NMLSim 2.0

User Guide

Written by Lucas Augusto Lascasas Freitas

Belo Horizonte, MG, BR – 06/05/2020

Introduction

NMLSim 2.0 is a CAD and simulation tool for in-plane nanomagnetic logic. The simulator has two engines, a behavior one and an LLG equation-based one. For the LLG engine, there are two methods to solve the equation, which are the RK4 and the RKW2.

The RK4 process is more precise and more reliable since it was properly compared with OOMMF. The RKW2 is an experimental mode, which allows the simulation of circuits with thermal noise involved. This simulation mode was compared with results in the literature.

The simulator has the following dependencies. Try to install the same versions or compatible ones to avoid complications.

- g++ (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609
    - sudo apt-get install g++
- GNU Fortran (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609
    - sudo apt-get install gfortran
- Python 3.5.2
    - Required only for the chat plotting script
    - sudo apt-get install python3
- Python 3 matplotlib library
    - Required only for the chart plotting script
    - sudo apt-get build-dep python-matplotlib
    - sudo apt-get install python3-matplotlib
- GNU terminal
    - Gnome terminal usually comes with ubuntu
- ImageMagick 6.8.9-9 Q16 x86_64
    - Required only for the gif generator
    - https://imagemagick.org/script/install-source.php

This document is organized as follows:

How to compile and run NMLSim 2.0

In the main folder of the tool, you will find a makefile code. This makefile contains five options listed and explained below:

make

This option compiles each module and link them, creating the executable for the NMLSim engine. You must use this command at least once in your machine.

make clean

This option erases the executable of NMLSim engine, as well as the compiled objects in folder "Objs" and the Fortran files in folder "ModFiles." This option is only needed if you intend to recompile the simulator engine and assure that the new compiled version does not have any of the older builds you made.

make eraseCSV

This option erases all CSV files in folder Files.

make interface32

This option runs the user interface for a Linux (Ubuntu preferred) system with 32 bits.

make interface64

This option runs the user interface for a Linux (Ubuntu preferred) system with 64 bits.

make run

This option runs the simulation engine with the default example file, which can be found in the folder Files, inside the main project folder.

make run input=inputFilePath

This option runs the simulation engine for a specific input file, given by its path. The output will be written in the same path, with the same name and the extension CSV. In case of an existing file, this option adds a "_#" to differentiate files, where # will be a number starting at 2. Note that the input format must be XML, and the structure of this file is detailed in the section 6.

make run input=inputFilePath output=outputFilePath

This option runs the simulation engine for specifics input and output files, given by their respective path. Note that the input must be in XML format, but the output doesn't have to be in the CSV format. We recommend using the CSV format to better integrate with the interface and the chart.py code.

At last, there is a python script called chart.py, which is used to plot simulation results. To use this script outside the user interface, follow this command template:

python3 chart.py -h

The call of the script has an argument parser with more details of usage with “-h” or “--help.”

It takes the following arguments:

- --input: path to the csv file created by the simulator engine.
- --fontsz: an integer number to define the font size.
- --range: two values in quotation marks split by semicolon ‘;’ defining the range of time to plot the precession of the magnetization.
- --magnets: a list of magnets under quotation marks split by semicolon ‘;’ defining the particles to be shown in the chart.
- --cols: the number of columns in the chart image.
- --comps: the components of the magnetization to plot (x, y and z), under quotation marks and split by semicolon ‘;’.
- --mode: two modes to plot chat, where split plots each particle in a chart and comparative plots all of them in the same area.

Simulation Parameters

This section will detail all the configurations for an NML simulation in both engines. How to program these parameters for the engine or in the interface will be further detailed in section 6 and section 7.

Both engines have parameters in common, which are listed and detailed below:

- **Mode:** there are four simulation modes, and some of them have additional parameters
  - **Verbose:** this mode writes the three components of the magnetization for all magnets over the simulation time following an additional parameter called **report step**.
  - **Direct:** this mode writes only the final state for the input and output magnets.
  - **Exaustive:** this mode performs an exhaustive (misspelled the name inside the simulator, for now I'll keep the wrong name and fix it latter) simulation. It generates the truth table for the inputs. This mode reports the initial and final state for inputs and outputs for each combination.
  - **Repetitive:** this mode performs a simulation with the defined configurations repetitively, based on the parameter **repetitions**. This mode is useful to ensure that a circuit has not a lot of variation in the same simulation scenario. Note that for the RK4 method, it will not usually have variation. This mode writes in the file the final state for all magnets among the repetitions.
- **Time Step:** this is the time discretization. The smaller the value, the more precise is the simulation and the longer it will take to simulate. We recommend using a maximum value of 0.005 ns to avoid numerical errors in the method.
- **Simulation Time:** this is the total simulation time in nanoseconds.
- **Neighborhood Radius:** this is the radius of the neighborhood. Magnets with the distance between central points higher than this value will not have their interaction computed.

The LLG engine has additional parameters not used in the behavior engine. These parameters are listed and explained below:

- **Gilbert Damping Constant:** this is the constant in the LLG equation. The value of this constant depends on the material of the magnet. For permalloy, we recommend using 0.05.
- **Temperature:** temperature of the system, only used in the RKW2 method.
- **Saturation Magnetization:** this is the magnetization of saturation of the magnet material. This value is responsible for confining the magnetization in a range –1 to 1. For permalloy, we recommend using 800.000.
- **Spin Angle:** this is the spin angle for the spin hall effect. We recommend using a value of 0.4 for this parameter.
- **Spin Diffusion Length:** this is the diffusion length of the spin hall effect, which depends on the substrate material. We recommend using a value of 3.5.
- **Heavy Material Thickness:** this is the thickness of the substrate. We recommend using 5 nm.

Defining the Clock System

The clock system can be configured in two simple parts, the phases and zones. The two engines have different ways to set the clock system; both will be detailed in this section.

Clock Phases

Behavior

The behavior engine uses an empiric representation for the clock signal. This signal should be confined between 0 to 1, where 0 represents the absence of the external field, and 1 is the presence with maximum power. Each phase must have a distinct name and a duration, as well as an initial and end value for the clock signal. The variation will be constant

LLG

The LLG engine has two modes of modeling clock, which are complementary. There is the Zeeman field and the spin Hall effect.

The Zeeman field is the external field applied over the circuit. To configure this field, the user must define an initial value and an end value for the signal, as well as the duration of the phase. Note that all phases must have unique names.

The spin Hall effect is caused by a current passing through the substrate. To configure this signal, the user must choose initial and end values for the current field. Note that you need to specify the current field and not the current itself.

The Zeeman field can work together with the spin Hall effect or both can work separately. All fields have the three components (x, y, z).

Clock Zones

After all clock phases are defined, it is time to define the clock zones that will use the phases. Both engines are similar in this process.

The user must define the phases that a clock zone will pass through. Zones should not have repeated phases in their definition to avoid errors in the engine. After all phases are done, the first phase will start again in a cyclical repetition.

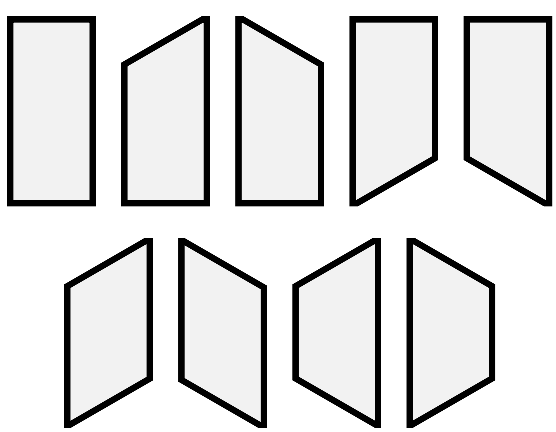
Note that it is possible to define clock phases with different duration. In this case, the synchronization is fundamental for the correctness of the circuit. Let’s say for example, that there are 2 phases A and B, and two zones 1 and 2. The phase A has twice the duration of B. Zone 1 does A then B, and zone 2 does B then A. In this scenario, when zone 2 finishes B, phase A is currently in the middle of its variation. Therefore, zone 2 would start phase A in the middle instead of the beginning.

Magnet Configurations

The magnets are highly configurable in the new version of the tool. Some configuration parameters need to be defined.

- **Id:** each magnet must have a unique Id, which can be anything.
- **Type:** there are three magnets types. Input magnets are used in the exaustive mode to generate the truth table. Output magnets are the ones with the values reported in the direct and exaustive modes. Regular magnets are neither of them.
- **Clock Zone:** the clock zone label. Input magnets don't have to be in a clock zone, since they have fixed magnetizations. However, we recommend putting them in a clock zone so that the interference of the neighbors in them is considered.
- **Magnetization:** the initial magnetization value of the particle. LLG engine uses three components and the behavior engine uses only one.
- **Fixed Magnetization:** weather the magnetization is affected by the clock or not.
- **Geometry:** width, height, and thickness of the particle.
- **Top Cut:** cut in the top part of the magnet. Positive cuts are made from right to left (the right corner will be lower than the top). Negative cuts are the opposite.
- **Bottom Cut:** cut in the bottom part of the magnet. Positive cuts are made from right to left (the right corner will be higher than the top). Negative cuts are the opposite.
- **Position:** position in the substrate grid.

With these configurations, it's possible to have the following shapes:



The XML File

One of the options to execute the new version of the NMLSim is to run the engine directly, without the interface. To use this option properly, there is a need to understand the XML file structure. This file is used to define the simulation parameters. There is an example file in the Files folder, which contains a lot of detailed comments (marked with <!-- -->). This section will use this example file.

Circuit Tag

The circuit tag is composed of all simulation properties. Starting with the fields **technology** (this property must be iNML only for now) and **engine** (this property can be LLG or Behaviour – sorry for the misspelling, it will be fixed in future releases).

There is the property **method** (RK4 or RKW2), that is only considered for the engine LLG. The following property is the **mode** (exaustive, direct, repetitive, and verbose – sorry again for misspelling). Then we have the properties **repetitions** (integer number) and **reportStep** (real number).

There are some parameters only used in the LLG engine. Such as, **alpha** (real number), which is the gilbert damping constant. Also, **Ms** (real number) for the magnetization of saturation. The RKW2 method uses the field **temperature** (real number), which will only be considered in this simulation mode.

The next fields regard time control. We have the **timeStep** (real number) to define the discretization of time. We also have the **simTime** (real number) to define the total simulation time of the circuit.

There are also the fields related to the spin Hall effect, which are the **spinAngle** (real number), **spinDifusionLenght** (real number) and **heavyMaterialThickness** (real number). At last, the **neighborhoodRatio** (real number), which has a misspelling, should be the neighborhood radius.

Clock Phase Tag

The clock phase tag is composed of all phases in the circuit, where each phase is an item tag. Each item needs a unique name, an **initialSignal** (a real value for Behavior engine and a 6 spaces vector for LLG – the vector values must be split with ‘,’), an **endSignal** (similar to the initialSignal), and a **duration** (real number).

The clock signal vector is composed of the three components of the Zeeman field (Zx, Zy, Zz) and the three components of the spin Hall field (Sx, Sy, Sz). The combination of both fields forms the clock signal (Zx, Zy, Zz, Sx, Sy, Sz).

This tag can have as many clock phases as needed.

Clock Zone Tag

The clock zone is composed of all clock zones, where each zone is an item tag. Each tag must have an index (integer number) as name, starting at 0 and following a sequential order, without skipping numbers. The clock zones are sensible to the order in which the phases are inserted.

This tag can have as many clock zones as needed.

Components Tag

The components tag describes the geometry of the particles in the circuit. This tag is composed of several items, which are different geometries. Each geometry has a unique **name**, **width** (real number), **height** (real number), **thickness** (real number), **topCut** (real number), and **bottomCut** (real number). This tag can have as many components as needed. We recommend not creating one component for each magnet to speed up the parsing of the file. Instead, try creating one component for each different geometry.

Design Tag

Finally, the design tag has the information of each magnet in the circuit. Each magnet is an item with a unique **name** as Id. Each item has a **component** (same name of the tag above), **myType** (input, regular or output), **fixedMagnetization** (true or false), **position** (real numbers for x and y, split with ‘,’), **clockZone** (integer number index of a valid clock zone), and **magnetization** (one real number for Behavior engine, and three values vector for LLG engine). The last one is just the initial value of the magnetization.

The **mimic** (a valid magnet Id) is a field used to approximate a signal cross, which is not currently available in the tool. This field links the magnetization of a magnet to another, making both to have the same value over time. Note that you should not make a magnet A mimic B, and make B mimic A at the same time. In this scenario, you will create an infinite loop, since A needs B to compute its magnetization to define his own and vice-versa.

This tag can have as many items as needed, and each item must be a magnet.

The User Interface

This section explains how to use the user interface to simulate circuits. The section is split into subsections that detail different parts of the interface.

Header

The header (Figure 1) has three parts, the **file**, **magnet**, and **substrate**. Each section has specific action buttons for the group. By clicking on the name of the section in the header, it expands showing the name of the buttons. Every button of the interface has an explanatory text if you place the mouse cursor over it for some seconds.

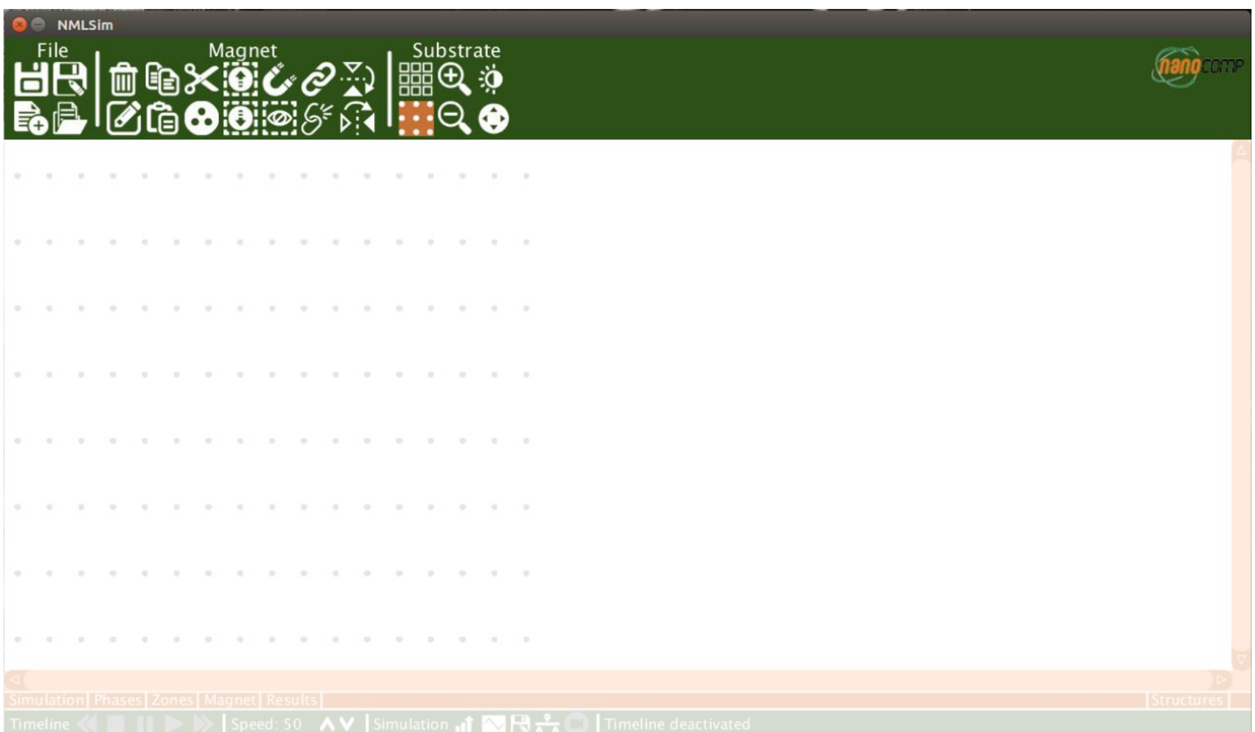


Figure 1. Interface with the Header highlighted.

The buttons of the **File** part (Figure 2) are detailed here:

- **Save:** Save the current project in his save folder. Each NMLSim project will be a folder composed of **nmls** file, a **csv** file (only after a simulation is performed), a **str** file, a **log** file (only after a simulation is performed) and an **xml** file.
- **Save As:** Save the current project in a new folder.
- **New:** Start a new empty project.

- **Open:** Open a previously saved project.

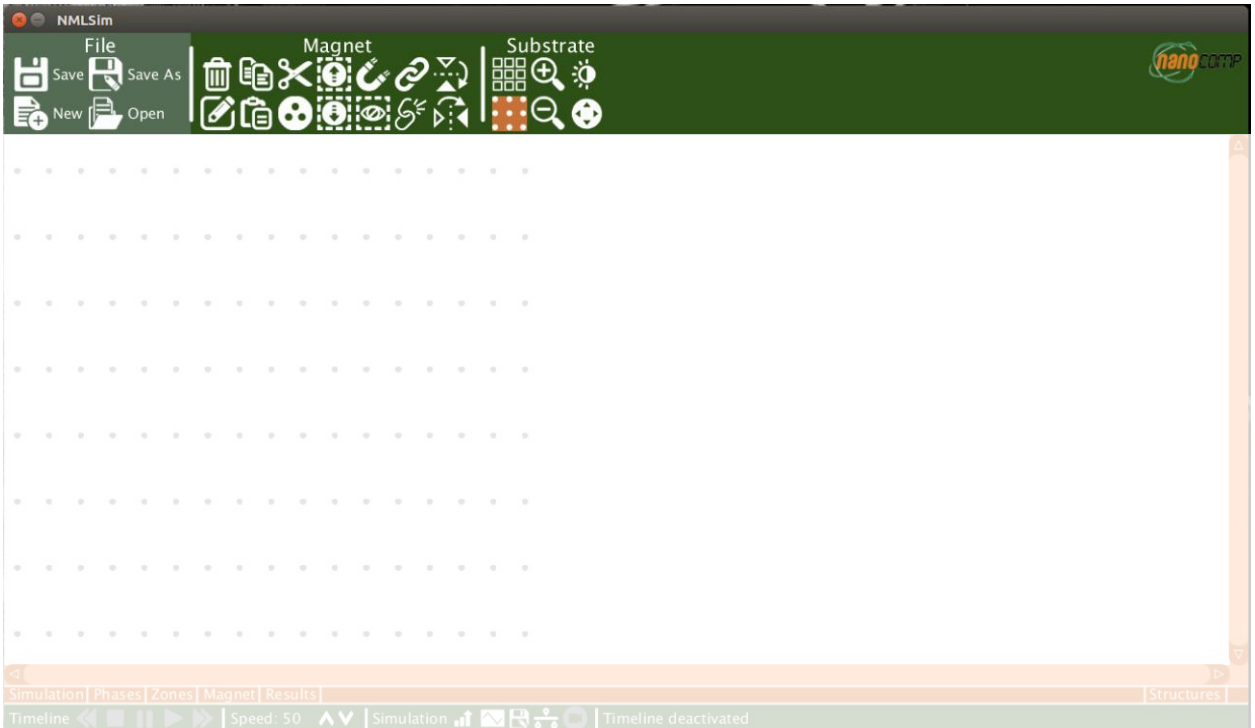


Figure 2. Interface with the <u>file</u> section expanded in the highlighted header.

The buttons of the **Magnet** part (Figure 3) are detailed here:

- **Delete:** remove the selected magnets from the substrate.
- **Edit:** opens the magnet panel (explained in the next subsection) to edit the selected magnet. In the case of multiple selected magnets, it only modifies the first one that was selected.
- **Copy:** copy the selected magnets to the clipboard.
- **Paste:** enables the paste mode, to replicate the structure from the clipboard.
- **Cut:** first copy, and then delete the selected structures.
- **Group:** group the selected magnets, making it easier to select all of them at once. To ungroup, cut and paste the group.
- **Up Zone:** set the selected magnets to the next clock zone relative to their current one.
- **Down Zone:** set the selected magnets to the previous clock zone relative to their current one.
- **Magnetization:** rotate the magnetization of the selected magnet clockwise.
- **Zone View:** toggles the view mode, one of them show the clock zone in the fill area of the magnet; while the other shows the magnetization in the fill and the clock zone in the stroke (Figure 16).

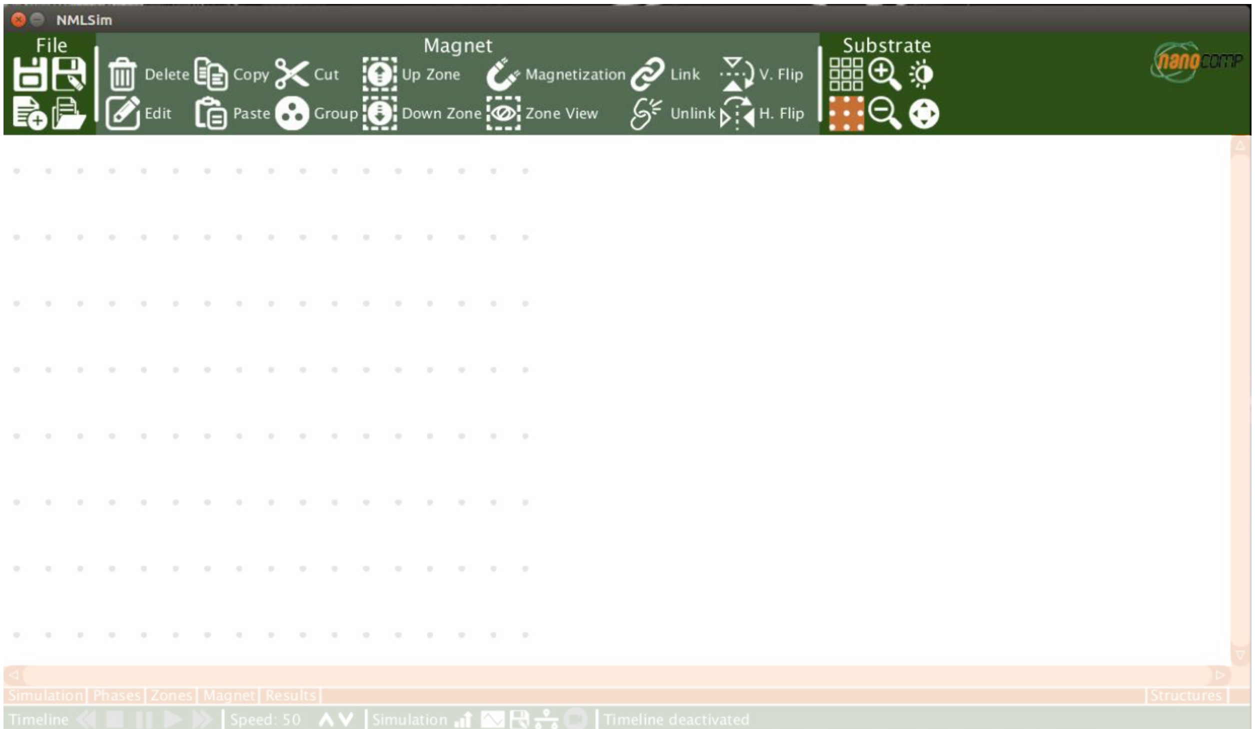


Figure 3. Interface with magnet section expanded in the highlighted header.

The buttons of the **Substrate** part (Figure 4) are detailed here:

- **Grid:** toggles the grid ruler visibility (Figure 14).
- **Bullet:** toggles the visibility of the bullets (Figure 13).
- **Zoom In:** zoom in the substrate.
- **Zoom Out:** zoom out the substrate.
- **Light:** toggles the light view mode (Figure 15).
- **Move:** enables the move mode for the user to move the substrate around using the mouse cursor.
- **Link:** creates a link from the first selected magnet to all other selected magnets. This link is the mimic property of a magnet, which means that all linked magnets have the same magnetization of the source. Only magnets in the same clock zone can be linked. Otherwise, numerical errors will happen generating "nan" (not a number).
- **Unlink:** breaks the mimic link from selected magnets. You only need to select the magnet that is mimicking the other.
- **V. Flip:** Flips a magnet vertically.
- **H. Flip:** Flips a magnet horizontally.

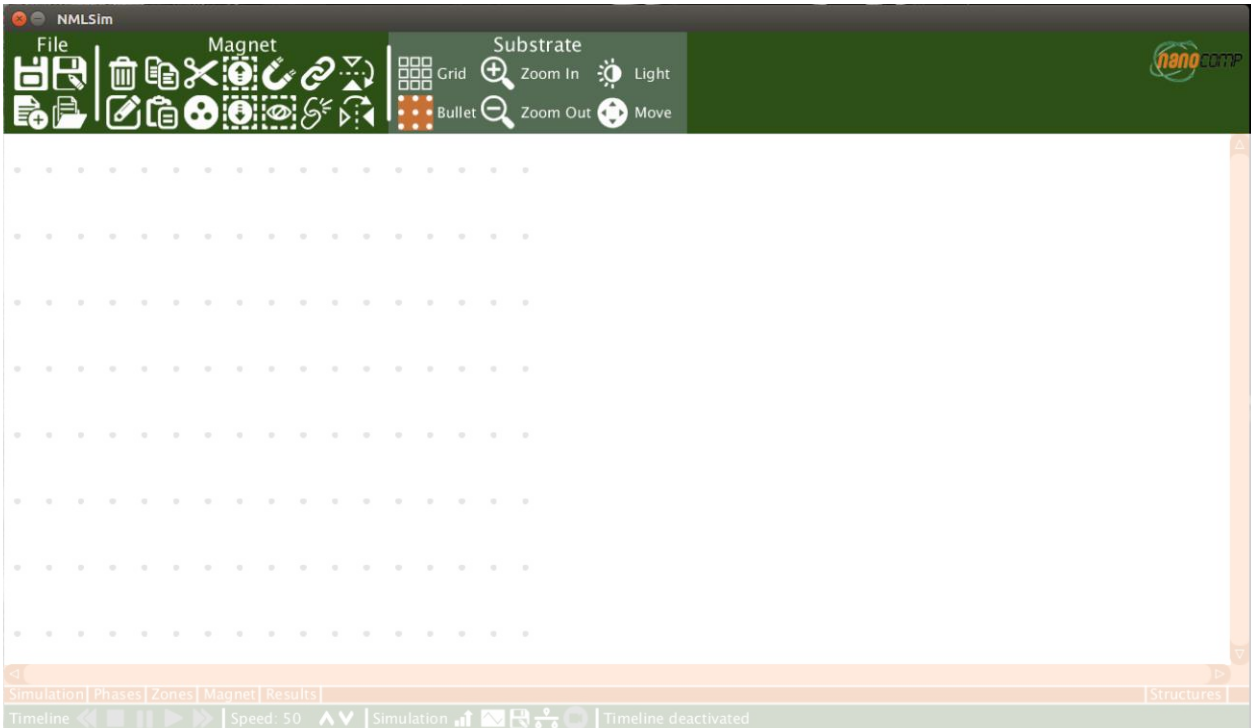


Figure 4. Interface with substrate section expanded in the highlighted header.

Panels

There are six different panels, all located in the bottom part of the interface screen (Figure 5). To expand panels, click in their names in the menu. All panels functionalities are detailed below:

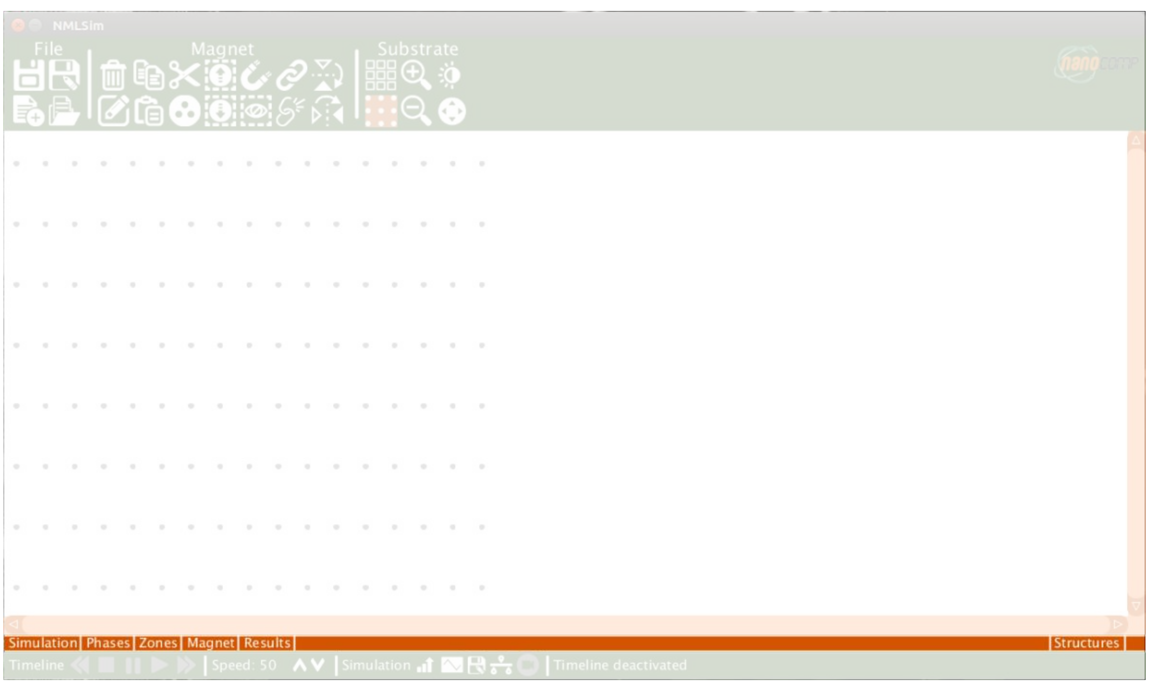


Figure 5. Panel menu highlighted.

Simulation

The simulation panel (Figure 6) is used to load simulation details, such as the engine, the mode, etc. However, this panel is also used to define the substrate size, as well as the reference grid ruler cell size and the distance between bullets. Note that the bullets are used to center the magnets above them. Therefore, if you want a horizontal spacing of 10 nm between magnets, you must add 10 to the width of the magnets to set the bullet spacing.

This panel has two buttons, one to restore the default values to all text fields, and one to clear all fields. One must be careful with these buttons since they erase the information in the panel, not being able to restore that information.

Note that invalid boxes will have dark orange fill and red stroke to indicate that the field must be corrected.

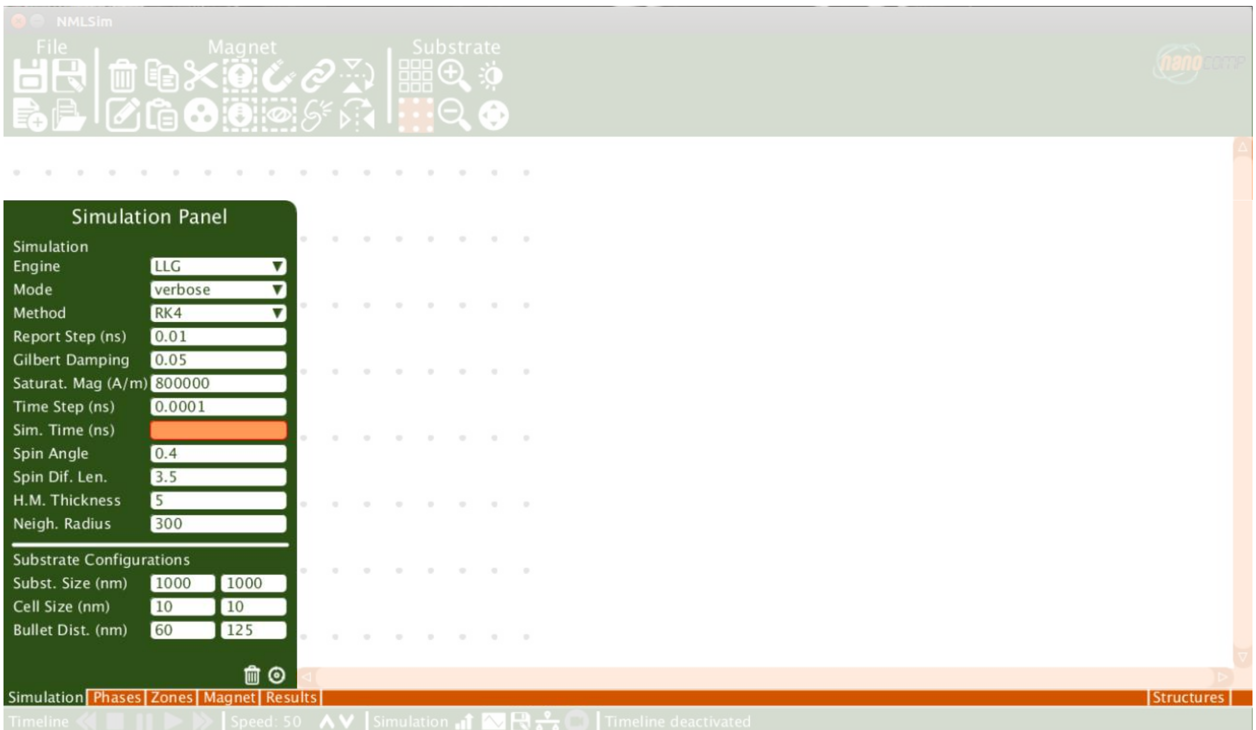


Figure 6. Simulation panel expanded and highlighted.

Phases

The phases panel (Figure 7) is used to define clock phases. Both engines have a similar panel; the difference is the number of text boxes for the signal values. The name of the phase can be anything with the keys that are recognized by the box. The duration must be a real value greater than 0, which is given in nanoseconds.

When all fields are correctly filled, the add button is enabled to create a new clock phase. If the phase has a repeated name, this button is replaced with the save button to save the modifications made. The trash button on top deletes the information from the fields.

In the second section, there is a list of all phases created. Each phase has an edit and a delete button associated. The edit button loads the information of the phase in the text boxes above. The delete button removes the phase from the simulation, also removing any clock zone that used that phase.

The last section is a chart of the clock signal over time. To display the legend of the chart, put the mouse over any part of the chart.

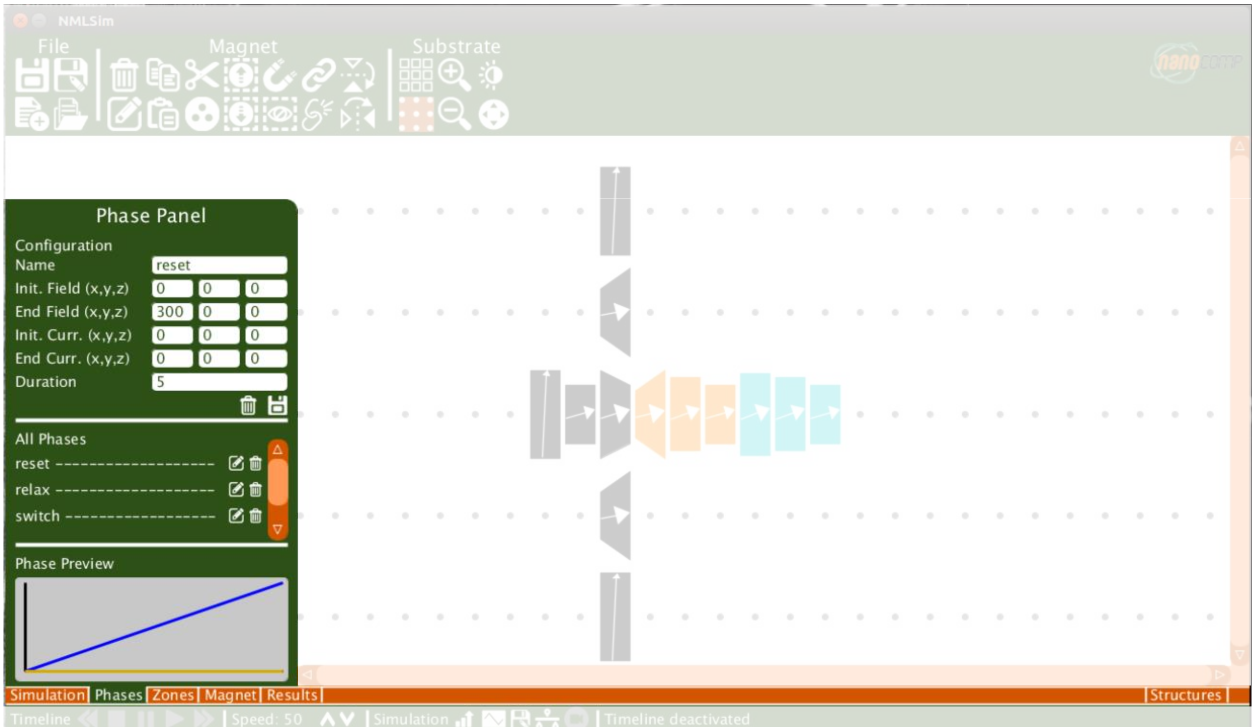


Figure 7. Phases panel expanded and highlighted.

<u>Zones</u>

The zones panel (Figure 8) is used to define clock zones. Both engines have similar panels, where the difference will be the chart. In the top part, there is the zone name, which can be anything (it will be converted into indexes to simulate the circuit). It also has the clock phase drop-down menu with an add button, allowing the user to add the phases in any desired order. At last, there is a color pallet for the user to define a visual representation of the clock zone.

In the top section, there is a list of phases that the current zone uses. This list has options to delete the phases from a zone or reorder with the up and down arrows. Once at least one phase is added and a valid label is entered, the new button will be available to create a new zone. If the zone has a repeated name, this button turns into the save button to save current changes.

In the middle section, there is a list of all clock zones in the project. Similar to the phases, this list has edit and delete buttons.

In the bottom section, there is a preview chart that shows the clock signal over time, like the phases panel.

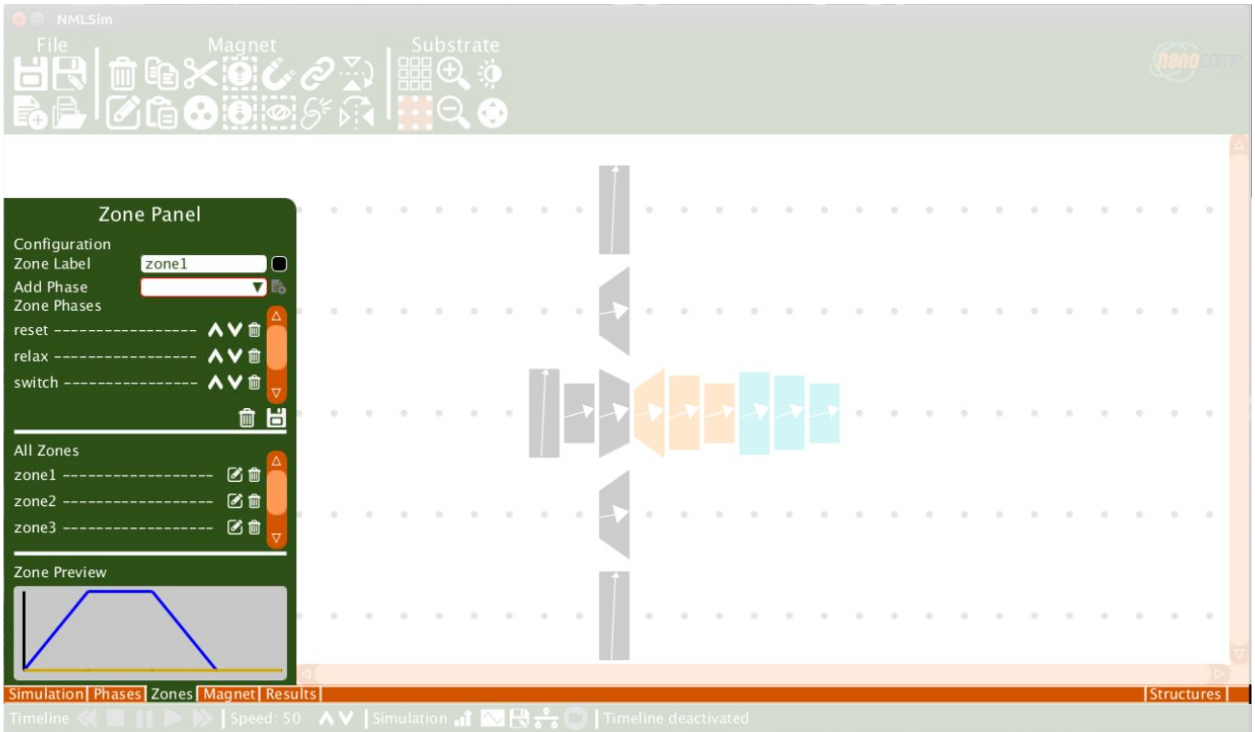


Figure 8. Zones panel expanded and highlighted.

<u>Magnet</u>

The magnet panel (Figure 9) is used to create or edit magnets. In the first section (in the top), there are the magnet id, type and clock zone. Note that if you add a magnet with the save Id of a current magnet in the circuit, it will replace the existing one.

The second section has magnetization information, the initial magnetization value and the flag for fixed magnetization.

The third section has information regarding the geometry of the particle and its position in space. All values must be positive, except for the cuts.

The last section has a preview of the magnet shape.

After all fields are filled with valid values, the add button is enabled, allowing the user to add the current magnet to the substrate. Note that if the position does not allow the magnet to be placed, the same won't be placed. The save as template button is also enabled, allowing the user to save the current magnet configuration as a template to replicate with ease. The trash can button clear all fields of the panel.

Once a magnet is selected and the edit button in the header is pressed, this panel enters the edit mode (Figure 10). In this mode, the add and save buttons are replaced with save and cancel buttons. The save button commits the changes in the magnet, whereas the cancel button ends the edit mode without changing the magnet. Note that if you put the magnet in an invalid position, the same will be removed from the circuit. While in editing mode, the substrate grid is locked from being used.

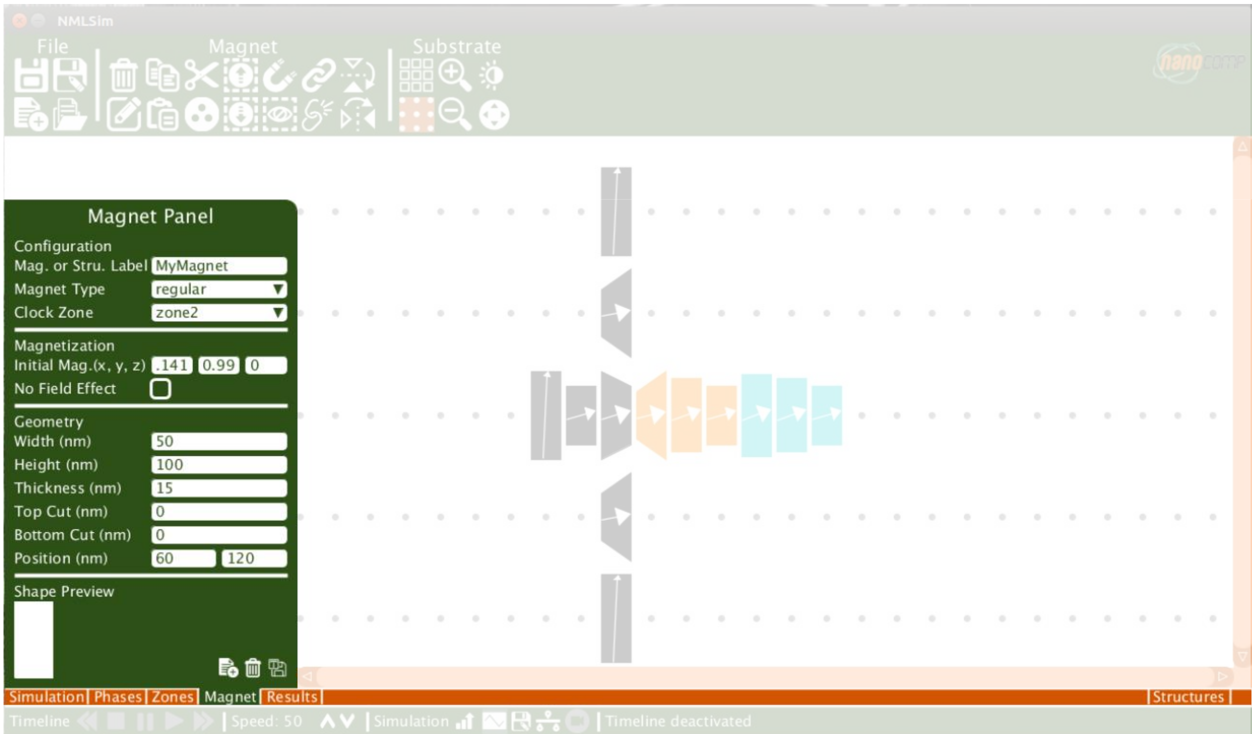


Figure 9. Magnet panel expanded and highlighted.

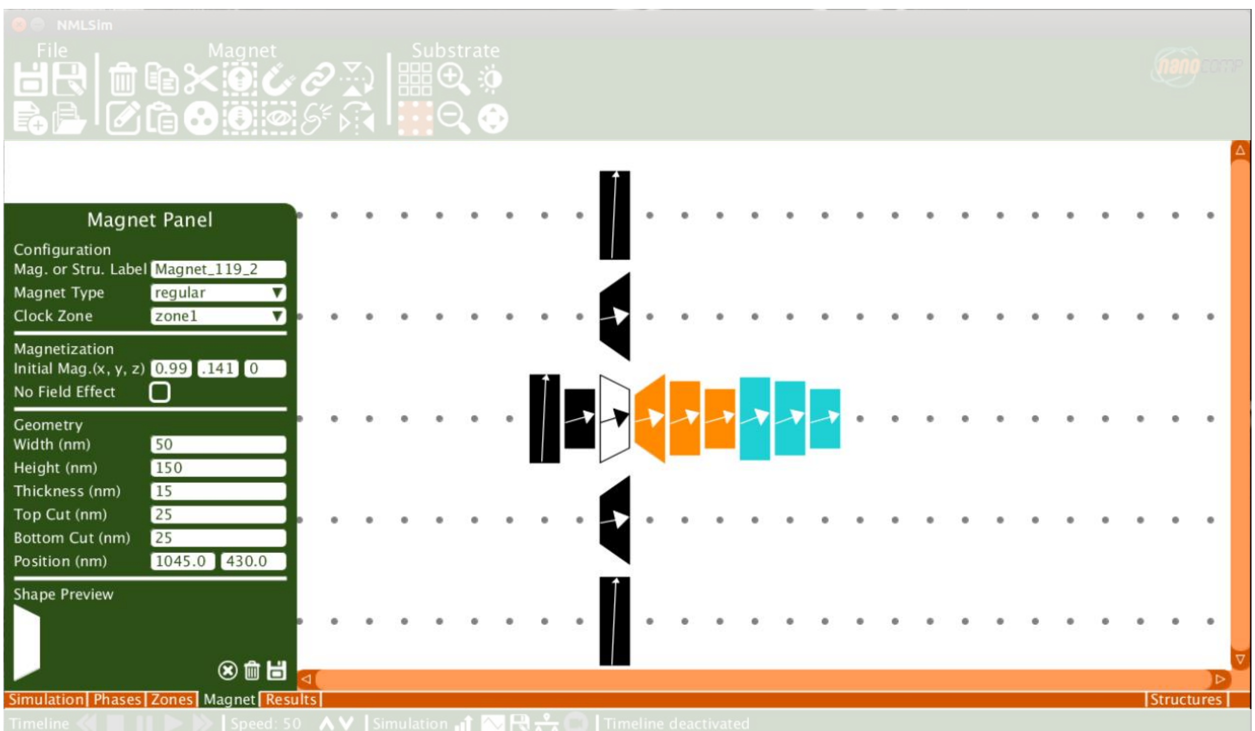


Figure 10. Magnet panel in editing mode, highlighted with the substrate grid.
Note the selected magnet in the substrate grid with a white fill.

Results

The results panel (Figure 11) allows the user to configure the options for the python script that plots the precession chart. There is the plot mode, which can be "comparative" to plot all magnets in the same chart or "split," which plots them separately. The user can also define the font size of the chart image. The range and number of columns are optional, where the user can leave the default option by not marking the box. Default values for start and end of the range are the initial time and end time of simulation, respectively. The default for the columns is automatic, where it keeps a maximum of 5 charts by column. Finally, the user can also chose which components of the magnetization will be plot in the chart.

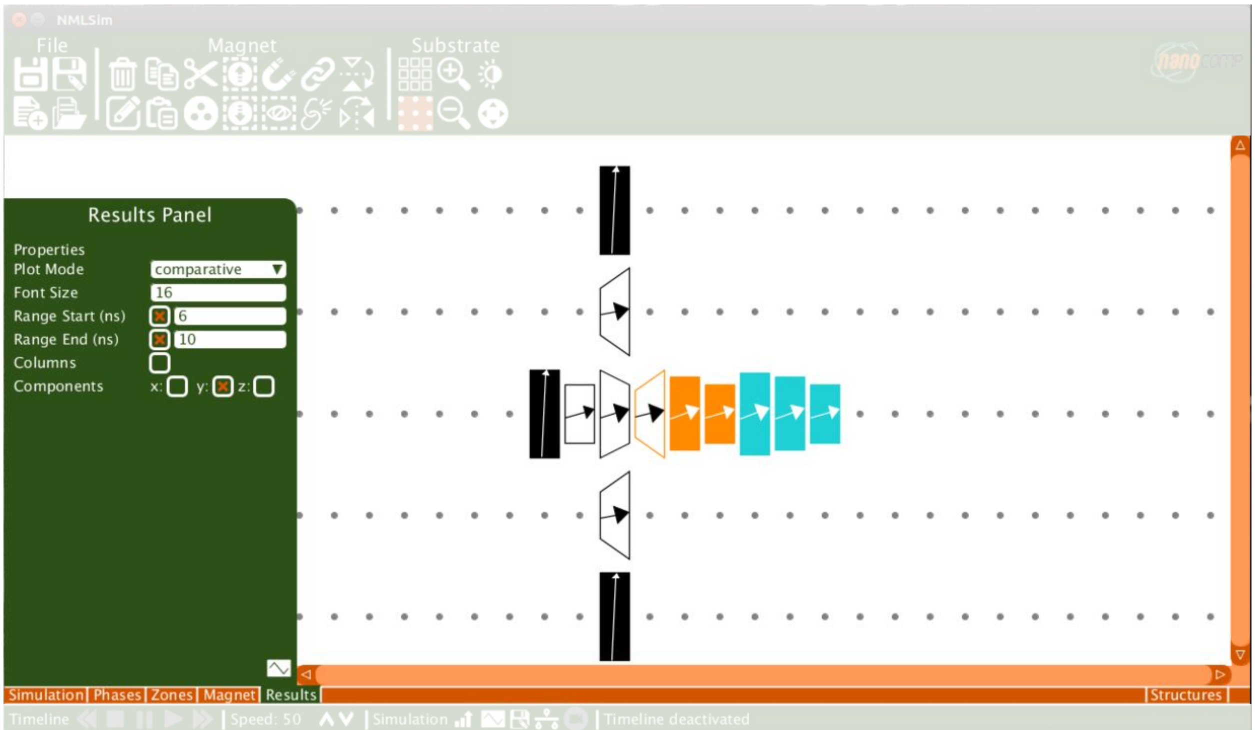


Figure 11. Results panel highlighted with the substrate grid.
Note selected magnets in the substrate grid with white fill.

Structures

The structure panel (Figure 12) has all the structures saved as templates. If you click on the name of a structure (which can be a single magnet or a group of them), the button will be selected, and the adding mode will be enabled. To disable the adding mode of a structure, click its name in the panel again.

In the bottom, there are three buttons related to the structures. The first of them is the import button, which allows the user to import structures from other projects by selecting the other project folder.

The second button is the save as template button. Once the user clicks this button, the selected magnets are going to be added as a new structure with a default name. The same structure can be added several times, although this is not recommended, since it uses more memory.

The last button is the edit button, which switches the panel to the edit mode (Figure 13), adding a delete button for each structure. By clicking a structure during edit mode, the button will be selected, allowing the user to redefine the name of the structures. The edit button is replaced with the save button, which commits the changes made in the structures panel. Note that if you leave a structure with no name, a random name will be assigned to it.

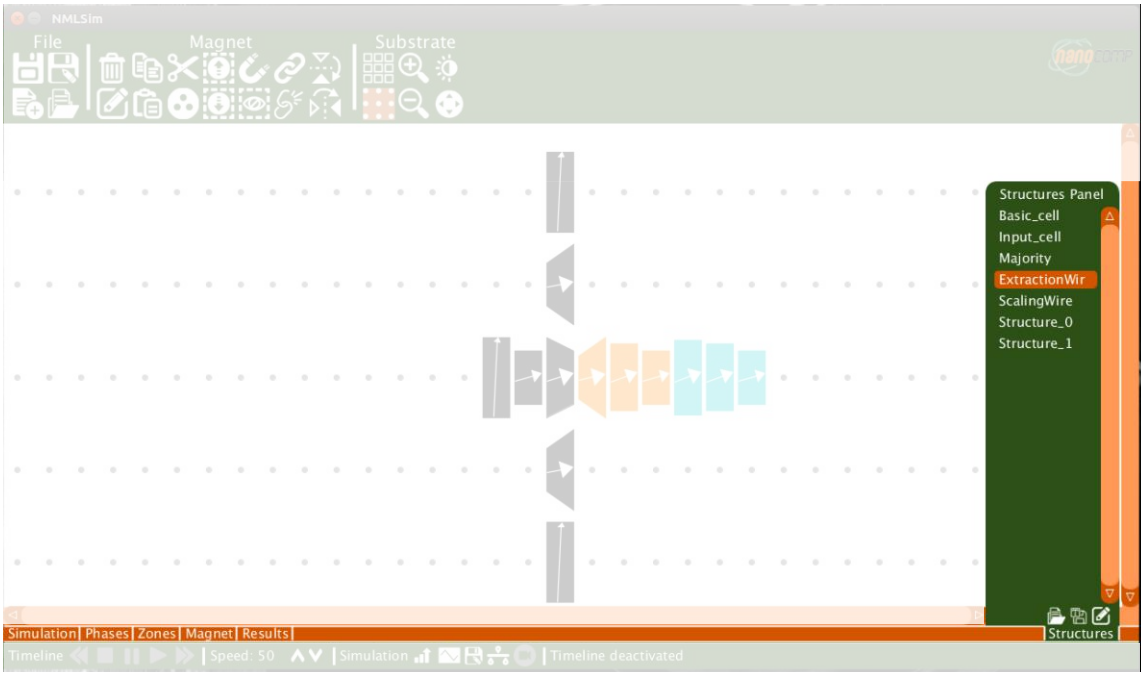


Figure 12. Structure panel expanded and highlighted.
Note the selected structure (ExtractionWire) in orange.

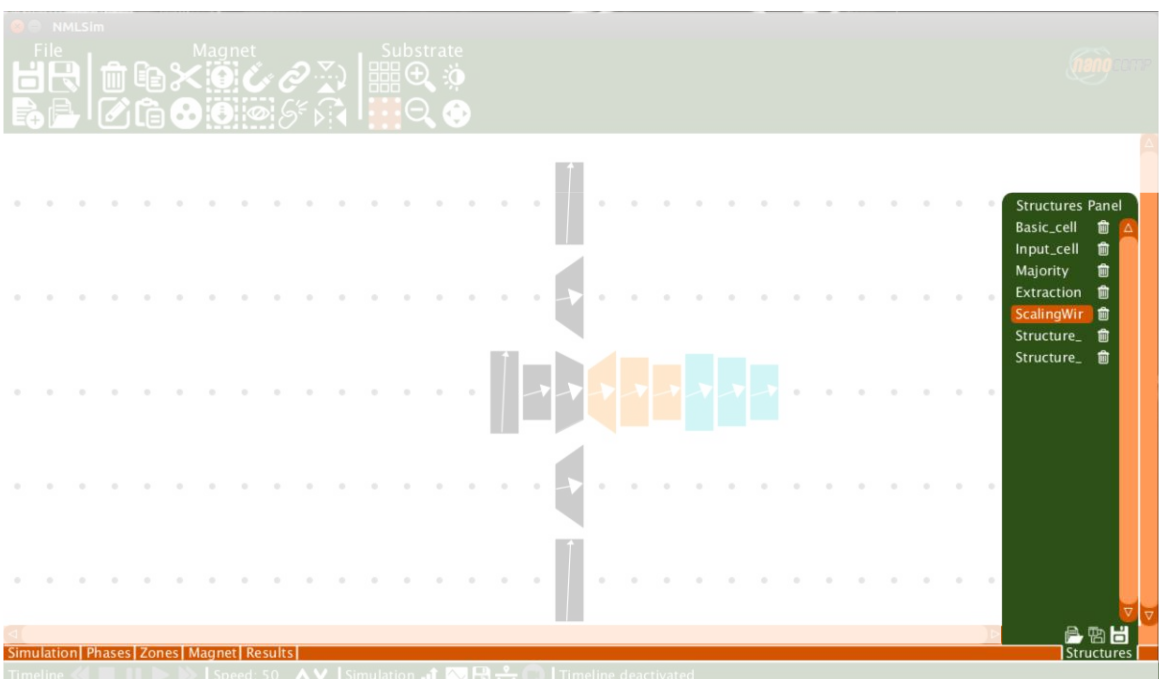


Figure 13. Structure panel in editing mode, highlighted.
Note the selected structure (ScalingWire) in orange.

Substrate

The substrate grid is the main workspace in NMLSim 2.0 (Figure 14). There is an option to activate a personalized ruler just for reference (Figure 15). This area has two light modes: dark and white (Figure 16). There are two scroll bars to move the substrate grid around. The bullet system is used to center the magnet on top of the bullet. Once this system is deactivated, the user can place the magnets and structures freehanded.

To select a single magnet, press on top of it with the movement mode disabled. To unselect magnets, you need to press on an empty space in the grid. To select multiple magnets, press SHIFT and keep it pressed while clicking on the magnets. To create a selection box, that select all magnets that overlay it, press CTRL and drag n' drop the mouse cursor. Note that by pressing CTRL and SHIT, you can create a selection box while keeping the selected magnets in the selection. Also, note that by using CTRL to select magnets, it ignores groups of magnets.

The substrate grid has two view modes, one where the magnet's fill represents its clock zone and one where it represents its magnetization status (Figure 17).

Note that once you press the edit button on the header, the edit mode is enabled, locking the substrate grid. Also note that if panels cover a part of the grid, it does not function.

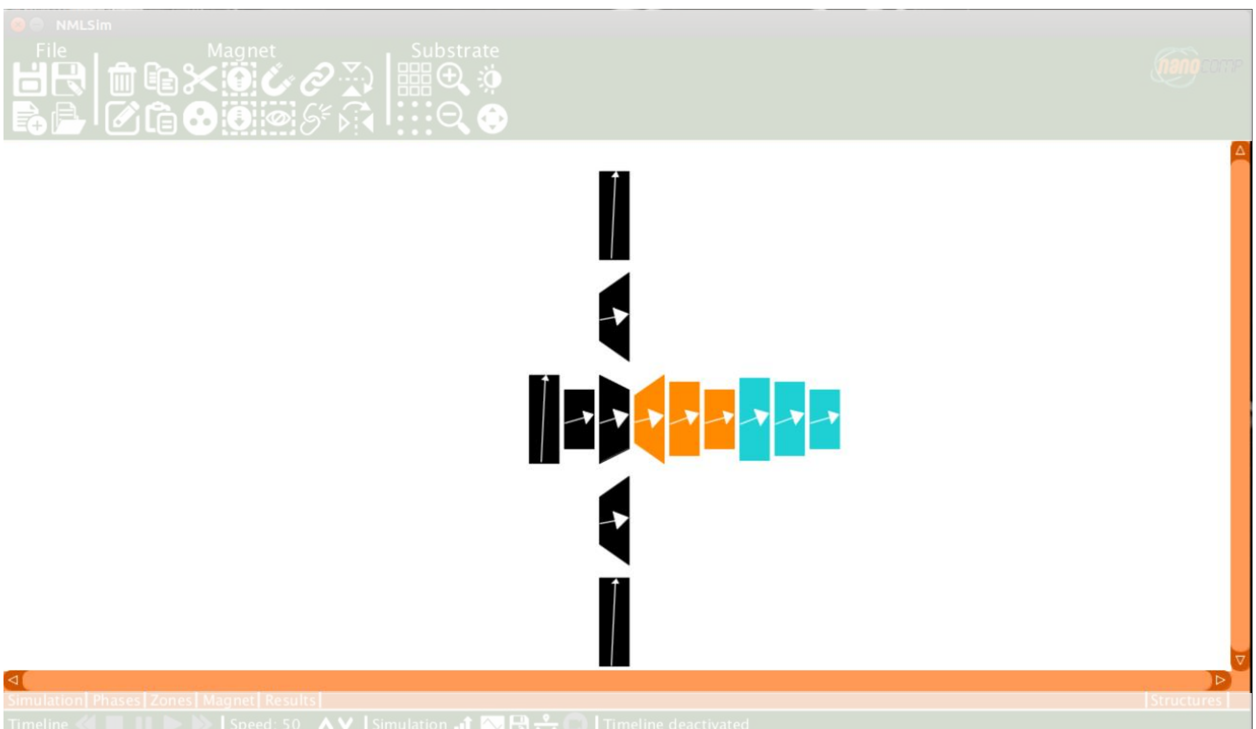


Figure 14. Substrate grid workspace highlighted.

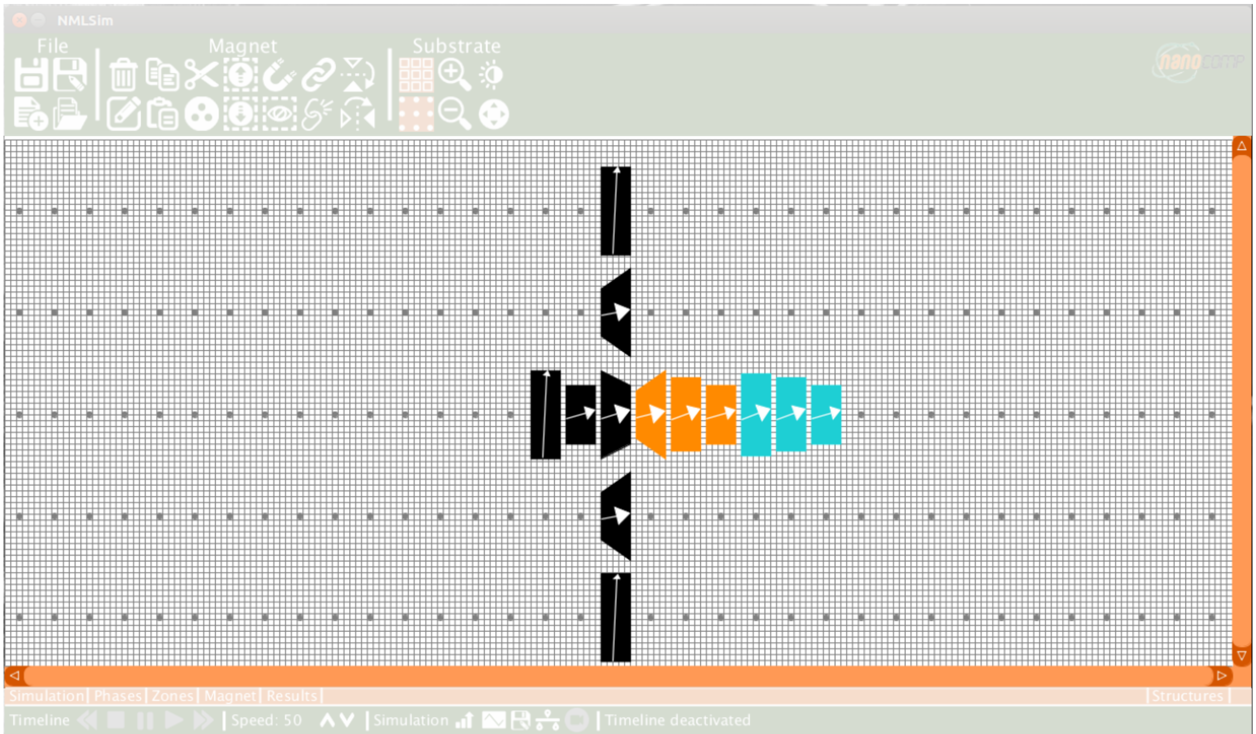


Figure 15. Substrate grid with active bullet references and rulers, highlighted in the figure.

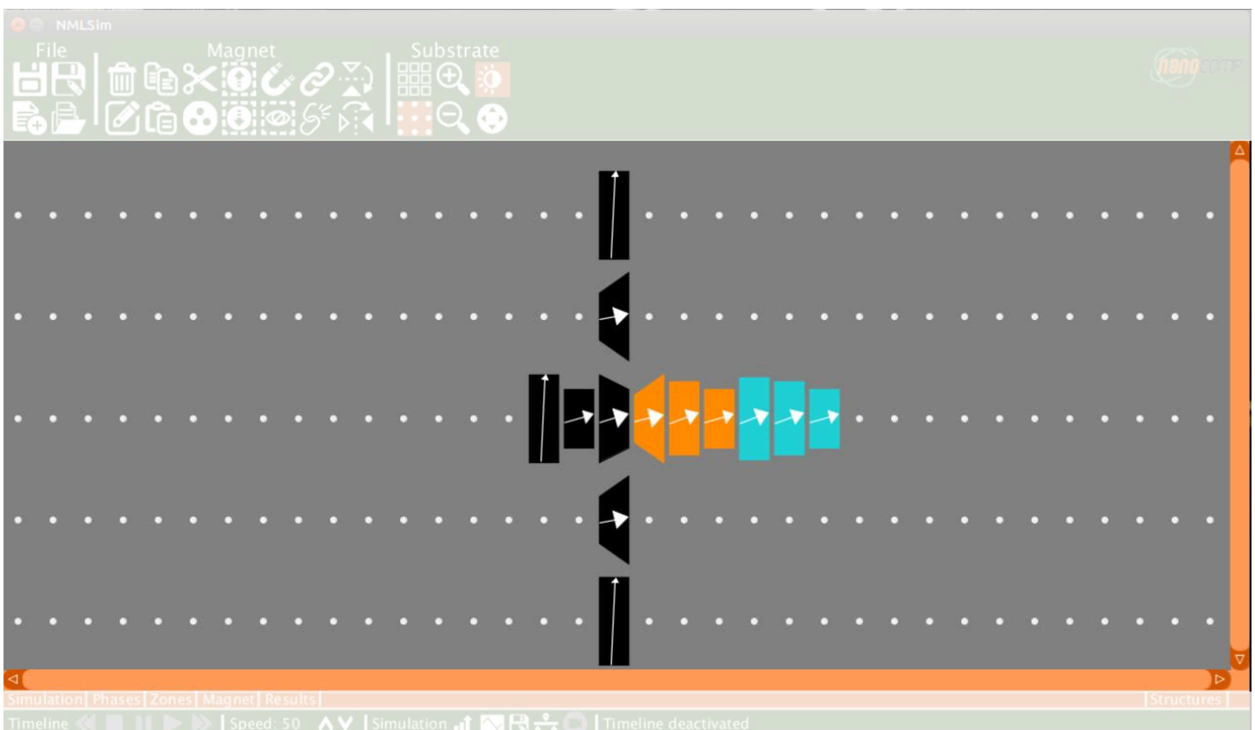


Figure 16. Substrate grid in dark mode, highlighted in the figure.

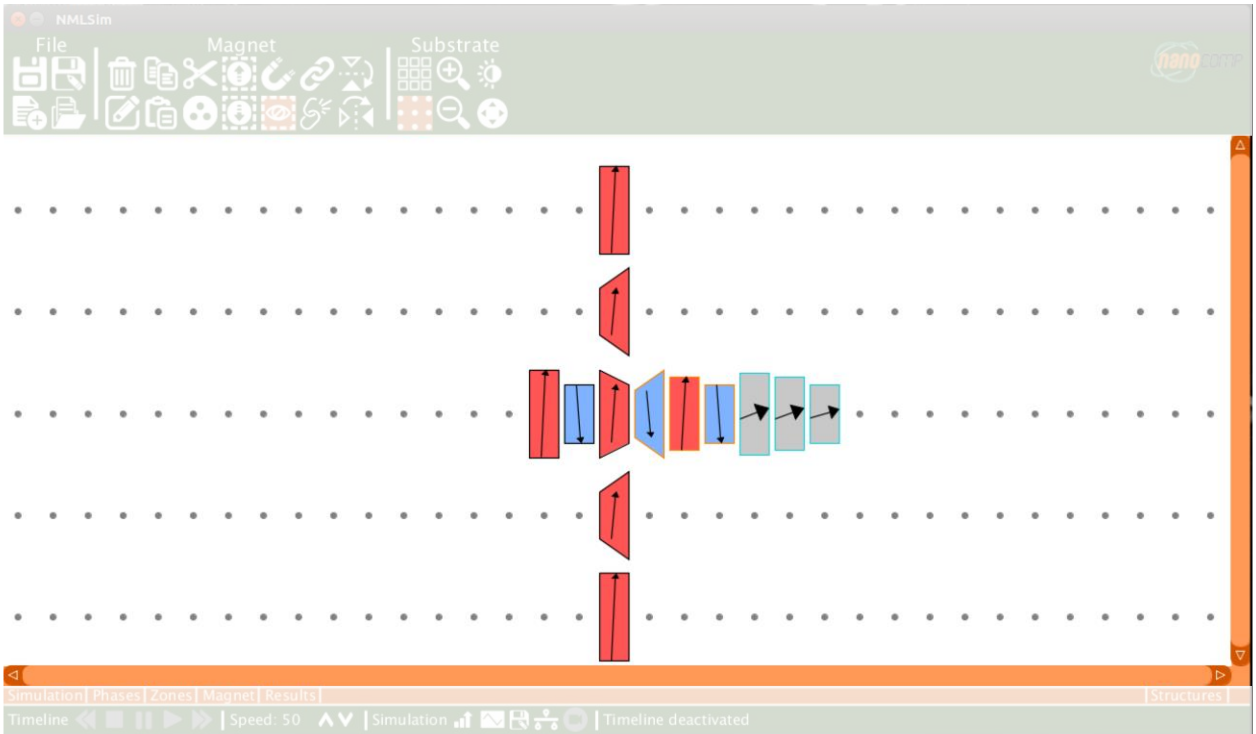


Figure 17. Substrate grid highlighted in magnetization view mode.
Note that previous figures have the clock zone mode view.

Simulation Bar

The simulation bar is composed of four sections (Figure 18). The first one is the timeline reproduction, with play, pause, stop, forward and backward buttons. This section is only enabled if the user clicks the button to enable it (in the third section).

The second section is the simulation speed controller. The user can raise or lower the animation speed by clicking the arrow buttons. The speed number is shown in this section.

The third section is the central simulation controller. Here we have five buttons which are, in order from left to right:

- **Simulate:** performs engine simulation. For that purpose, it opens a gnu-terminal (Figure 19). When the terminal closes, the simulation is done.
- **Plot Chart:** plots the chart of the magnetization over time for the selected magnets (Figure 20). Note that this button plots with default values and not the ones set in “results panel.”
- **Export XML:** exports the xml file for the user to edit or simulate directly with **make run** commands.
- **Enable Animation:** check if the simulation was performed and enables the timeline animation. This button loads the CSV file in the system.
- **Record Gif:** press once to start recording a GIF of the screen. Press again to compile the saved images and generate the GIF. We recommend using only for simulations with a low frame rate (try using a repot step of 0.1) and low simulation time (around 30 ns). Otherwise, *convert* might freeze the system by consuming a lot of RAM and CPU usage.

The last part is just the timeline animation status.

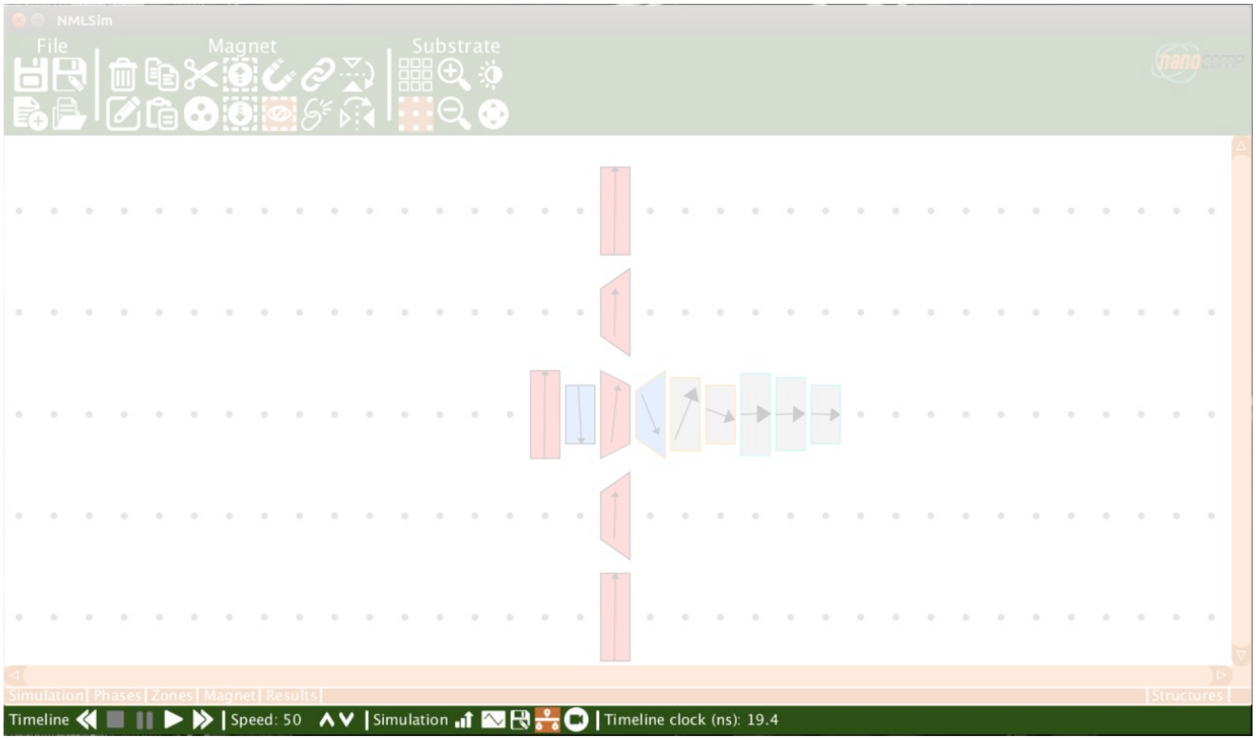


Figure 18. Simulation bar highlighted.

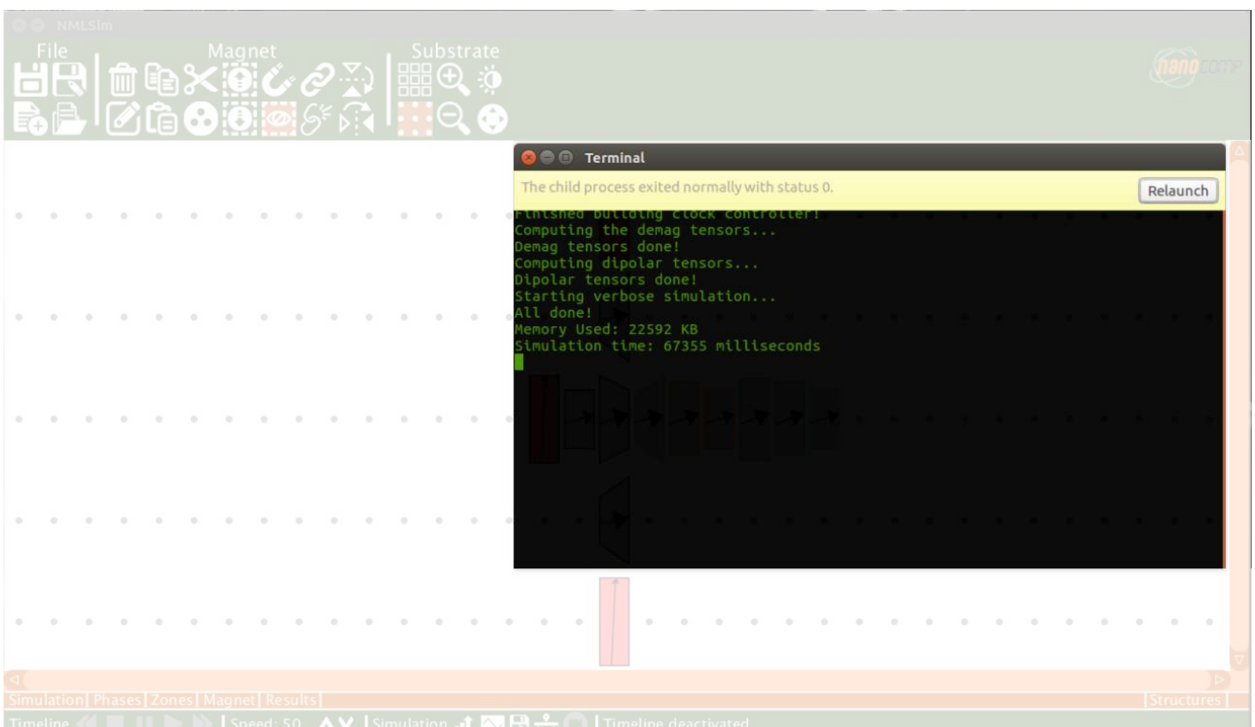


Figure 19. Simulation being performed in a gnu-terminal, highlighted in figure.

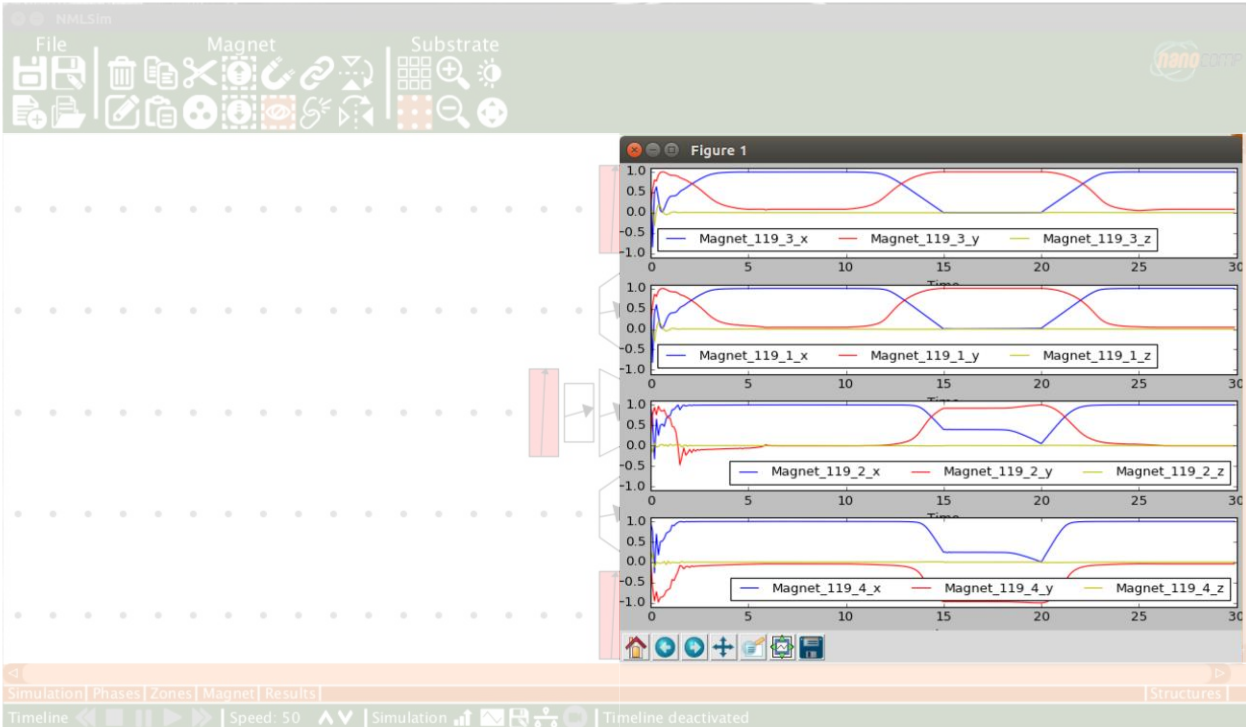


Figure 20. Chart plotted by python matplotlib, highlighted in red.

Shortcuts

The new simulator has the following shortcuts:

- **Ctrl + o:** open project.
- **Ctrl + n:** new project.
- **Ctrl + s:** save project.
- **Ctrl + Shift + s:** save project as a new project.
- **Alt + m:** rotate magnetization.
- **Alt + v:** toggle zone view mode.
- **Del:** delete selected magnets.
- **Ctrl + e:** edit selected magnet.
- **Ctrl + c:** copy selected magnet.
- **Ctrl + v:** toggles paste mode.
- **Ctrl + x:** cut selected magnets.
- **Ctrl + '+':** zone up selected magnets.
- **Ctrl + '-':** zone down selected magnets.
- **Alt + l:** link magnets.
- **Alt + u:** unlink magnets.
- **Ctrl + f:** horizontal flip.
- **Ctrl + Shift + f:** vertical flip.
- **Ctrl + r:** toggles the grid.
- **Ctrl + b:** toggles the bullet.

- **Ctrl + l:** toggles the light mode.
- **Ctrl + m:** toggles the substrate move mode.
- **Ctrl + t:** save selected structure as a template.
- **Alt + s:** simulate the circuit.
- **Alt + c:** plot the chart for selected magnets.
- **Alt + e:** export the XML file.
- **Alt + t:** enables timeline animation.
- **Alt + r:** record GIF.
- **Alt + p:** play/pause.
- **Alt + Shift + p:** stop.
- **Left Arrow:** backward.
- **Right Arrow:** forward.
- **Ctrl + "Mouse Scroll":** zoom in and out of the substrate.
- **Shift:** select multiple magnets.
- **Ctrl + "Mouse drag n' drop":** create selection box.

My First Circuit

This section provides a detailed tutorial on how to create a simple majority gate on NMLSim 2.0. Each subsection explains how to use a specific set of tools in the interface. Note that you can create the same circuit with the XML file. The interface is just an easier way to manipulate and visualize the circuit.

Simulation Properties

To start your first NMLSim circuit, hit the new button or just run the interface. The first panel to be filled is the simulation panel. Start by choosing LLG engine, to have a simulation closer to the reality. Select the verbose mode, which will enable the timeline animation in latter steps. Choose the RK4 method to simulate without temperature, which gives a more stable result.

The next step is to fill some of the textboxes. Since we want to simulate permalloy, Gilbert Damping constant should be 0.05, and the saturation magnetization should be 800.000. If you want to use different values for those fields, be mindful that they have a very powerful and unpredictable effect in your simulation. The time step and report step have a direct balance of performance and precision in your simulation and results. The bigger those values are, the faster the tool will simulate, and the less precise the results will be. For this project, let's use a time step of 0.0001 to have a very precise result and a report step of 0.1 to have less frames for the timeline animation.

This circuit will use Zeeman field for the clock system, which means that the three fields of the Spin Hall effect (Spin Angle, Spin Dif. Len., and H. M. Thickness) will be disregarded.

The remaining fields can be left unchanged for now. The simulation time depends on the circuit configuration. The neighborhood radius depends on the dimensions of the basic cell. At last, the substrate configurations depend on both the basic cell and the circuit.

Clock Phases

The next step is to configure the clock phases. Since this project uses the Zeeman field of 300 mT, the textboxes related to the initial and end current will remain 0. Note that these values must be filled even when not used. In this project, you are going to use a 4 phases clock scheme, to avoid back propagation of the signal. All phases are going to have the same duration, which will be 5 nanoseconds.

The values and names of each phase should be as follows (name: initial -> end):

- Switch: (300, 0, 0) -> (0, 0, 0)
- Hold: (0, 0, 0) -> (0, 0, 0)
- Reset: (0, 0, 0) -> (300, 0, 0)
- Relax: (300, 0, 0) -> (300, 0, 0)

All phases are showcased in the example bellow (Figure 21).

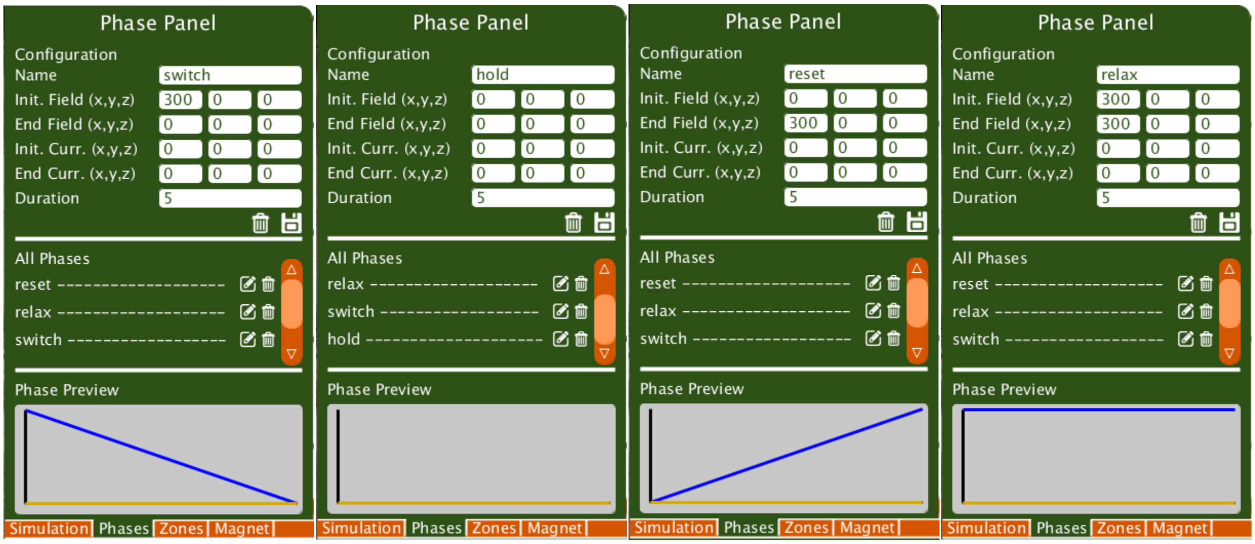


Figure 21. All clock phases in their respective panels.

Clock Zones

After defining the clock phases, the next step is to define the clock zones. Since there are four phases, there should be four zones as well. To build the first zone, type a name for it and choose a color. Then, add the phases in the following order: reset, relax, switch, hold. After adding all phases, click the button to add the zone to the “All Zones” list.

The second zone should have a different label and color. Add the phases in the following order: hold, reset, relax, switch. Note that the only difference is that the last phase of the previous zone is the first phase of this zone, but the order is preserved. Do the same for the two remaining zones.

All zones are showcased in the example bellow (Figure 22).

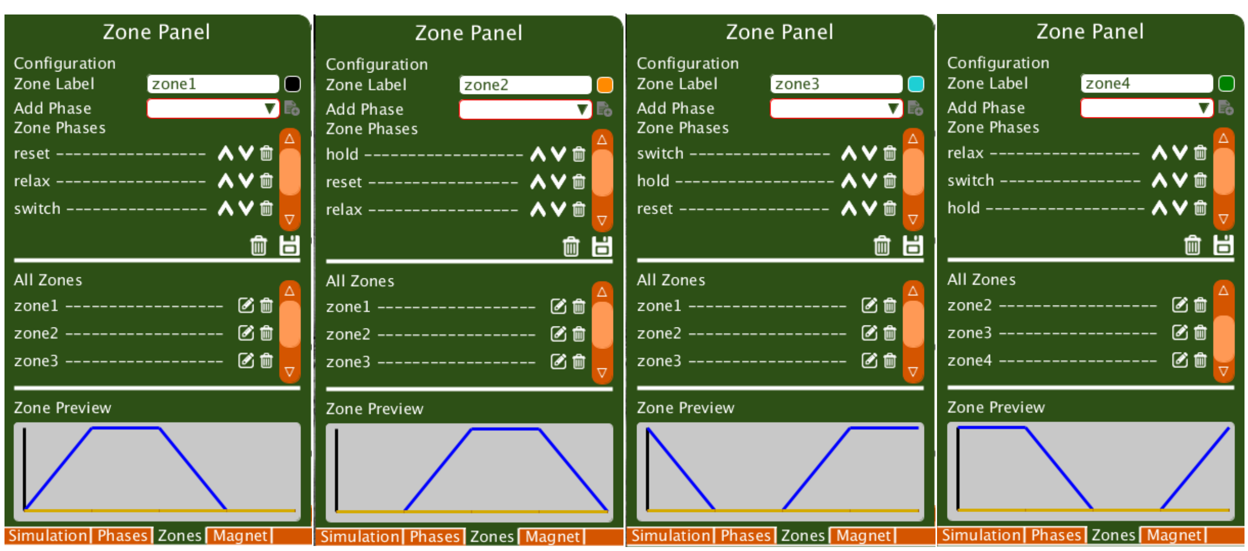


Figure 22. All clock zones in their respective panels.

Basic Cell

Before starting to build your circuit, consider saving a magnet as your basic cell. This will ease the process of circuit building. For this project, let's configure the basic cell in the magnet panel. Start by adding a label to the structure, such as "Basic_Cell" or something similar. Chose the regular type and a random clock zone. The initial magnetization is not important for a template structure; therefore, it's recommended using (0.141, 0.99, 0 – A magnet pointing up with a slight right tilt). Leave the "No Filed Effect" unchecked. For the geometry, let's use 50 nm width, 150 nm height, 15 nm thickness, and no cuts. Since the position is not important for the template, use a random positive number (greater than 0).

After all configurations are defined, the magnet panel should look something like the example bellow (Figure23). Click the "save template" button and check if the structure was added to the structure panel.

It is useful to define also a template for the input. Change the magnet type to input and check the "No Field Effect" box.

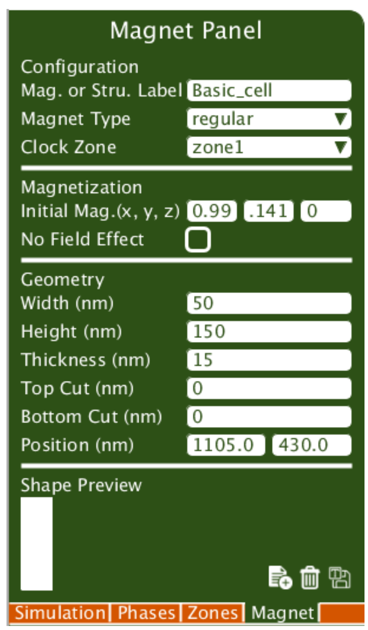


Figure 23. Magnet panel for the basic cell.

Building the Foundation

Now that your basic cell and input cell are configured as templates, it is time to revisit the Simulation panel and fill out more configurations. Since the magnet shape is 150x50, it is recommend using a neighborhood radius of 300 nm. The radius is a difficult value to obtain, where the simpler way is to perform experiments and compute the coupling tensor between magnets. In the NMLSim tool, the way to define is by moving a magnet away from a fixed one until you notice that they do not interact with each other anymore. For most simulations 500 nm is a safe enough distance for the radius. Note that the smaller the radius is, the faster the simulation will be.

For this project, the distance between cells will be 20 in the horizontal and 24 in the vertical. Therefore, set the bullet spacing to 70 by 174 (remember to consider the width and height of the basic cell on the bullet spacing). The substrate must have at least 420 nm in width, and 870 nm in height. Let's set the substrate to (2000, 900) to have more space. Note that the dimensions of the substrate do not change the simulation speed, it is only useful for the interface and can be edit at any time without causing any loss.

Now that you have the basic cells saved as templates and the substrate grid is configured properly, it's time to build the foundation to the majority gate. Start by adding one cell in the center of the substrate, this is going to be the output of the gate. Add one cell on the top of the first, another in the bottom and one last in the left. For each new cell, add an input cell following the same order. Remember to put all cells in the clock zone 1. After these steps, your circuit will look like the example bellow (Figure 24).

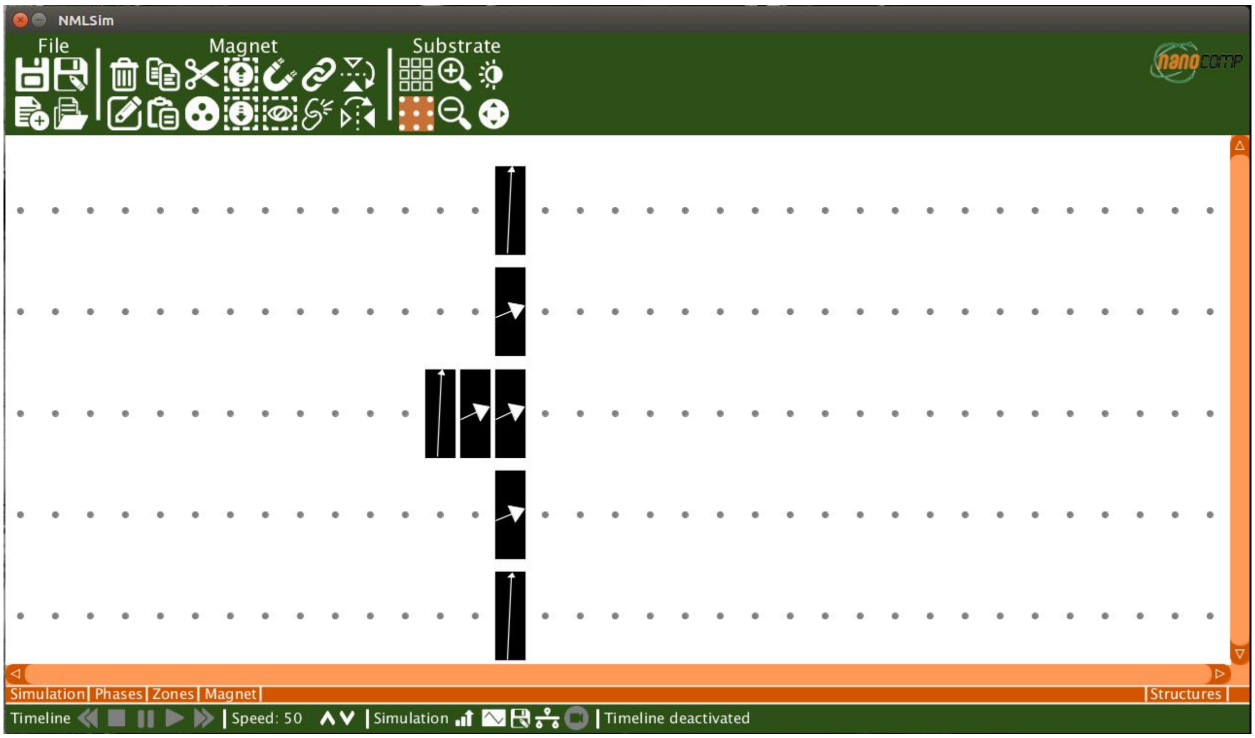


Figure 24. Foundation for the majority gate.

After having the gate done, let's add a wire composed of two clock zone to extract the information from the gate and propagate it. Remember that wires longer than 3~5 magnets tend to have propagation issues. Therefore, lets add 3 magnets to the right of the output magnet in the clock zone 2. Finally, add 3 magnets completing the anti-ferromagnetic wire in the clock zone 3. You should have something like the example bellow (Figure 25).

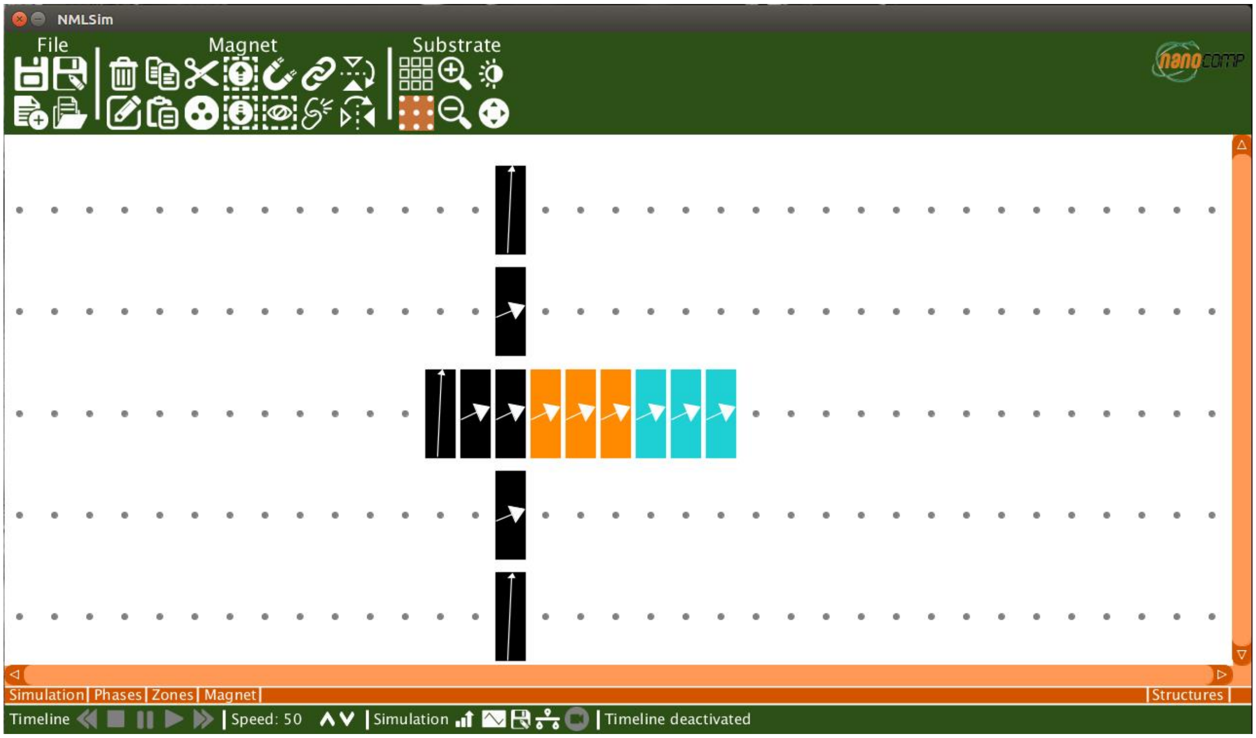


Figure 25. Foundation with the wire.

Information Extraction

If you simulate the previous example (Figure 25), you will notice that the circuit does not behave as desired. That happens because the interaction of the particles is not optimized. Even if the majority gate works as it is, the diagonal influence from the cells in the gate and the first magnet of the orange zone is stronger than the interaction of this magnet and the output.

There are some ways to fix this problem in NML. In this example, you will play with the geometry of the cells to favor the desired coupling. Start by editing the output magnet of the majority, adding a top and bottom cut of 25 nm. Then, edit the magnet in the left side of the output to have 100 nm of height. The next step is to edit the magnets on top and bottom of the output (one at a time), adding a top and bottom cut of -35 nm (which means 35 nm on the left side). At last, edit the first magnet of the orange wire to have a top and bottom cut of -35 nm.

With these modifications, you make the anti-ferromagnetic coupling from the output and the wire stronger, while diminishing the undesired diagonal coupling. The final configuration should look like the example bellow (Figure 26).

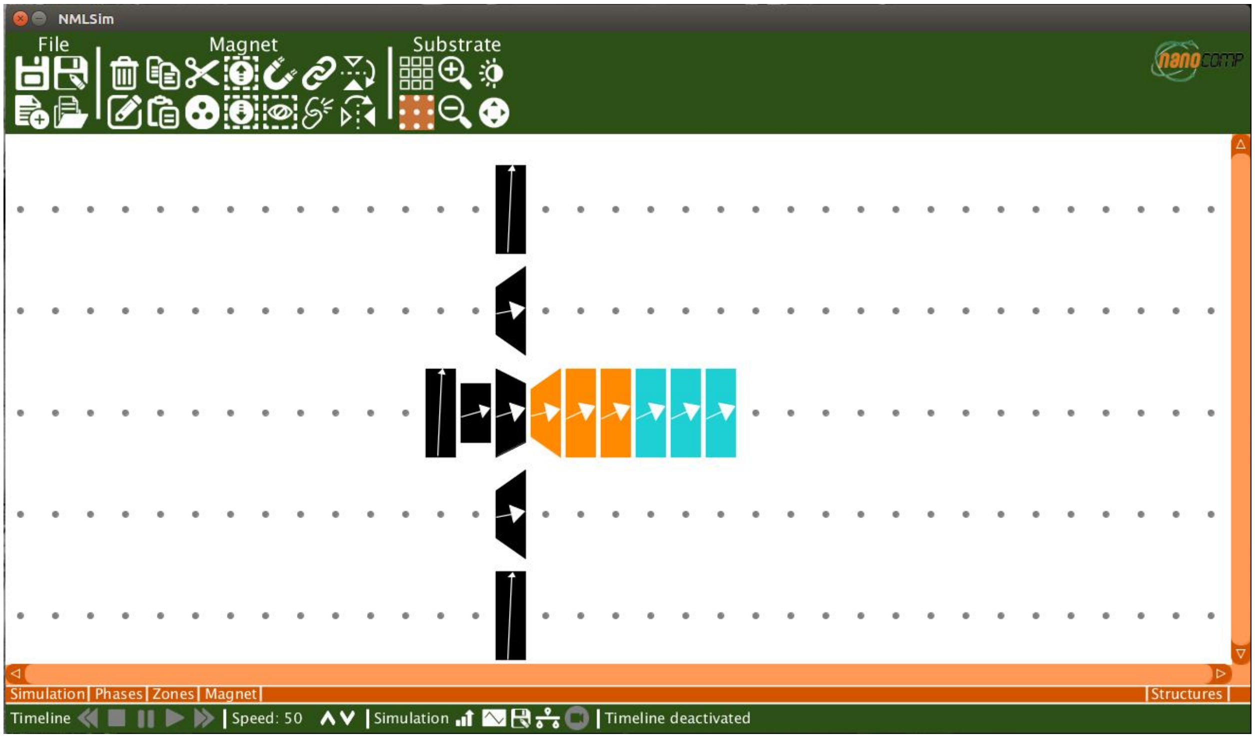


Figure 26. Majority with the extraction issue fixed.

Order of Propagation

When all magnets in a wire have the same size, they tend to magnetize simultaneously. This can be a huge issue when some cells define their magnetization before their predecessors. There is a simple way to fix this issue by changing the dimensions of the cells. Bigger magnets tend to magnetize faster than smaller ones, due to the interaction with the clock field. Let's fix the order of propagation in our wires by having the first magnet with 150 nm of height, the second with 125 nm of height and the third with 100 nm of height. Another thing that can be problematic is the influence of the 100 nm height orange magnet with the 150 nm height blue magnet. To fix this issue, edit the blue magnet to have 140 nm instead of 150 nm in height. The final version of the circuit should look like the example bellow (Figure 27).

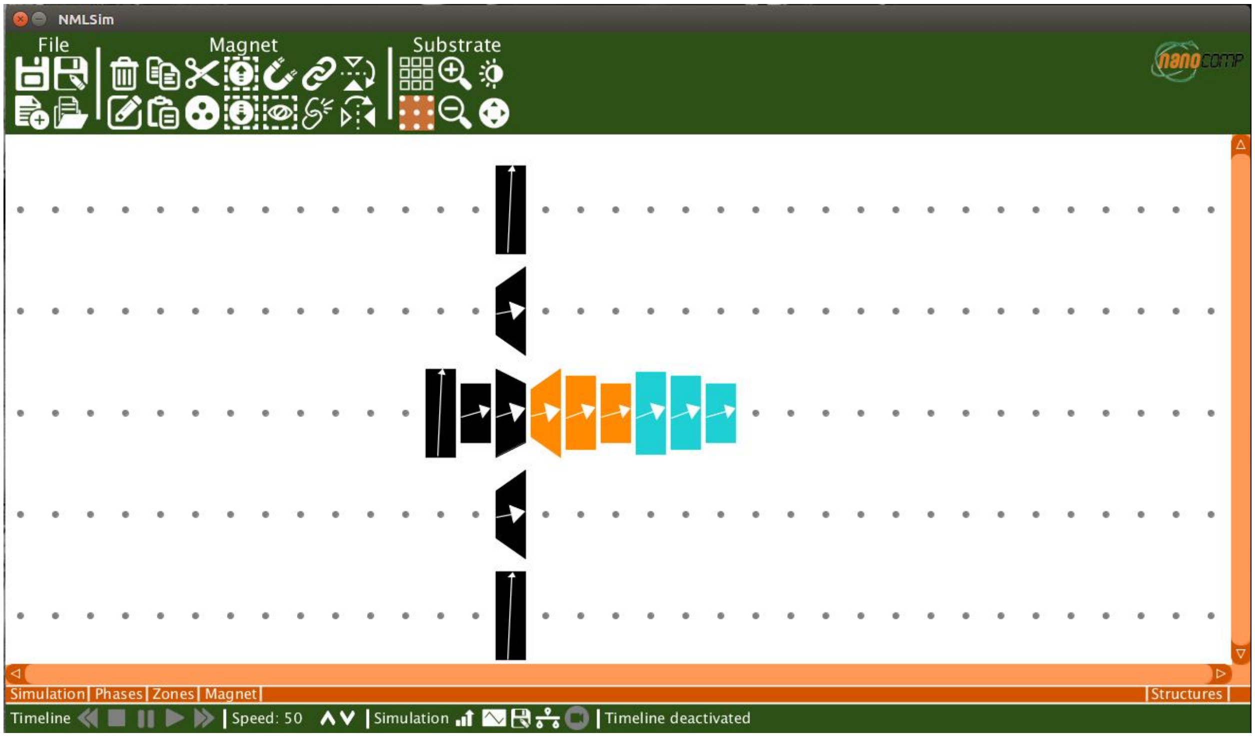


Figure 27. Circuit with propagation issue fixed.

Simulating and Visualizing Results

At this point, you already know how long your circuit will take to propagate the signal to the last magnet. In this case, the simulation time will be 30 ns, since the zone 1 takes 20 ns to reach hold, the zone 2 takes 5 additional nanoseconds, and the zone 3 takes another 5 additional nanoseconds. Return to the simulation panel and configure the simulation time field.

Now that everything is ready to go, click the simulation button in the simulation bar in the bottom. It's important to wait for the terminal to close or complete (this depends on your gnu-terminal configurations) to continue using the interface.

After the simulation is complete, you can visualize the animation by clicking the timeline button and then the play button. You can control the speed of the animation in the simulation bar. After seeing that everything is working fine, you can record a GIF image by:

- pressing the stop button
- pressing the record GIF button (it will turn orange)
- pressing the play button (the animation will be slower while recording)
- when the animation reaches the end, press the record GIF button again

When recording a GIF, the gnu-terminal will open. Remember to wait for it to finish before using the interface again. The speed of the timeline has no influence in the FIG, since it uses frame capture to record the GIF. Remember that convert does not work well with too many frames.

To visualize the magnetization chart over time, select the desired magnets and then press the chart button in the simulation bar.

The timeline animation works only with the verbose mode. The chart works with the verbose and the repetitive modes. To visualize the results for the direct and exhaustive modes, you will need to navigate to the project folder and open the CSV file.

Next Steps

When building bigger circuits, remember to consider the interactions of the cells close to each other. You might need to change the dimensions and cuts of the particles in the majority gate example to fix undesired interactions.

Final Considerations

This is a version of the tool in beta tests, expect bugs! There are some known issues reported in this section. If you find any other problems, please compile and report them to lucaslascasas5@gmail.com. Please try not to spam a lot of issues, compile all of them in a single email.

The algorithm to compute the demagnetization tensor can generate outliers in some very small cases. If you don't agree with the behavior of your simulated circuit, delete the ".log" file in the project folder and simulate it again. This file has a log of all demagnetization tensors used in your current project, this log speeds up the simulation time.

Sometimes the structure panel bugs the scroll bar, making it way longer than it needs to be. There is no harm caused by this. This bug is being fixed.

Note that the engine does not compile the XML file. If you make a typing error in the file, it will only generate an error and stop the simulation.

We are open to suggestions of improvement in the interface on the same email. Please use the indicator **[NMLSim]** in your email titles to help me organize the inbox.

We appreciate you being one of our beta testers.

Best Regards,
Lucas Lascasas