Petaspin_microelectrics Solver

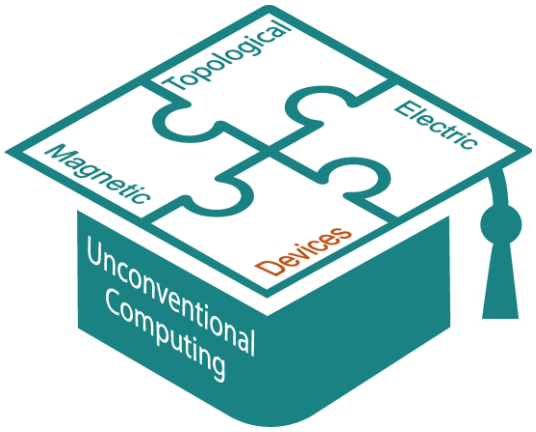
A desktop GUI for the TOPOCOM ferroelectric phase-field solver.

Authors: Ali Hasan[1], Abdul Basit[1], Davi Rodrigues[2], Giovanni Finocchio[1]
[1]Department of Mathematical and Computer Sciences, Physical Sciences, and Earth Sciences, University of Messina, I-98166, Messina, Italy
[2]Department of Electrical and Information Engineering, Politecnico di Bari, I-70125 Bari, Italy

The Petaspin_microelectrics Solver is a PyQt5-based desktop application that provides a complete graphical interface to a high-performance CUDA™ (Compute Unified Device Architecture) C/C++ phase-field solver for ferroelectric materials. It lets researchers configure a three-dimensional ferroelectric geometry, define constitutive parameters, apply external electric-field excitations, run the simulation, and visualize output: all without editing configuration files by hand.

Developed as part of the TOPOCOM initiative (NTNU) within the PETASPIN group (University of Messina).

TOPOCOM Doctoral Network · Marie Skłodowska-Curie Actions



Funded by the European Union

Table of Contents

Funded by the European Union

Funded by the European Union

Funded by
the European Union

Installation

The Petaspin_microelectrics Solver is distributed as a Windows installer package. Follow these steps to install it on your machine.

Step 1: Obtain the Installer

Locate the installer file (topocom_simulator_setup.exe) provided by the TOPOCOM webpage.

Step 2: Run the Installer

Double-click topocom_simulator_setup.exe to launch the setup wizard.

Step 3: Select Language

Choose the setup language (English or Italian) and click OK.

Step 4: Choose Installation Location

The installer will prompt you to select the destination folder. The default location is:

```
C:\Program Files\Petaspin Simulator
```

You can accept the default or click Browse… to choose a different folder. Click Next to continue.

Step 5: Additional Tasks

If desired, check the Create a desktop icon option to place a shortcut on your desktop. Click Next.

Step 6: Review and Install

Review the settings shown on the summary page, then click Install to begin copying files. The installation typically takes 30 seconds to 2 minutes depending on your system.

What the Installer Does

The installer performs the following actions automatically:

- Installs the application to your chosen program folder. This includes the GUI executable (GUI.exe), all Python runtime dependencies, PyQt5 libraries, matplotlib, numpy, KaTeX assets, and application icons.
- Creates C:\MicroElectricTemplate on the system drive. This template directory contains:
- microelectric_gpu.exe: the CUDA/GPU version of the solver.
- microelectric_cpu.exe: the CPU-only version of the solver.
- file_configuration/: a folder holding default .dat configuration files (geometry, Landau coefficients, gradient terms, electrostatic settings, external field, initial polarization, output settings).
- Registers Start Menu shortcuts under Petaspin Simulator.

Funded by the European Union

- Optionally creates a desktop icon if you selected that option.

Step 7: Launch the Application

After installation completes, check Launch Petaspin Simulator and click Finish to run the application immediately. Alternatively, launch it later from the Start Menu or desktop shortcut.

Uninstalling

To remove the application, open Settings → Apps → Installed apps, locate Petaspin Simulator, and click Uninstall. The template directory C:\MicroElectricTemplate is preserved during uninstall so that any custom modifications you may have made to the template configuration files are not lost. You can delete it manually if you no longer need it.

Funded by
the European Union

Minimum System Requirements

- Operating System: Windows 10 (64-bit) or Windows 11 (64-bit).
- Processor: x86-64 compatible CPU, 2 GHz or faster.
- Memory: 8 GB RAM minimum; 16 GB or more recommended for large simulations.
- Disk Space: 1 GB for the application and template; additional space for simulation outputs.
- Graphics (for GPU backend): NVIDIA GPU with CUDA compute capability 6.0 or higher, and up-to-date NVIDIA drivers.
- Display: 1366 × 768 minimum; 1920 × 1080 or higher recommended.

The CPU backend requires no GPU or CUDA installation and works on any 64-bit Windows machine.

Getting Started

Launching the Application

Double-click the Petaspin Simulator shortcut (desktop or Start Menu). The launcher window opens showing the TOPOCOM logo, a welcome message, and a single button:

- PETASPIN Microelectric: opens the main solver GUI.

Click the button to proceed. The main window will appear centered on your screen.

The TOPOCOM launcher window

Funded by the European Union

Creating Your First Project

When the main window opens, a startup dialog invites you to either:

- New Project: creates a fresh project folder with default .dat files and both solver executables copied from the template.
- Open Project: opens an existing project folder that already contains the required structure.

Choose New Project. A folder-picker dialog appears: select any location on your disk (for example, D:\MySimulations\Project1). The application will:

- Create the folder if it does not exist.
- Copy file_configuration/ from C:\MicroElectricTemplate into the project folder.
- Copy both microelectric_gpu.exe and microelectric_cpu.exe into the project folder.
- Set the project folder as the current working directory.

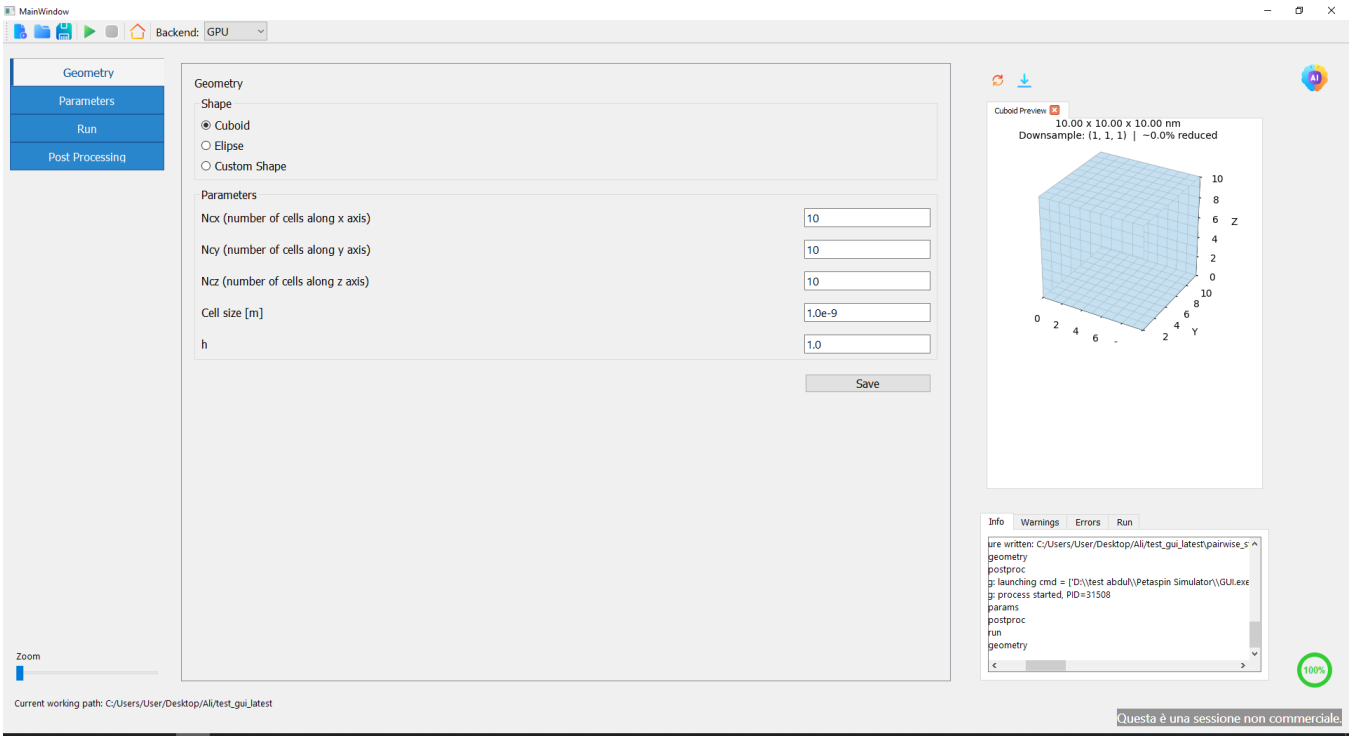
The current project directory is displayed in the toolbar bar so you always know where your work is being saved.

The Main Window

The main window is divided into four main areas:

- Top toolbar: global actions (New, Open, Save, Play, Stop, Home) and the CPU/GPU backend selector.
- Left navigation: four tabs: Geometry, Constitutive Parameters, Run, Post-Processing.
- Center content area: the parameter panel for the currently selected tab.
- Right display area: geometry previews, equation renders, file viewers, and log tabs.

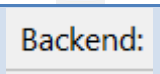
Funded by the European Union



The four main areas of the main window

1. Toolbar: New / Open / Save / Play / Stop / Home + backend selector
2. Left navigation: Geometry, Parameters, Run, Post-Processing
3. Center panel: Settings for the selected page
4. Preview / log panel: Geometry preview, equations, run log

Toolbar Actions

Icon	Name	Description
	New	Create a new project in a chosen folder.
	Open	Open an existing project folder.
	Save	Save all pages at once (writes all .dat files).
	Play	Run the simulation using the currently selected backend.
	Stop	Stop a running simulation.
	Home	Return to the launcher window.
	Backend: GPU / CPU dropdown	Select which solver executable to run. Default is GPU.

Funded by the European Union

The Save button writes changes on every page simultaneously. Individual page-level Save buttons are also available inside each page for finer-grained control.

Log and Preview Panel

The AI button in the top-right corner of the window is a placeholder for a planned future feature and is not yet functional.

The right-hand panel shows four tabs: Info, Warnings, Errors, and Run.

- Info logs routine actions with a timestamp, for example saving a page, opening a project, or updating a flag. Example: [2026-08-29 08:31:58] ✅ Landau YES: Landau_free.dat updated (alpha0).
- Warnings flags things that may need attention but do not stop the interface, for example: [2026-08-29 08:31:58] ⚠️ Comment not found in external_field.dat: 'FLAG_EXT_FIELD 0 (No), 1 (yes)'.
- Errors reports validation failures that prevent saving or running, for example: [2026-08-29 12:41:55] ❌ Custom shape validation failed: Expected 125000 rows (Ncx×Ncy×Ncz), got 2.
- Run shows the live output of the solver executable while a simulation is in progress.

When a file is selected for upload (a custom shape mask, a non-uniform parameter file, and so on), its contents are displayed directly in the preview panel above these tabs, so the values can be checked before saving.

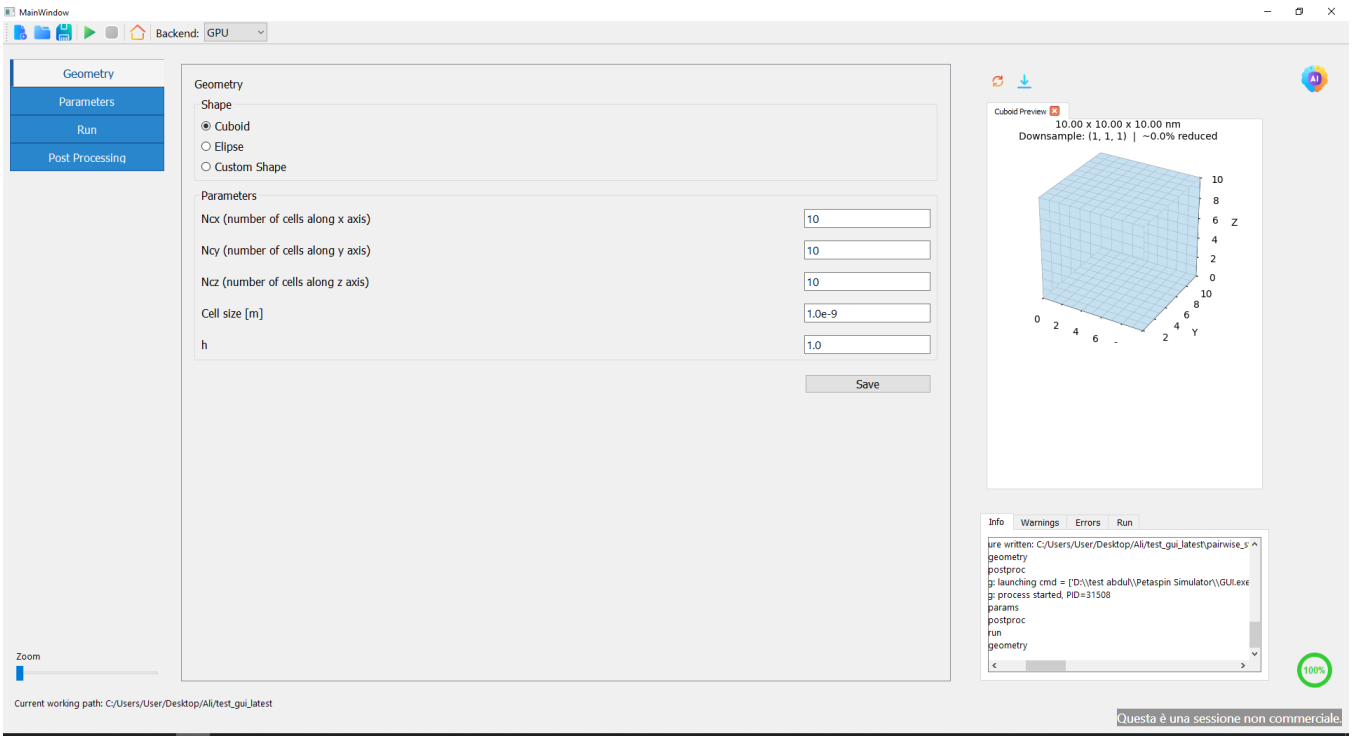
Zoom Slider

The bottom of the window includes a horizontal zoom slider that scales all UI text and equation renderings from 0.6× to 1.2× (default 1.0×). Useful when working on high-DPI monitors or when presenting.

Funded by the European Union

Geometry Page

Configure the shape and mesh of the ferroelectric sample.



The Geometry page

1. Shape: Cuboid, Ellipse, or Custom Shape (upload a .dat mask)
2. Mesh parameters: Ncx, Ncy, Ncz cell counts, cell size, and h
3. three-dimensional preview: Rotate, zoom, and export the current geometry

Shape Selection

Choose one of three shape types:

- Cuboid: a rectangular block. Dimensions are Ncx × Ncy × Ncz cells with a uniform cell size.
- Ellipse: an ellipsoidal region carved out of the Ncx × Ncy × Ncz bounding grid. Cells whose center lies inside the ellipsoid are active.
- Custom Shape: upload a .dat file (one column of 0 or 1 values, one entry per cell, Ncx × Ncy × Ncz rows in total) to define an arbitrary voxelized shape.

Geometry Parameters

- Ncx / Ncy / Ncz: integer number of cells along each axis. Only integers are accepted.

Funded by the European Union

- Cell size [m]: the spatial discretization step. Applied to both Δx and Δy. Δz is computed automatically as h × cell_size.
- h: a dimensionless multiplier relating the z-axis spacing to the cell size, Δz = h × Δx (equivalently, h = Δz / Δx).

Default values are loaded from ferro_geometry.dat.

Custom Shape Upload

When you select Custom Shape, an Upload .dat button becomes active. Click it to browse for a shape file. The file must:

- Contain exactly one column of values.
- Have exactly Ncx × Ncy × Ncz rows.
- Contain only 0 (inactive cell) or 1 (active cell).

If the file passes validation, the shape file name is stored in ferro_geometry.dat, the file is copied to the project directory, and a three-dimensional voxel preview is displayed.

Preview

The Refresh Preview button generates an interactive three-dimensional voxel figure of the current shape. You can:

- Rotate with left-click + drag.
- Zoom the whole figure with the mouse wheel.
- Pan by dragging the scroll bars around the preview area.

The Export Preview button saves the current preview as a PNG file.

Saving

Click Save Geometry to write the current settings to ferro_geometry.dat. This also updates the shape flag (FLAG_SHAPE: 1 = cuboid, 2 = custom shape, 3 = ellipse) and refreshes the electrostatic padding boxes on the parameters page.

Funded by the European Union

Constitutive Parameters Page

The constitutive parameters page is divided into seven sub-pages, accessible from a top row of tabs:

- Landau: Landau–Devonshire free energy coefficients ($\alpha_0, \alpha_{11}, \alpha_{12}, \alpha_{111}, \alpha_{112}, \alpha_{123}, T, T_0$).
- Gradient: gradient energy coefficients ($G_{11}, G_{12}, G_{44}, G'_{44}, G_{110}$).
- Electrostatic: vacuum and background permittivity, padding size along each axis (default padding along *x* = Ncx, padding along *y* = Ncy, padding along *z* = Ncz) and field weights.
- External Field: DC field, AC field, phase, field pulse, and sinc excitation.
- Strain: currently under development; a placeholder message is displayed.
- Run Parameters: time integrator, energy computation, run mode (Relaxation TDGL / Dynamics LKT), initial polarization, relaxation coefficient, anisotropy, damping (LKT-only), inertia (LKT-only).
- Output: controls which output files the solver writes (OOMMF, local polarization, average polarization, local energy, average energy, local field, average field), each with a Yes/No toggle and an interval.

Each sub-page displays LaTeX-rendered equations (via KaTeX) for reference and reads default values from the appropriate .dat file when the project is opened.

Yes/No Sections

For pages that support a Yes/No toggle at the top (Landau, Gradient, External Field, sinc excitation, etc.), selecting No disables the inputs on that section but keeps the section visible with all fields grayed out. Selecting Yes enables them.

Uniformity Dropdowns

Many parameters offer three uniformity modes:

- Uniform: a single constant value used everywhere.
- Thickness-Dependent: upload a .dat file with one value per z-layer (Ncz rows).
- Spatially Dependent: upload a .dat file with one value per cell (Ncx × Ncy × Ncz rows).

Selecting a mode switches the input to the appropriate widget (textbox for uniform, file picker for thickness- or spatially-dependent).

Parametric Studies

Landau coefficients, external DC field magnitudes, external AC field magnitudes, frequencies, and phases each support parametric studies. When a study mode is selected, the user provides a range or a list of values, and the GUI expands the study into multiple .dat files, one per parameter combination, when the simulation is launched.

Run Parameters Highlights

- Time Integrator: Heun, RK45 Fixed, RK45 Variable, or AB3M2 Fixed. RK45 Variable enables extra tolerance controls (initial adaptive steps, error tolerance, max/min timestep factors).

Funded by the European Union

- Stopping Criteria: a minimum effective-field threshold. In Relaxation (TDGL) mode, once the overall field magnitude drops below this value, the sample is considered to have reached equilibrium and the simulation stops automatically.
- RK45 Adaptive Method Settings (Initial Adaptive Steps, Error Tolerance, Maximum and Minimum Time-Step factors) are only editable when Time Integrator is set to RK45 Variable. For every other integrator these fields are grayed out and have no effect.
- Run Mode: Relaxation (TDGL) for standard first-order dynamics, or Dynamics (LKT) for second-order Landau-Khalatnikov-Tani dynamics. When Relaxation is selected, the damping and inertia inputs are disabled.
- Initial Polarization: Uniform, Thickness-Dependent, Spatially Dependent, or Random. Random draws each component independently from a uniform distribution over the specified per-axis min/max bounds.
- Equilibrium Polarization [C/m²]: a reference value used to normalize the polarization. The default, 1.0, means the simulation runs directly in SI units with no rescaling.
- α1 Normalizer: a reference value used to normalize the α1 coefficient. Default 1.0.
- Relaxation Coefficient [s^{-1} m^3 C^{-1} N^{-1}]: a single reference value used purely to normalize simulation time (it rescales the total simulation time and time step into the solver's internal units, alongside the α1 Normalizer above). It is not itself a physical relaxation rate. Default 1.0.
- Relaxation Weight (Relaxation Weight Uniformity): unlike the Relaxation Coefficient above, this is the actual physical relaxation coefficient *L* used directly in the TDGL/LKT equations, typically taken from experiment. It supports Uniform, Thickness-Dependent, or Spatially Dependent input, which is useful for samples made of more than one stacked material.
- Anisotropy Uniformity: the uniaxial anisotropy coefficient, also available as Uniform, Thickness-Dependent, or Spatially Dependent.
- Damping coefficient [Ωm]: used only in Dynamics (LKT) mode; supports uniform, thickness-dependent, or spatially-dependent input.
- Inertia coefficient [J m s^2 C^{-2}]: used only in Dynamics (LKT) mode.

DC Field: Coordinate System

The DC field can be entered in either of two coordinate systems, chosen from the Field Coordinate System dropdown at the top of the External Field page.

- Cartesian: components *Ex*, *Ey*, *Ez* [V/m].
- Polar: magnitude *E* [V/m], polar angle θ[°], and azimuthal angle φ [°].

Each coordinate system supports the same three uniformity modes described above, plus a Parametric Study. When uploading a file for Thickness-Dependent, Spatially Dependent, or Parametric Study mode, the required columns depend on the coordinate system selected:

- Cartesian file: three columns, in order *Ex*, *Ey*, *Ez*.
- Polar file: three columns, in order *E* (magnitude), θ[°](polar angle), φ (azimuthal angle).

As with other non-uniform inputs, the row count must match the mode: Ncz rows for Thickness-Dependent, Ncx × Ncy × Ncz rows for Spatially Dependent, or one row per simulation for Parametric Study.

Funded by the European Union

Field Pulse

Include Field Pulse: Yes / No. When enabled, the pulse is defined by a two-column file, with one row per pulse point:

- Column 1: the field value at that point.
- Column 2: the corresponding time value.

The solver linearly interpolates between consecutive rows as the simulation advances. The Number of Pulses field sets how many rows the file must contain.

The pulse direction is set separately from the file, with two manually entered angles that apply to the whole pulse: Pulse Polar Angle θ[°] and Pulse Azimuthal Angle φ [°].

Sinc Excitation

The External Field sub-page also includes a broadband sinc excitation, a spatially Gaussian, temporally sinc-shaped electric pulse. This is the excitation used to extract ferron dispersion relations, since its broad frequency content probes many modes in a single run.

Include sinc excitation: Yes / No. When enabled, the following parameters become editable.

- *Ex*, *Ey*, *Ez* [V/m]: peak amplitude of the sinc pulse, per Cartesian component.
- Width of the Gaussian spatial envelope [cells]: controls how localized the excitation is in space.
- Cutoff frequency *f*c [Hz]: sets the frequency bandwidth of the pulse.
- Temporal center of the sinc pulse *t*0 [s].
- Sinc excitation start time and end time [s]: the active window during which the pulse is applied.

Saving

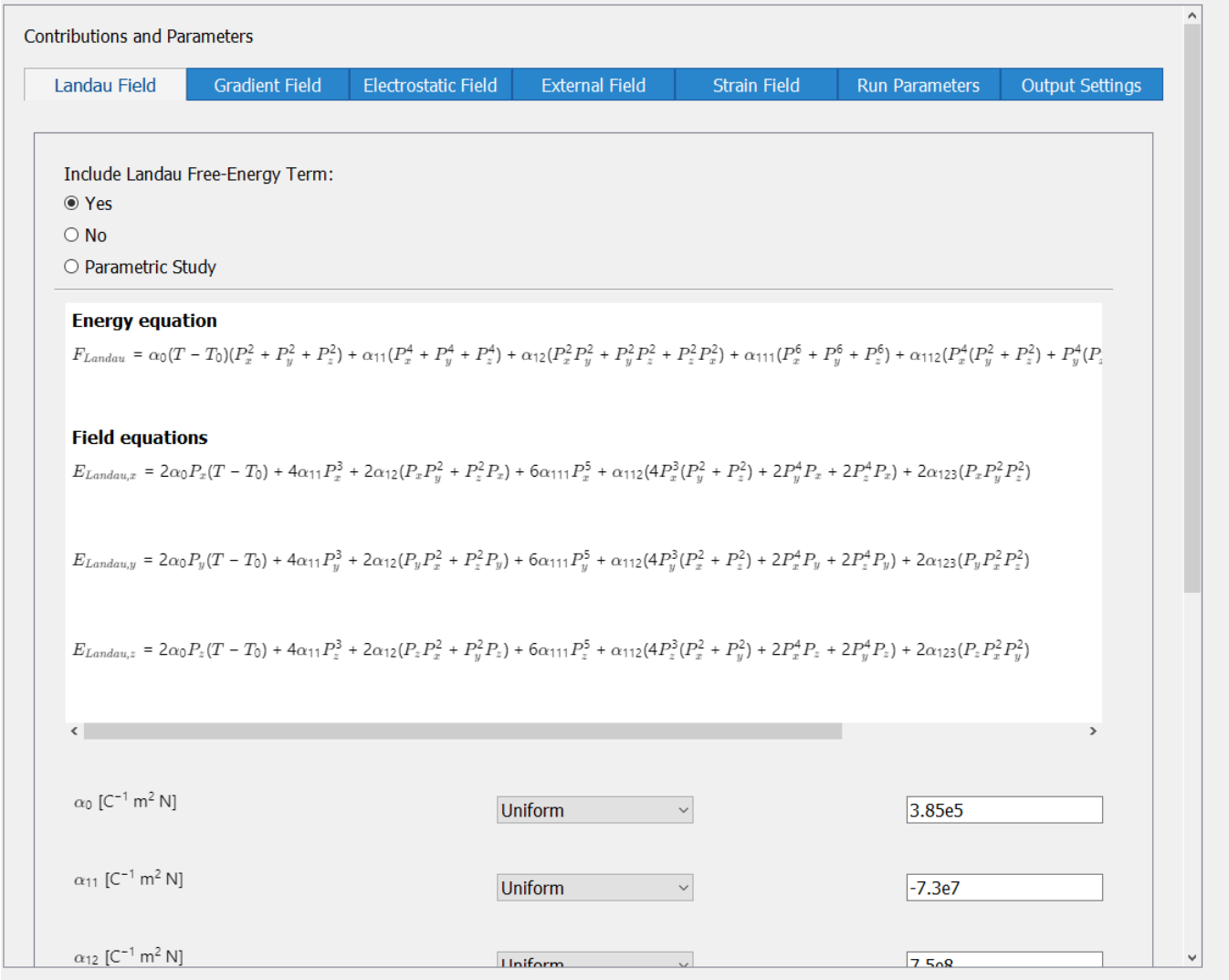
Each sub-page has its own Save button, or use the toolbar Save to save all pages at once. Changes are not written to disk until a Save action is triggered.

Sub-Page Reference

Views of each of the seven Constitutive Parameters sub-pages, for quick visual reference, with the corresponding settings listed beneath each one.

Funded by the European Union

Landau Field

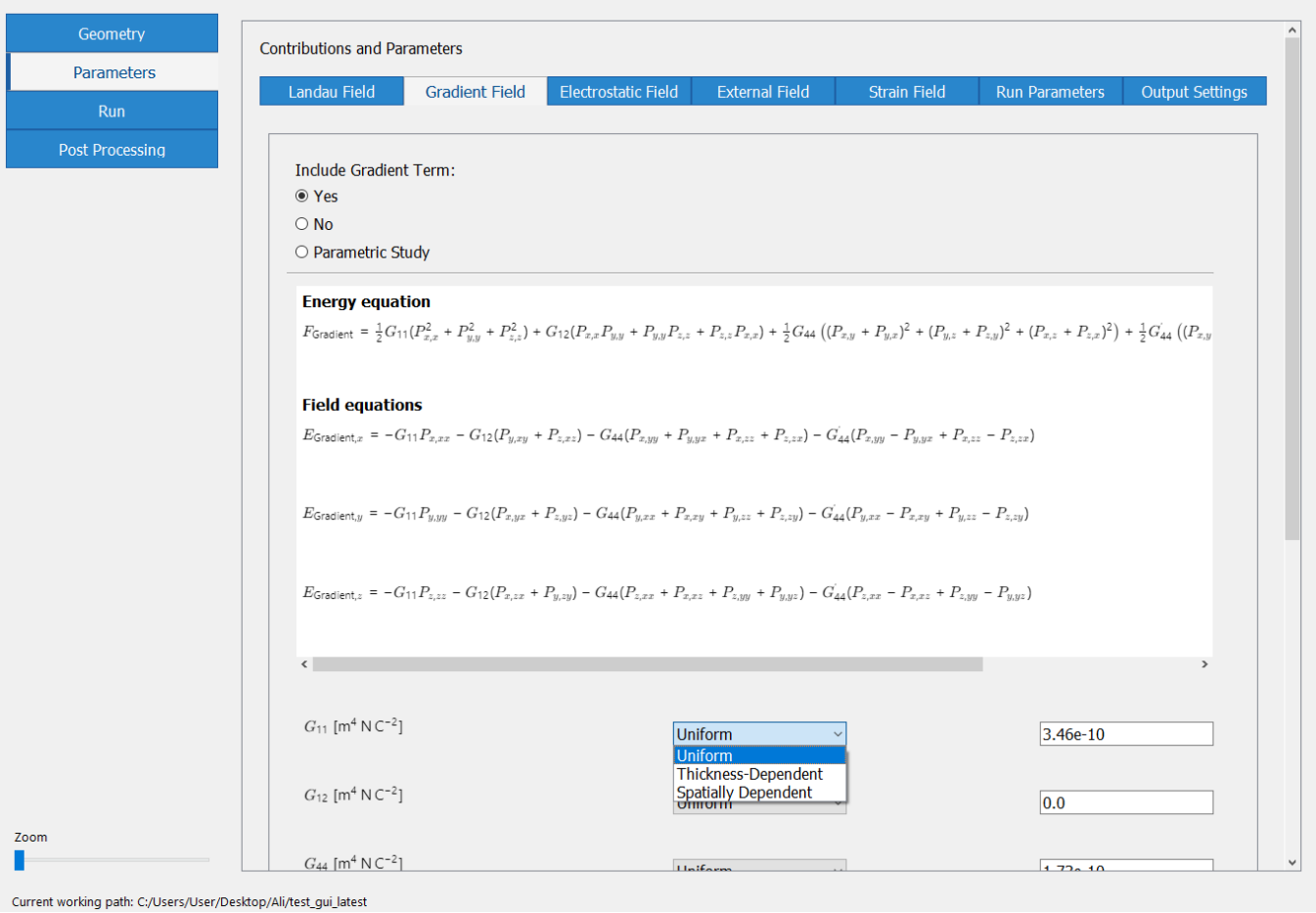


The Landau Field sub-page

1. Enable toggle: Yes / No / Parametric Study
2. Reference equations: Rendered energy and field equations
3. Coefficients: α_0...aα_{123}, each Uniform / Thickness / Spatially dependent

Funded by the European Union

Gradient Field

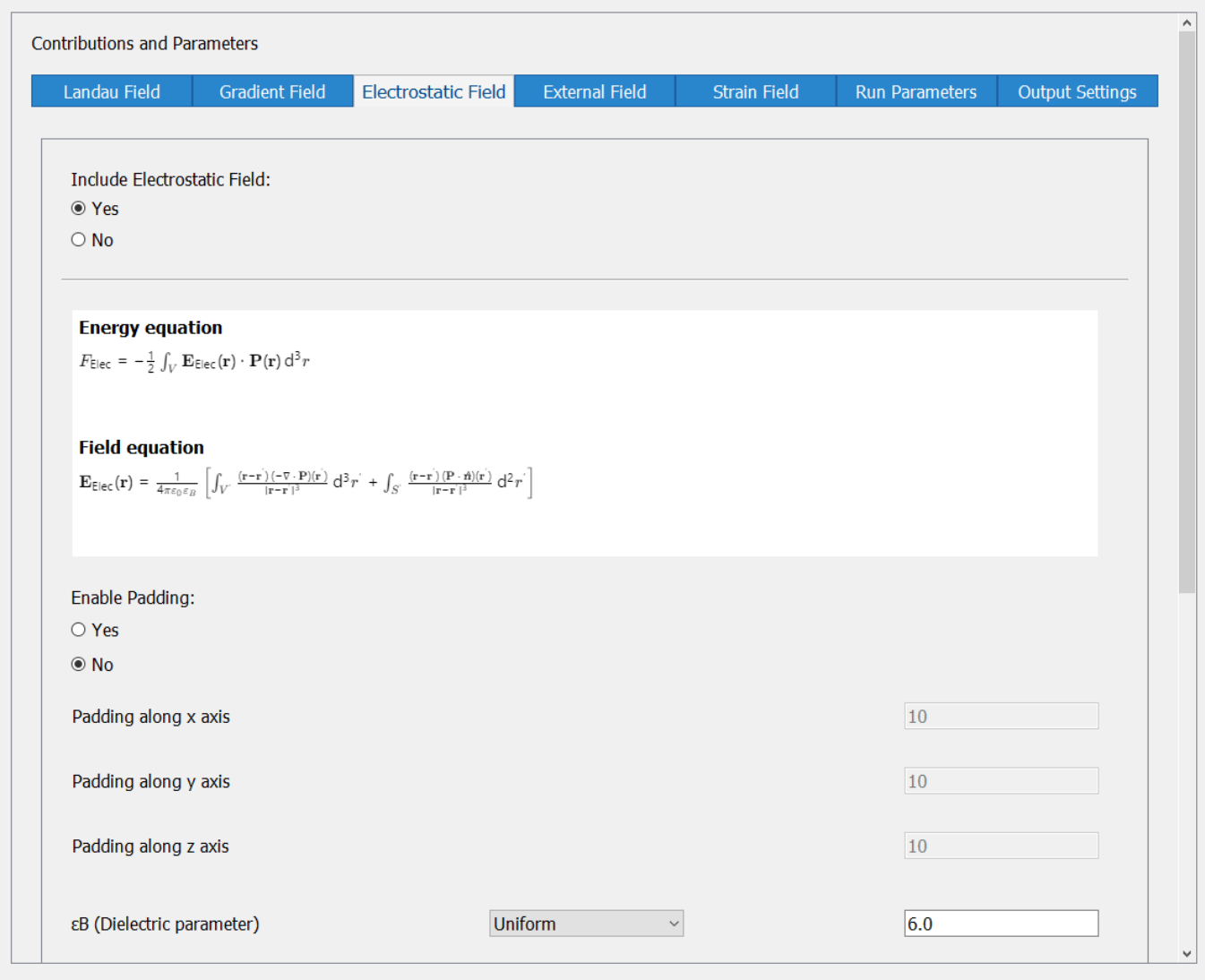


The Gradient Field sub-page

1. Enable toggle: Yes / No / Parametric Study
2. Reference equations: Gradient energy and field equations
3. Uniformity dropdown: Uniform / Thickness-Dependent / Spatially Dependent

Funded by the European Union

Electrostatic Field

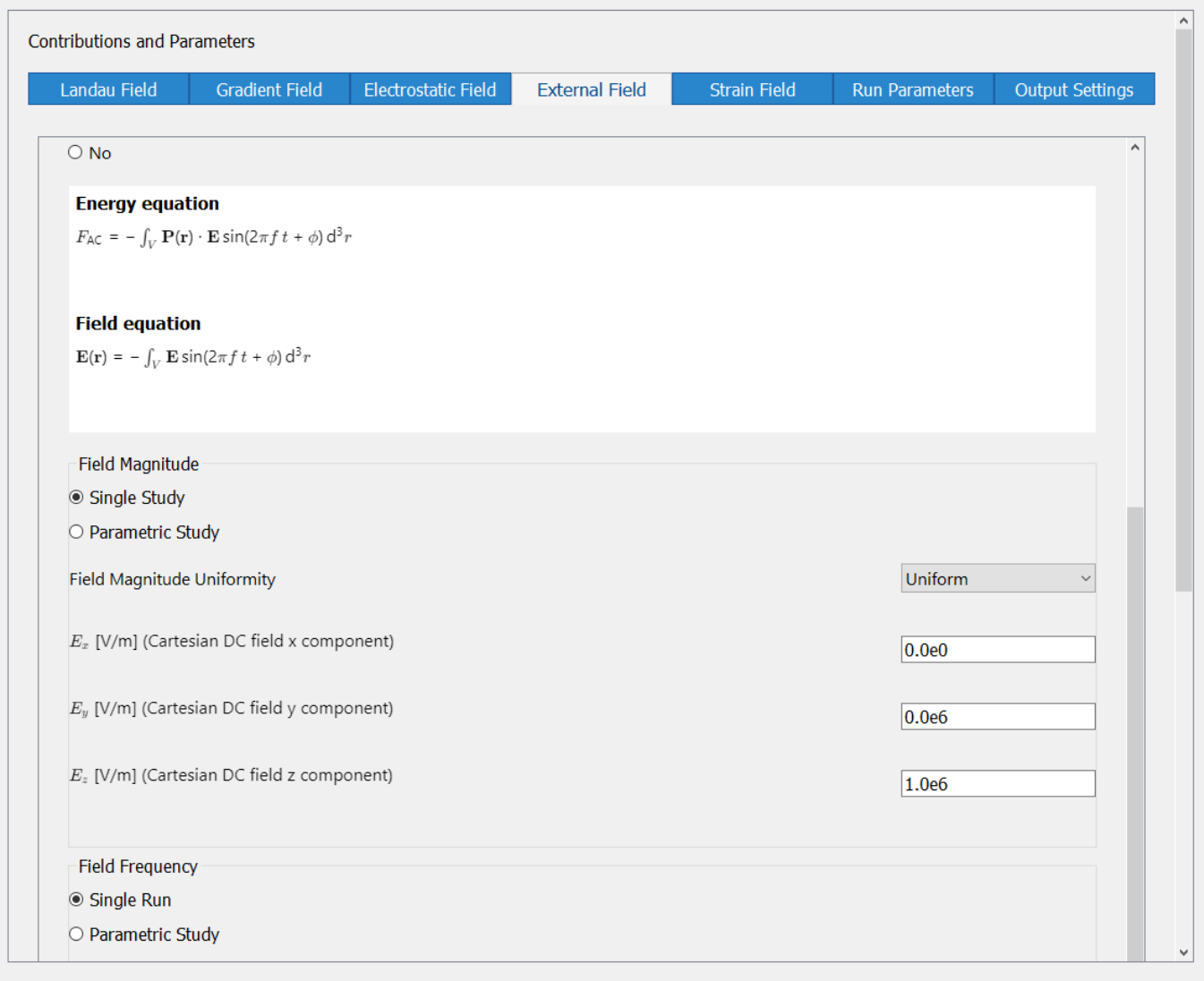


The Electrostatic Field sub-page

1. Enable toggle: Include Electrostatic Field: Yes / No
2. Reference equations: Energy and field equations
3. Padding: Optional padding along x / y / z
4. Permittivity: Background dielectric parameter

Funded by the European Union

External Field

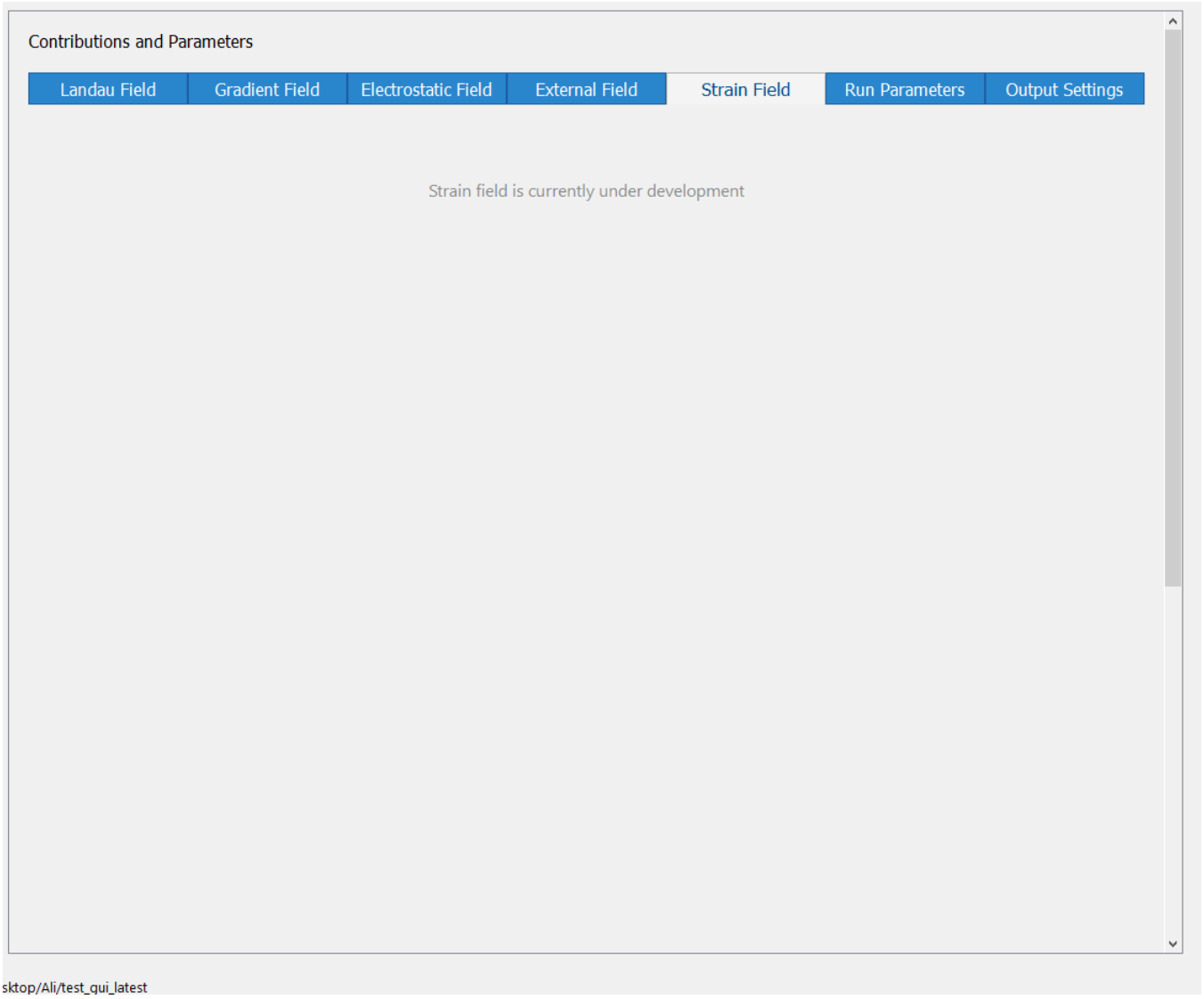


The External Field sub-page

1. Field type: DC / AC / Pulse / Sinc, each independently toggled
2. Reference equations: Energy and field equations per excitation type
3. DC field components: Ex, Ey, Ez in V/m
4. Study mode: Single value or parametric sweep

Funded by the European Union

Strain Field

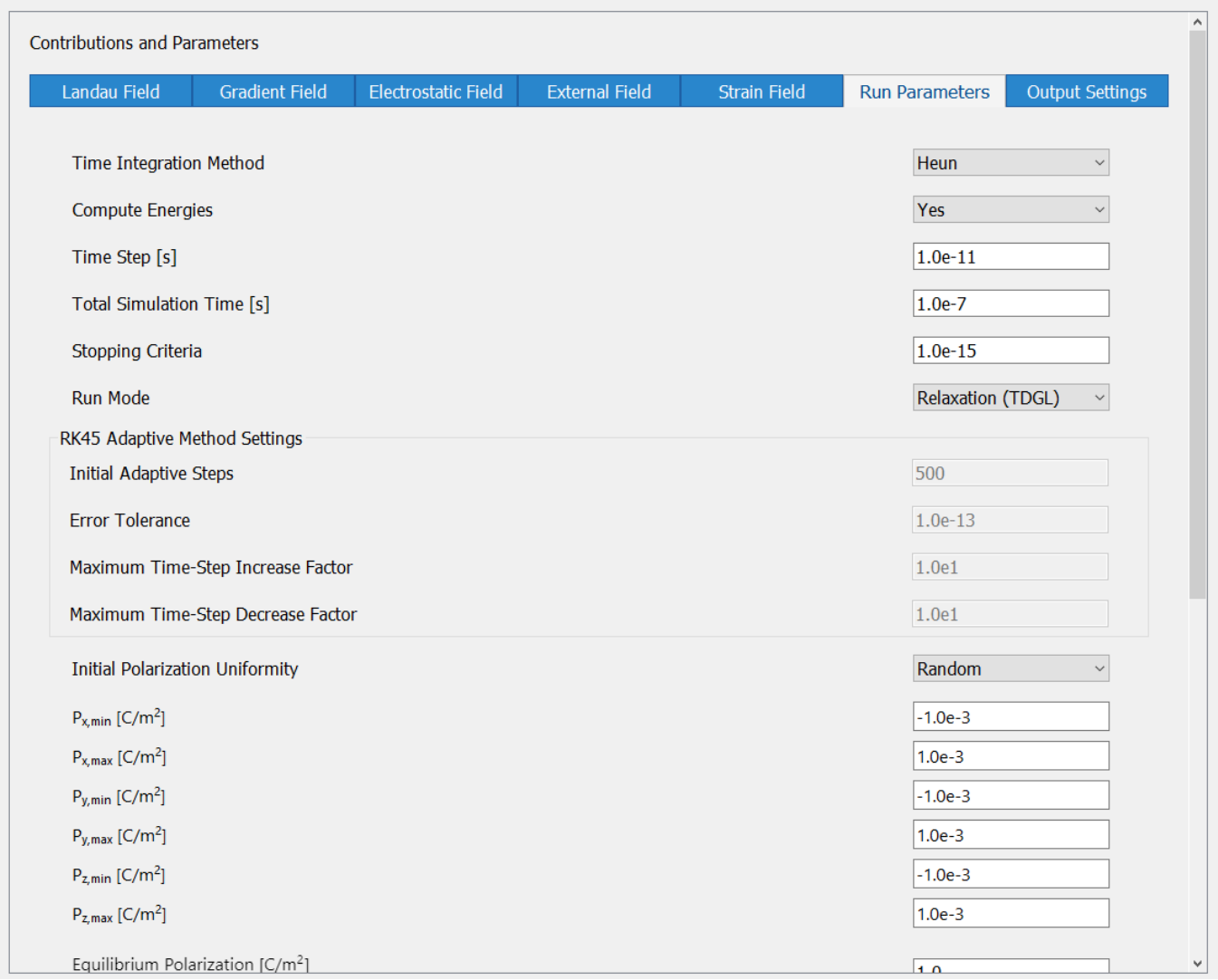


The Strain Field sub-page

1. Placeholder: This tab is reserved for a future elastic/strain module

Funded by the European Union

Run Parameters

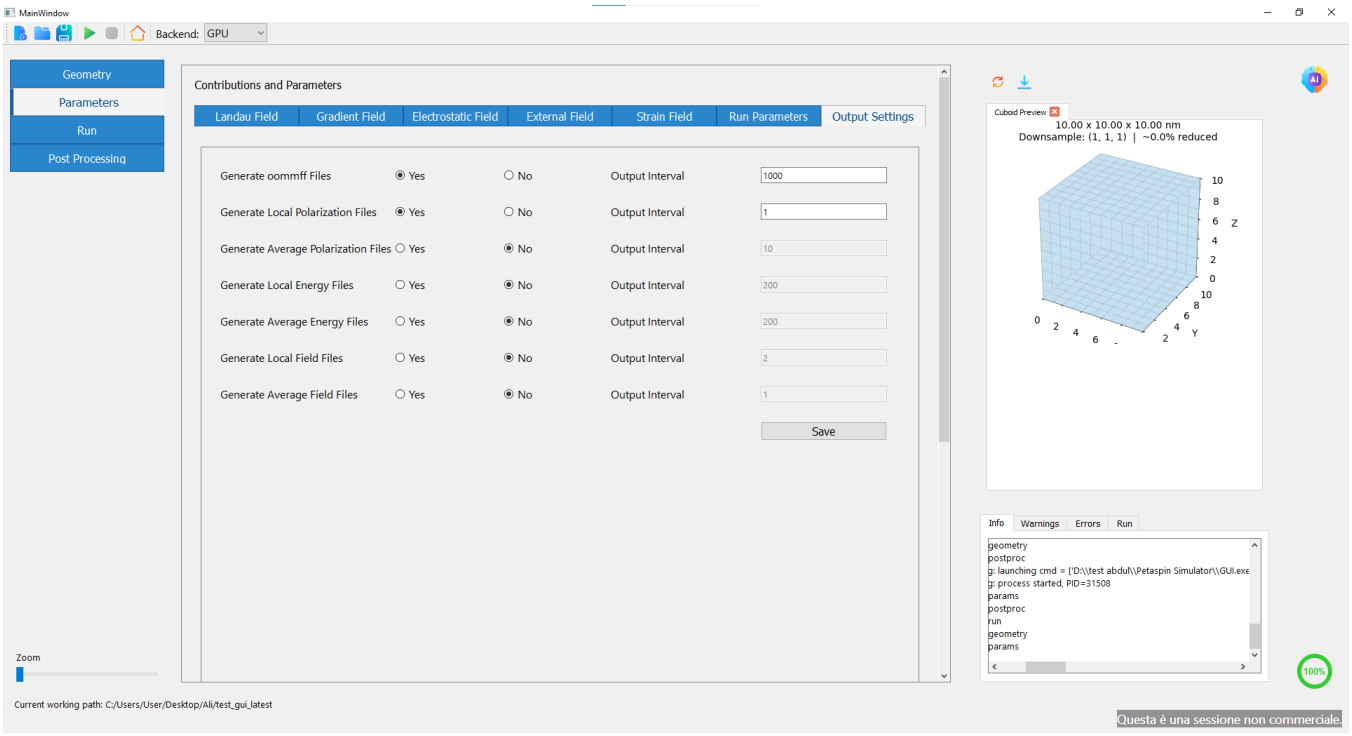


The Run Parameters sub-page

1. Time integrator: Heun / RK45 fixed / RK45 adaptive / AB3M2
2. Run mode: Relaxation (TDGL) or Dynamics (LKT)
3. Initial polarization: Uniform / Thickness / Spatially dependent / Random
4. Random bounds: Per-axis min/max, only active in Random mode

Funded by the European Union

Output Settings



The Output Settings sub-page

1. OOMMF / polarization: Local and average polarization output
2. Energy output: Local and average energy, each with an interval
3. Field output: Local and average effective field

Funded by the European Union

Run Simulation Page

The Run page controls the actual solver execution.

Backend Selection

The CPU / GPU dropdown in the top toolbar controls which executable will be launched:

- GPU (default): runs microelectric_gpu.exe. Requires an NVIDIA GPU and up-to-date CUDA drivers.
- CPU: runs microelectric_cpu.exe. Runs on any 64-bit CPU; slower for large problems.

The choice can be changed at any time before pressing Play.

Starting a Simulation

Click Play in the toolbar (or the Run button on the Run page). The application will:

- Check whether any unsaved changes exist. If so, a dialog asks whether to Save All and continue.
- Run the parametric study orchestrator to expand any studies into individual configuration files.
- Clear all log tabs.
- Switch the log area to the Run tab.
- Launch the selected solver executable with the project directory as the working directory.
- Stream the solver's stdout into the Run tab as it runs.

Progress Indicator

The right side panel shows a circular progress bar that:

- Displays yellow while running.
- Fills from 0% to 100% as the solver reports progress (parsed from [XX%] markers in the solver output).
- Turns green when the simulation finishes successfully.
- Turns red if the simulation fails.

Stopping a Simulation

Click Stop in the toolbar at any time to terminate the running solver process.

Funded by the European Union

Post-Processing Page

The Post-Processing page visualizes simulation output.

Draw Final Polarization 3D Graph

The Draw Final Polarization 3D Graph button loads output/simulation_output_0.txt from the project directory when simulation completes and opens a three-dimensional vector-field plot in a separate window. The plot displays polarization arrows on three orthogonal slice planes (X-Y, X-Z, Y-Z), each equipped with an interactive slider so you can scrub through the volume. Axes are labeled in nanometers, and colors indicate vector-component magnitude.

Live File List

Below the Draw Graph button, a live file list monitors:

```
<project>/output/polarization/localPolarization_device_0/
```

The list refreshes every 5 seconds, showing all polarization snapshot files produced by the solver during and after the run. Files are naturally sorted by their trailing number.

- Before the simulation writes any output, the list shows (waiting for simulation output…).
- Once files exist, they become clickable list entries.

Plot Selected File

Select a file from the list and click Plot selected file to open a three-dimensional visualization of that specific snapshot. This lets you inspect any intermediate state of the polarization field during a running simulation without waiting for it to finish.

Funded by the European Union

Project Files & Data Format

Project Directory Structure

A well-formed project directory looks like this:

```
<project_root>/
├── microelectric_gpu.exe
├── microelectric_cpu.exe
├── file_configuration/
│   ├── ferro_geometry.dat
│   ├── Landau_free.dat
│   ├── elastic_field.dat
│   ├── electrostatic.dat
│   ├── external_field.dat
│   ├── gradient_field.dat
│   ├── initial_polarization.dat
│   └── output.dat
└── output/                    ← created by the solver
    ├── simulation_output_0.txt
    └── polarization/
        └── localPolarization_device_0/
            ├── localPolarization_0.txt
            ├── localPolarization_1.txt
            └── ...
```

The .dat File Format

Each configuration file uses a plain-text line-per-parameter format:

```
1.0e-11    !step size (s)    //Initial time integration step
```

Each line has three sections separated by ! and //:

- Left of !: the parameter value.
- Between ! and //: the parameter's comment key. The GUI locates each parameter by searching for this comment text when it saves changes. It does not rely on line position.
- Right of //: a human-readable description ignored by the solver.

Important: this comment-key lookup only describes how the GUI itself reads and re-saves these files. The solver executable parses every .dat file by fixed line position, not by comment text -- it reads each value in a strict, predetermined order regardless of what the comment says. If you edit a .dat file by hand, do not reorder, insert, or delete lines, or the solver will silently read the wrong value for every parameter that follows -- with no error message. Only change the values themselves, in place, and always prefer using the GUI over manual edits.

You can edit these files directly in a text editor if needed, but the GUI is the recommended interface.

Funded by the European Union

Output File Format

The solver writes plain-text output files, one snapshot per output interval. This allows the time evolution of a run to be inspected, in addition to its final state. Each write of a local (per-cell) quantity produces one file, named with an increasing snapshot index, for example localPolarization_0.txt, localPolarization_1.txt, and so on, one file per output interval elapsed.

Each local output file contains one row per grid cell, with columns *ix*, *iy*, *iz* for the cell indices along *x*, *y*, and *z*, followed by the physical quantity at that cell (three components P_x, P_y, P_z for polarization or field output, one value for energy output). Rows are ordered with ix varying fastest, then iy, then iz, giving Ncx × Ncy × Ncz rows in total, matching the grid size set on the Geometry page.

Average (spatially integrated) output files contain one row per output interval, with two columns, time and the average value of the corresponding quantity. The average is the arithmetic mean over the sample, the sum of the quantity over all cells divided by the number of cells.

Units Reference

All parameter fields in the interface are labeled with their expected unit directly on screen. The table below summarizes the units used throughout, for quick reference.

Quantity	Unit
Length (cell size, padding)	m
Time (step size, total duration)	s
Polarization	C/m²
Electric field	V/m
Temperature	K
Damping coefficient	Ωm
Inertia coefficient	J m² s⁻² C⁻²
Dielectric parameter (εB)	dimensionless
Landau, gradient, and electrostrictive coefficients	as labeled on each parameter field in the interface

Funded by the European Union

Simulation Backends (CPU/GPU)

The Petaspin_microelectrics Solver ships with two backend executables:

GPU Backend (microelectric_gpu.exe)

- Uses NVIDIA CUDA for massively parallel computation.
- Recommended for production simulations with grid sizes above 64^3.
- Requires an NVIDIA GPU with CUDA compute capability 6.0 or higher and up-to-date NVIDIA drivers.
- Substantially faster than the CPU backend on typical problem sizes.

CPU Backend (microelectric_cpu.exe)

- Runs entirely on the CPU using OpenMP parallelization.
- Recommended for small test runs, debugging, or systems without a compatible GPU.
- Works on any 64-bit Windows machine without additional drivers.

Switching Between Backends

Use the Backend dropdown in the top toolbar. The choice takes effect the next time you click Play; no restart is required.

If the selected backend executable is missing from the project directory (for example, if you deleted one after project creation), the GUI displays an error dialog when you attempt to run and suggests switching to the other backend.

Funded by the European Union

Tips & Best Practices

- Save regularly using the toolbar Save button. It writes all pages at once.
- Test on small grids first. Set Ncx = Ncy = Ncz = 10 and use the CPU backend for quick sanity checks before scaling up.
- Using the GPU backend is recommended and use CPU backend for the on machines without a supported NVIDIA GPU.
- Preview geometry before running. Click Refresh Preview on the Geometry page to confirm the shape and dimensions look correct.
- Check the log area at the bottom of the window regularly. It shows info, warnings, errors, and the live solver output on separate tabs.
- Do not edit .dat files while the solver is running: the solver reads them at startup, and changes made during a run will not take effect.
- Live polarization snapshots: to see snapshots appear in the Post-Processing list, ensure that Create local polarization files is enabled on the Output settings page.

Funded by
the European Union

Troubleshooting

The GPU backend crashes immediately

Symptoms: the process exits with code 0xC0000409 or STATUS_STACK_BUFFER_OVERRUN shortly after starting.

Likely causes:

- Your machine does not have a compatible NVIDIA GPU.
- NVIDIA drivers are outdated or CUDA runtime DLLs are missing.
- Grid dimensions are so large that the GPU runs out of memory.
- A file from the output directly is opened anywhere on the machine.

Fix: switch the toolbar Backend dropdown to CPU and try again. If CPU runs but GPU does not, update your NVIDIA drivers to the latest version.

"microelectric.exe not found in project folder"

The project directory is missing one of the solver executables. Options:

- Manually copy microelectric_gpu.exe and/or microelectric_cpu.exe from C:\MicroElectricTemplate\ into your project folder.
- Or create a fresh project via New and copy your customized .dat files into it.

KaTeX / QtWebEngine Warnings at Startup

Messages such as Unable to move the cache: Access is denied or Shader Cache Creation failed are harmless. They come from the KaTeX rendering engine and do not affect functionality. They usually appear when a previous instance of the GUI is still holding the browser cache open. Close all Python/GUI processes and reopen the application to clear them.

"Please enter all three values" (Initial Polarization)

When Uniform is selected as initial polarization, all three components (pol_x, pol_y, pol_z) must be filled in before saving. Fill in all three or leave all three blank (in which case the setting is skipped and the flag is not changed).

Simulation Progress Never Advances

If the progress bar stays at 0% but the process is still running, the solver may not be writing [XX%] markers to stdout for that particular configuration. Watch the Run log tab for other output messages to confirm the solver is actually working.

Still Need Help?

For further information or questions about the solver, please contact ali.hassan@unime.it.

Funded by the European Union

Example

To help new users get started, this configuration is included as the default file_configuration shipped with the solver. Running the executable straight after installation, without changing any parameters, reproduces the result shown below. The exact output folders, containing all local_polarization_*.dat files for the polarization field, will be provided separately for the CPU and GPU runs, alongside the README, so that users can compare the values obtained on both backends.

Example: Spontaneous Domain Formation in PbTiO3

This example demonstrates spontaneous ferroelectric domain formation, starting the solver from a random initial polarization and letting it relax freely, with no external field, elastic coupling, or applied stress. It also serves as a working demonstration of the random initial-polarization mode (Section 6, Run Parameters).

Setup

Material	PbTiO3 (Parameters are summarized in Table 1)
Grid	50 × 50 × 50 cells, 1.0 nm cell size (50 × 50 × 50 nm domain)
Run mode	Relaxation (TDGL)
Initial polarization	Random, uniform in [-1.0×10^{-3}, 1.0×10^{-3}] C/m² per component (fixed seed for reproducibility)
External field	Off
Elastic coupling	Off
Thermal noise	Off
Time step / duration	1.0×10^{-11} s / 1.0×10^{-6} s
Temperature	300 K

Table 2. Material parameters used in the simulations for $PbTiO_3$ in the example.

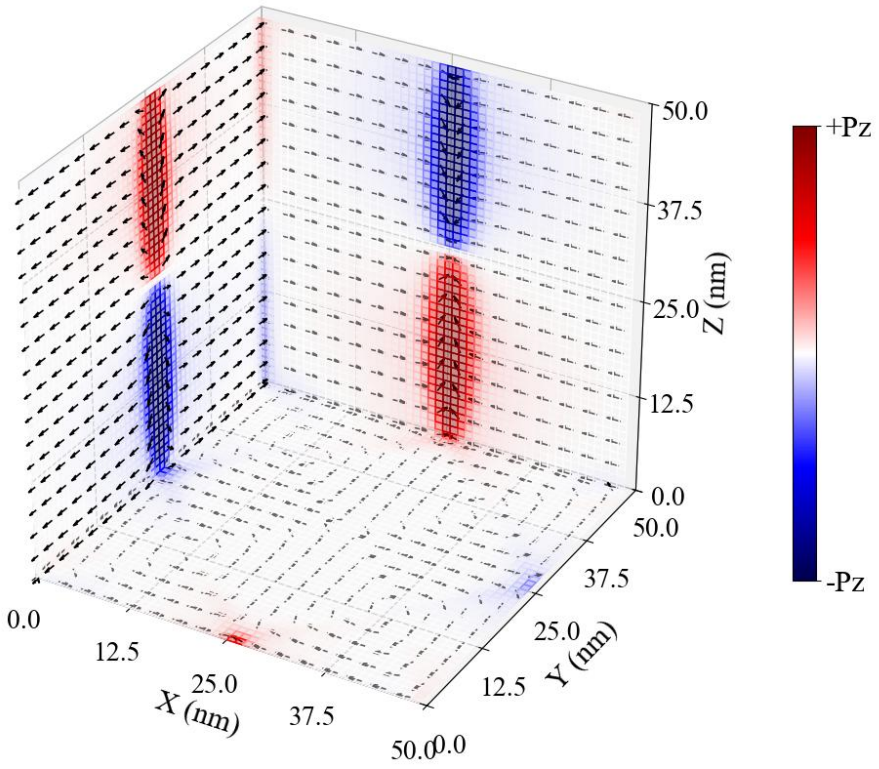
Parameter	Unit	$PbTiO_3$
$\alpha_1 = \alpha_0(T - T_0)$	Nm^2C^{-2}	$3.85 \times 10^5(T - 752)$
α_{11}	Nm^6C^{-4}	-7.3×10^7
α_{12}	Nm^6C^{-4}	7.5×10^8

Funded by the European Union

α_{111}	$\mathrm{Nm^{10}C^{-6}}$	2.6×10^{8}
α_{112}	$\mathrm{Nm^{10}C^{-6}}$	6.1×10^{8}
α_{123}	$\mathrm{Nm^{10}C^{-6}}$	-3.7×10^{9}
G_{11}	$\mathrm{Nm^{4}C^{-2}}$	3.46×10^{-10}
G_{12}	$\mathrm{Nm^{4}C^{-2}}$	0.0
G_{44}	$\mathrm{Nm^{4}C^{-2}}$	1.73×10^{-10}
G'_{44}	$\mathrm{Nm^{4}C^{-2}}$	1.73×10^{-10}
G_{110}	$\mathrm{Nm^{4}C^{-2}}$	1.0
ϵ_B	-	6.0
L	$\mathrm{s^{-1}m^{-2}C^{2}N^{-1}}$	1.0

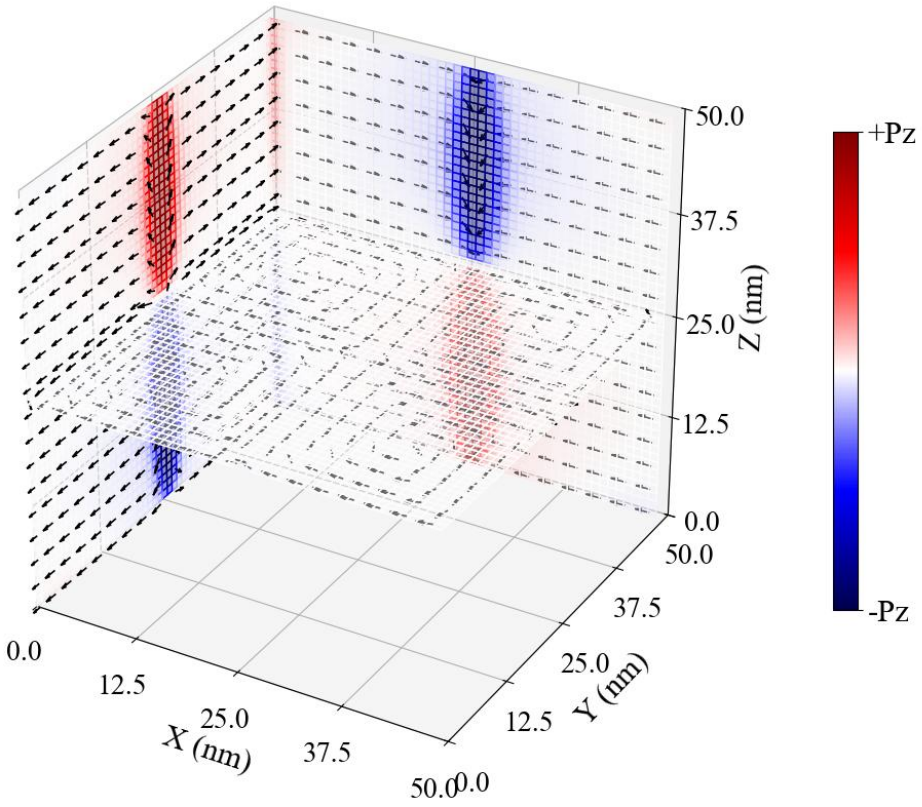
Result

With no applied field or elastic coupling to bias the outcome, the random initial polarization relaxes into four periodic 180° polarization domains, alternating between $+P_z$ (red) and $-P_z$ (blue), running the full height of the sample. Viewed from the top, down the z-axis, each x-y cross-section shows four vortex-like in-plane polarization patterns, consistent with flux-closure structures forming at the domain boundaries.



Final polarization state: isometric view, showing the four alternating $+P_z/-P_z$ domain columns in bottom z-layer.

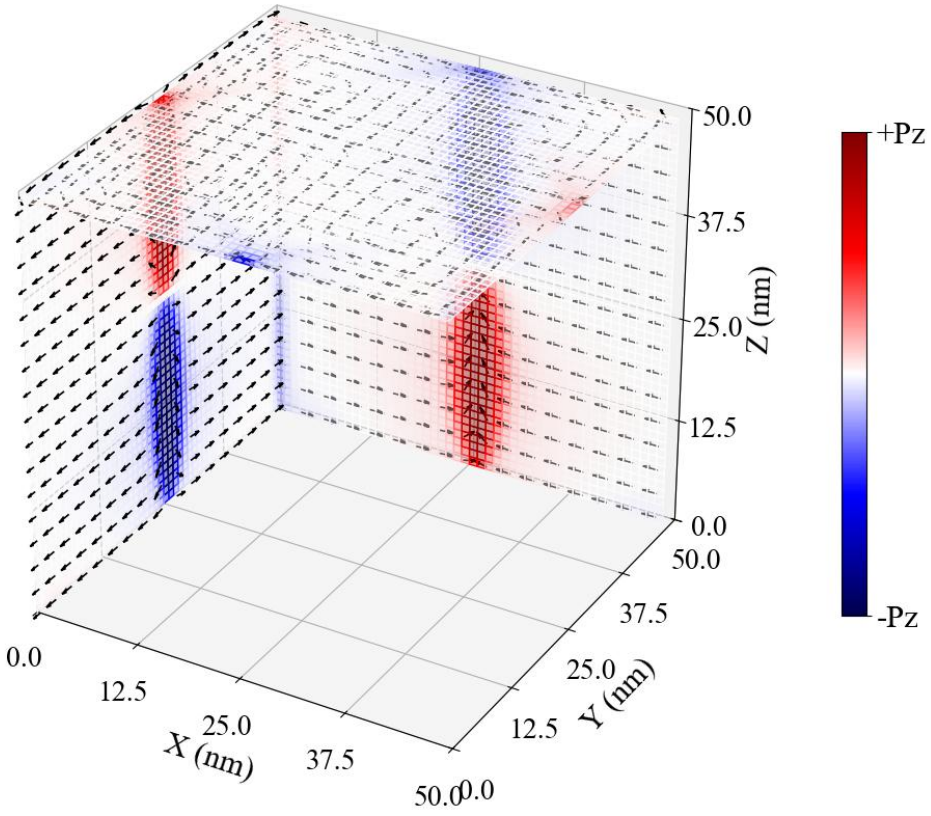
Funded by the European Union



Same result, showing the vortex-like in-plane pattern in the middle z-layer.

.

Funded by the European Union



Same result, top z-layer.

This is a purely relaxational (TDGL) result: no field, stress, or noise was applied after the random start, so the four-domain pattern and its vortex-like cross-sections are the solver's own spontaneous free-energy minimization, not an imposed outcome.

Authorship and Acknowledgement

Author

Ali Hasan, PETASPIN group, Department of Mathematical and Computer Sciences, Physical Sciences and Earth Sciences, University of Messina, Messina, Italy

Supervision

Prof. Giovanni Finocchio and Prof. Davi Rodrigues

Funded by the European Union

Acknowledgement

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie Doctoral Network TOPOCOM, Grant Agreement No. 101119608.

For questions or further information, please contact ali.hassan@studenti.unime.it , davi.rodrigues@poliba.it , giovanni.finocchio@unime.it

How to Cite This Code

If you use this solver or publish results obtained with it, please cite the following work.

[1] A. Hasan et al., A GPU-based Solver for Polarization Dynamics in Ferroelectric Materials, arXiv:2605.28800 (2026).

Funded by the European Union