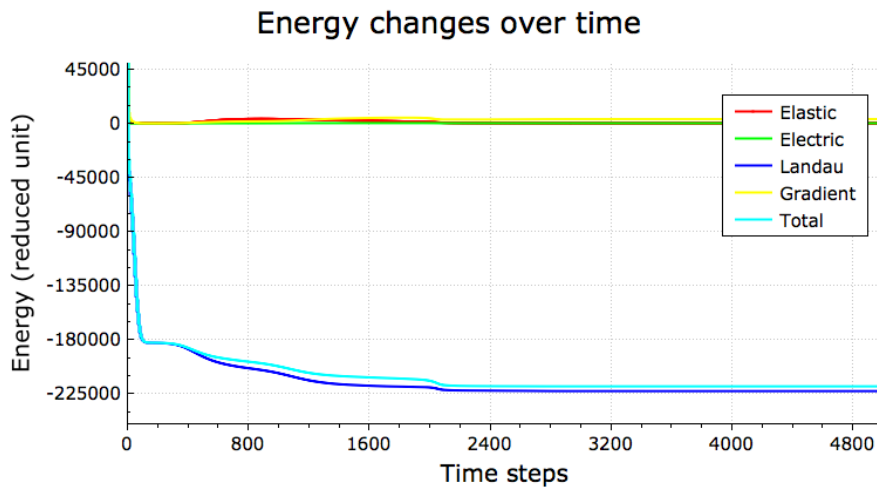


Figure 5.1: How energy components vary with time

If we use the second column as x axis and 3 to 8 columns as y axis to plot the changes of energy components in the whole system with respect to time steps, as it is shown in Figure 5.1. We can see that after about 2500 steps, the total free energy flatten out and never decrease any more. You should be careful that energy reach a plateau does not necessarily mean you have found the most stable structure of lowest energy. It may be merely a metastable structure, that the driving force of polarization is not large enough to kick the domain configuration out of this local minima. In this example, though we start from random noise, the final domain structure is extremely simple, as shown in figure 5.2. For this case, it is safe to say that what we have got is a stable domain structure. Otherwise, you should always try several different random seed to make sure what you have got is the most stable pattern. The two domains in figure 5.2 are two tetragonal domains, the yellow one is C+ domain and the black one is a2+ domain, each occupies 50% of the whole system.

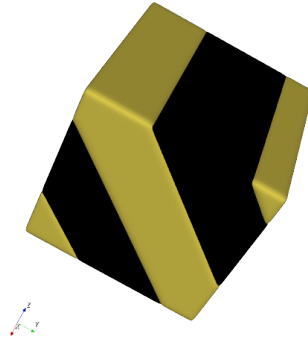


Figure 5.2: The final stable domain structure

After checking of the domain structure, you might want to take a look at the distribution of electric related variables such as electric potential, charges, electric field and electric energy, as well as elastic related variables such as total strain, eigenstrain, stress, elastic driving force and elastic energy. All of these output files can be helpful under different using situations. The ones that we strongly recommend you to visualize in 3D and investigate whether the distribution is reasonable or meets your expectation, are charge, electric potential and stress distributions. The reasons are, first, checking their distribution can let you know if you are on the right track for simulation, and whether any parameters need to be further changed before you do a lot of trials and finally realize something seems not correct or you are not changing the most relevant parameter; second, compare these relatively straight forward variables to your expectation can help you gain better feeling and understanding of the ferroelectric system, since electric and elastic contribution are extremely important to ferroelectric domain structures, they are the tool you have at hands to manipulate ferroelectric domains and in many cases helps to explain why some domain patterns are more favorable.

The charge distribution and electric potential distribution is shown in figure 5.3. From which we can see, the bound charges are mainly distributed along the domain wall. This meets our expectation, because inside a domain the polarization are the same, so there should be no bound charge. While for domain wall, as the polarization is varying, inevitably some bound charge will appear. The electric potential distribution might seems surprising at first glance, but when you read the legend, you will realize that the scale of this potential is very small.

Due to the occurrence of a2+ and C+ domain, and the stress free boundary condition we applied, the distribution of σ_{22} , σ_{33} and σ_{23} is reasonable. σ_{11} appears through the interaction with σ_{22} and σ_{33} . The two x direction related shear stress, σ_{13} and σ_{12} , are 0 because all domains are tetragonal domains, shear strain can only be induced by the domain wall and

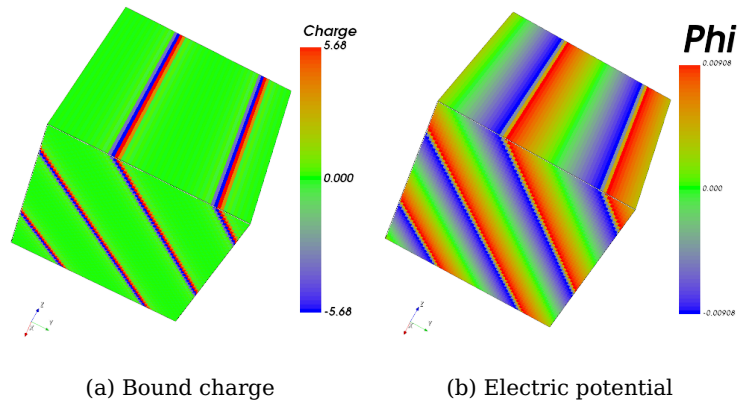


Figure 5.3: Distribution of electric related variables

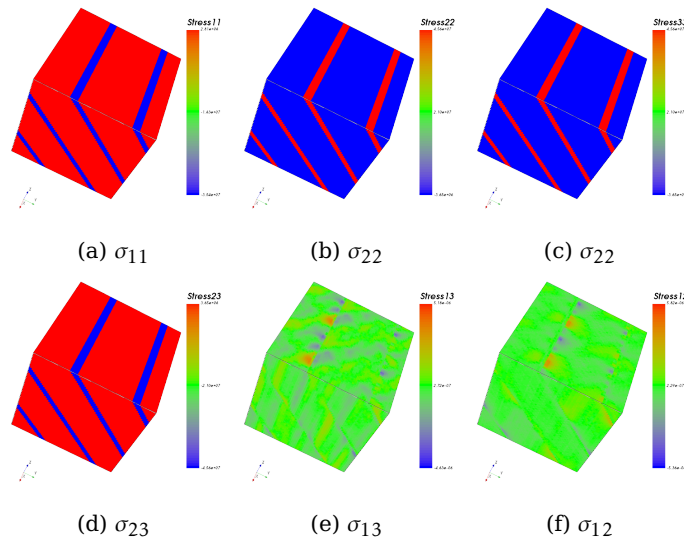


Figure 5.4: Stress distribution

as there is no a1 domains, no a1 related domain walls exist to introduce x direction related shear stress into the simulation system. You can consider this 3D simulation as an extension of 2D a2+ and C+ domains, and then the reason for missing σ_{13} and σ_{12} should become clear.

For most of your simulations, the domain structure will be more complicated than it is in this example. You may not be able to perform such straight forward analysis of stress distribution. In that case, it is a good idea to study the elastic driving force first, which is more directly related to polarization.

Another thing you should remember is, in this ferroelectric module, we are only solving an elastic mechanical equilibrium equation, $\sigma_{ij,j} = 0$, no plasticity, no large deformation is involved. In this scenario, stress is calculated from elastic strain, which is the difference between total strain and eigen strain, $\sigma_{kl} = C_{ijkl}(\epsilon_{ij} - \epsilon_{ij}^0)$, you can read this solid reference from Dr. Li published in 2002 for more details [3].

Chapter 6

Recommended references

6.1 Potential related

1. **Appendix A of the book "Physics of ferroelectrics: a modern perspective".**[\[4\]](#)

In this appendix, the Landau potential and several important physical properties that are essential to phase-field simulation are provided for many materials.

2. **A phenomenological thermodynamic potential for BaTiO₃ single crystals.**[\[5\]](#)

An eighth order polynomial of Landau-Devonshire expansion was employed, which reproduce bulk properties including the three possible ferroelectric transition temperature and their dependence on electric fields, as well as the dielectric and piezoelectric constants. This potential is also applicable to thin films under large compressive biaxial strains.

3. **Temperature-pressure phase diagram and ferroelectric properties of BaTiO₃ single crystal based on a modified Landau potential.**[\[6\]](#)

A modified eighth order Landau potential was proposed for the BaTiO₃ by taking into account the quantum mechanical effects at low temperature, and the influence of hydrostatic pressure.

4. **A modified Landau-Devonshire thermodynamic potential for strontium titanate.**[\[7\]](#)

Experiment results of strained thin film and phase-field simulations are used to refine the values of Landau energy coefficient.

5. **Tuning the remanent polarization of epitaxial ferroelectric thin films with strain.**[\[8\]](#)

The effect of biaxial strain on the remanent polarization of epitaxial thin films of various ferroelectric materials is studied using phenomenological Landau-Devonshire theory.

6. **Computer simulation of ferroelectric domain structures in epitaxial BiFeO₃ thin films.**[\[9\]](#)

Domain structure of (001), (101) and (111) oriented epitaxial BiFeO₃ thin films were studied.

6.2 Elastic related

1. **Effect of substrate constraint on the stability and evolution of ferroelectric domain structures in thin films.**[\[3\]](#).

The stability and evolution of ferroelectric domain structures in thin films are studied. Elastic solution are derived for both elastically anisotropic and isotropic thin films with arbitrary domain structures, subject to the mixed stress free and constraint boundary conditions.

6.3 Electric related

1. **Effect of electrical boundary condition on ferroelectric domain structures in thin films.**[\[10\]](#)

The effect of electrical boundary conditions on the domain morphologies and volume fractions are studied. It is shown that different electric boundary conditions may have a significant effect on the domain structures.

2. **Ferroelectric domain structures in $\text{SrBi}_2\text{Nb}_2\text{O}_9$ epitaxial thin films: Electron microscopy and phase-field simulations.**[\[11\]](#)

The effects of ferroelastic distortion and dielectric constant on ferroelectric domains are systematically analyzed. It is demonstrated that electrostatic interactions which favor straight domain walls are not sufficient to overcome the domain wall energy which favors curved domains in $\text{SrBi}_2\text{Nb}_2\text{O}_9$.

3. **Phase-field simulation of polarization switching and domain evolution in ferroelectric polycrystals.**[\[12\]](#)

A phase-field model is developed for predicting the polarization switching and domain structure evolution under an applied electric field in ferroelectric polycrystals. A hysteresis loop, average polarization as a function of applied electric field, is computed, and the detailed domain evolution process during switching is analyzed.

6.4 Tip related

1. **Effect of ferroelastic twin walls on local polarization switching: Phase-field modeling.**[\[13\]](#)

Local polarization switching in epitaxial ferroelectric thin films in the presence of ferroelastic domain walls was studied using phase-field approach. The nucleation bias profile across a twin wall was analyzed, and the localization of preferential nucleation sites was established.

2. **Nanoscale mechanical switching of ferroelectric polarization via flexoelectricity.**[\[14\]](#)

We examine the competition between pizeoelectric and flexoelectric effects and provide an understanding of the role of flexoelectricity in the polarization switching.

Bibliography

- [1] Fei Xue, Yijia Gu, Linyun Liang, Yi Wang, and Long-Qing Chen. Orientations of low-energy domain walls in perovskites with oxygen octahedral tilts. *Physical Review B*, 90(22):220101, 2014.
- [2] Anthony C Fischer-Cripps. *Introduction to contact mechanics*. Springer, 2000.
- [3] YL Li, SY Hu, ZK Liu, and LQ Chen. Effect of substrate constraint on the stability and evolution of ferroelectric domain structures in thin films. *Acta materialia*, 50(2):395–411, 2002.
- [4] Karin M Rabe, Charles H Ahn, and Jean-Marc Triscone. *Physics of ferroelectrics: a modern perspective*, volume 105. Springer Science & Business Media, 2007.
- [5] YL Li, LE Cross, and LQ Chen. A phenomenological thermodynamic potential for ba ti o 3 single crystals. *Journal of Applied Physics*, 98(6):064101, 2005.
- [6] JJ Wang, PP Wu, XQ Ma, and LQ Chen. Temperature-pressure phase diagram and ferroelectric properties of batio 3 single crystal based on a modified landau potential. *Journal of Applied Physics*, 108(11):114105, 2010.
- [7] Guang Sheng, YL Li, JX Zhang, S Choudhury, QX Jia, Venkatraman Gopalan, Darrell G Schlom, ZK Liu, and LQ Chen. A modified landau-devonshire thermodynamic potential for strontium titanate. *Applied Physics Letters*, 96(23):232902, 2010.
- [8] JX Zhang, DG Schlom, LQ Chen, and CB Eom. Tuning the remanent polarization of epitaxial ferroelectric thin films with strain. *Applied Physics Letters*, 95(12):122904, 2009.
- [9] JX Zhang, YL Li, S Choudhury, LQ Chen, YH Chu, F Zavaliche, MP Cruz, R Ramesh, and QX Jia. Computer simulation of ferroelectric domain structures in epitaxial bi fe o 3 thin films. *Journal of Applied Physics*, 103(9):094111, 2008.
- [10] YL Li, SY Hu, ZK Liu, and LQ Chen. Effect of electrical boundary conditions on ferroelectric domain structures in thin films. *Applied physics letters*, 81(3):427–429, 2002.
- [11] YL Li, LQ Chen, G Asayama, DG Schlom, MA Zurbuchen, and SK Streiffer. Ferroelectric domain structures in srbi 2 nb 2 o 9 epitaxial thin films: electron microscopy and phase-field simulations. *Journal of applied physics*, 95(11):6332–6340, 2004.
- [12] S Choudhury, YL Li, C EIII Krill, and L-Q Chen. Phase-field simulation of polarization switching and domain evolution in ferroelectric polycrystals. *Acta Materialia*, 53(20):5313–5321, 2005.
- [13] S Choudhury, JX Zhang, YL Li, LQ Chen, QX Jia, and SV Kalinin. Effect of ferroelastic twin walls on local polarization switching: phase-field modeling. *Applied Physics Letters*, 93(16):162901, 2008.

- [14] Yijia Gu, Zijian Hong, Jason Britson, and Long-Qing Chen. Nanoscale mechanical switching of ferroelectric polarization via flexoelectricity. *Applied Physics Letters*, 106(2):022904, 2015.