NMLSim 2.0: a robust CAD and simulation tool for in-plane nanomagnetic logic based on the LLG equation

Conference Paper · August 2019

DOI: 10.1145/3338852.3339856

CITATIONS
0

READS
69

4 authors:

Lucas Augusto Lascasas Freitas
Federal University of Minas Gerais
2 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Omar Vilela Neto
Federal University of Minas Gerais
64 PUBLICATIONS **342** CITATIONS
SEE PROFILE

João Guilherme Nizer Rahmeier
Carleton University
5 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Luiz G. C. Melo
Federal University of Minas Gerais
34 PUBLICATIONS **433** CITATIONS
SEE PROFILE

NMLSim 2.0: A robust CAD and simulation tool for in-plane Nanomagnetic Logic based on the LLG equation

Lucas A. Lascasas Freitas
Omar P. Vilela Neto
lucas.freitas@dcc.ufmg.br
Federal University of Minas Gerais
Department of Computer Science
Belo Horizonte, MG, Brazil

João G. Nizer Rahmeier
joaoguilhermenizerra@cmail.carleton.ca
Carleton University
Department of Electronics
Ottawa, ON, Canada

Luiz G. C. Melo
luiz.melo@atelier7hz.com
Atelier 7hz
Acoustics and Vibration engineering
Montreal, QC, Canada

ABSTRACT

Nanomagnetic Logic (NML) is a new technology based on the magnetization of nanometric magnets. Logic operations are performed via dipolar coupling through ferromagnetic and antiferromagnetic interactions. The low energy dissipation and the possibility of higher integration density in circuits are significant advantages over CMOS technology. Even so, there is a great need for simulation and CAD tools for the proper study of large NML circuits. This paper presents a high-efficiency tool that uses the Landau–Lifshitz–Gilbert equation to evolve the magnetization of the particles over time in a monodomain approach. The new version of NMLSim comes with flexibility in its code, allowing expansion of the tool with ease and consistency. The results of simulated structures show the reliability of the simulator when compared with the current state of the art Object-Oriented Micromagnetic Framework (OOMMF). It also presents an improvement of up to 716 times in execution time and up to 41 times in memory usage.

CCS CONCEPTS

• **Hardware → Spintronics and magnetic technologies**.

KEYWORDS

Nanomagnetic Logic, CAD tool, LLG, Beyond CMOS

ACM Reference Format:
Lucas A. Lascasas Freitas, Omar P. Vilela Neto, João G. Nizer Rahmeier, and Luiz G. C. Melo. 2019. NMLSim 2.0: A robust CAD and simulation tool for in-plane Nanomagnetic Logic based on the LLG equation. In *32nd Symposium on Integrated Circuits and Systems Design (SBCCI '19), August 26–30, 2019, Sao Paulo, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3338852.3339856

1 INTRODUCTION

Throughout the years, we have seen an exponential density increase in CMOS chips following Moore's law, which lead to modern integrated circuits [1]. However, CMOS technology is close to its physical limits, therefore threatening this exponential growth for future devices. Several new technologies were proposed in recent years to overcome this problem [2]. One of them is Nanomagnetic Logic (NML) which is a technology that operates at ultra-low energy dissipation level [3, 4].

NML circuits are composed of a grid of nanomagnets placed on a plane interacting through the magnetostatic dipolar coupling. The magnetic polarization is associated with the binary values '0' and '1', used to perform Boolean logic. The information in an NML circuit is transmitted through ferromagnetic and antiferromagnetic couplings. Nanomagnets position and geometry defines the coupling interaction, which affects the logic of the component. Some simple circuits have been experimentally demonstrated in the past years [5–7], showing that NML is capable of performing majority logic and information propagation. The functionality of NML circuits is highly dependent on the correct positioning of the particles to avoid undesired interactions among neighbor cells. The validation of large NML devices demands the use of simulation tools that predict physical behavior and that are capable of dealing with a high number of particles without compromise computational resources and execution time.

To proper capture the physical behavior of a circuit, the Landau–Lifshitz–Gilbert equation (LLG) must be solved for every spin in a magnetic device [8]. However, when dealing with a large circuit, this method is computationally expensive. There are solutions for this simulation issue using high-level hardware description languages [9, 10], however, these solution neglects the physical aspects of the circuit, as they are a behavioral model, making these approaches inaccurate. This paper presents a new version of the NMLSim CAD and simulation tool [11]. The new NMLSim comes with a monodomain approach to solve the LLG equation, finding an ideal balance between physical model accuracy and simulation efficiency.

The paper is organized as follows. Section 2 provides a background on NML technology. In section 3, related works and the major motivation of this paper are presented. Section 4 presents the the new NMLSim engine in details. Section 5 shows the simulations performed using the proposed tool, compared to OOMMF. Finally, in section 6 we review and conclude the paper.

2 NANOMAGNETIC LOGIC

The most simple and basic structure for NML circuits is a rectangular nanosized magnet, which has a single-domain behavior. A 3D vector represents the magnetization of this structure. Because of the shape of the particle and its weak anisotropy constant, single Permalloy magnets tend to have their magnetization along their longer axis to minimize the demagnetizing energy. This energy has two stable minima, one where the vector points downwards and

SBCCI '19, August 26–30, 2019, Sao Paulo, Brazil

ACM ISBN 978-1-4503-6844-5/19/08…$15.00
https://doi.org/10.1145/3338852.3339856

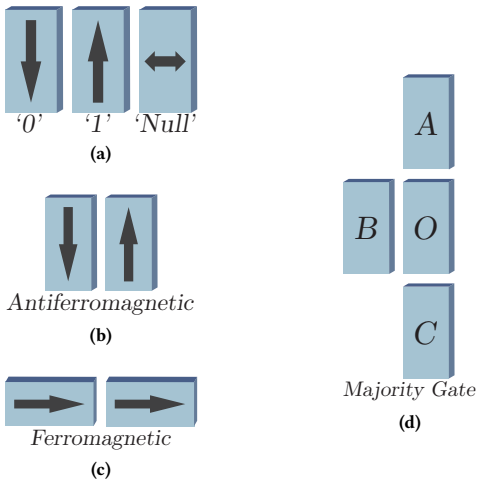

Figure 1: a) binary representation of the magnetization. b) antiferromagnetic coupling (AF). c) ferromagnetic coupling (FM). d) majority logic gate.

one upwards, allowing the definition of the binary values '0' and '1', respectively (Figure 1a). When applying an external field perpendicular to the particle, its magnetization is put into a meta-stable state where the vector matches the external source, defining a 'null' logic value (Figure 1a).

Nanomagnets can couple in a ferromagnetic (FM) and an antiferromagnetic (AF) way. We obtain an AF coupling by placing particles side by side, parallel to their longer axis, where the magnetization vectors of particles assume anti-parallel directions (Fig. 1b). Also, for rectangular particles aligned by the short axis, the FM coupling is dominant, where both magnetization vectors tend to point in the same direction (Fig.1c).

Through the two magnetic coupling, it is possible to propagate signal and perform logic operations in a magnetic system. However, to do so, the correct placing of the particles is necessary for performing the desired Boolean function. Any complex structure in NML also requires a well-elaborated clocking system to properly synchronize the inputs and outputs of each logic gate in the system.

In NML circuits, wires are made using either one of the two couplings described above. Due to weak dipolar interactions and thermal noise, nanomagnetic wires correctly propagate information provided the magnetic chain has up to around five nanomagnets [12]. In a FM wire all magnets present the same magnetization orientation throughout the magnetic chain. Conversely, AF wires form dipolar pairs that intercalate the magnetization orientation. An odd number of magnets present the same binary result at the last particle, and an even number of specimens works as an inverter.

The majority logic gate is one of the essential structures in NML circuit building. The Fig. 1d shows the majority gate implementation in NML, where the three magnets A, B and C work as inputs for the gate and the magnet O works as the output. While A and C have a ferromagnetic coupling with O, the magnet B has AF coupling with the output. Therefore the Boolean logic operation performed is $MAJ(A, \overline{B}, C)$. The logic gates *AND* and *OR* can be easily implemented using the majority gate, by using the magnet B as

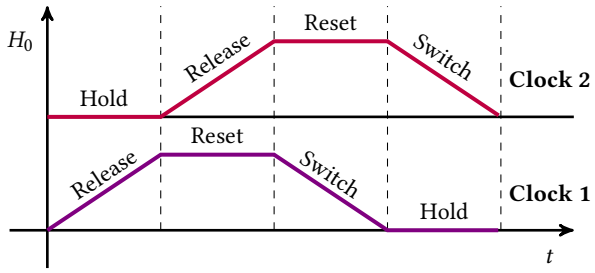

Figure 2: Clock phases *Hold*, *Release*, *Reset* and *Switch*. There is phase shifting of 90° between Clock 1 and Clock 2 signals.

a particle with fixed magnetization in the system. If B has its logic value fixed in 1, the majority gate then becomes $AND(A, C)$. On the other hand, when B has its logic value fixed in 0, the operation performed becomes $OR(A, C)$.

As NML circuits become more complex, the use of a clocking system becomes essential for timing the logic performed. For large circuits, many problems can occur without the correct use of a clocking system, such as propagation control loss, thermal influences on long wires and logic unities synchronization problems. For better comprehension and implementation of a clocking system, the clock signal can be considered as an external field source that controls the magnetization of the structures. This external field is usually perpendicular to the nanomagnets easy axis, this way when the field is applied the nanomagnets are forced into a 'null' meta-stable state. Once the external source is removed, the magnets are allowed to precess and interact with their neighbors.

The organization of the clock circuit is performed through the segmentation of the particles in groups called clock zones. The external field behavior during a given period is called a clock phase. Clock phases define the behavior of magnets in the system, and clock zones split the system into regions that perceive different clock phases in a given moment. Each clock zone has a list of phases, which is cyclical and ordered.

There are different approaches when modeling the clock system of a circuit; in this work, we use a four-phase clocking system as an example. Figure 2 shows the system working on two clock zones. In this system, the *Release* phase starts with no field applied and gradually increase it, forcing the magnets into a 'null' state. The *Reset* phase starts with the external field at maximum and keep the field applied, giving time for the magnets to stabilize at the metastable state. During the *Switch* phase, the external field previously applied is gradually removed, allowing magnets to return to a stable logic state of either 0 or 1. The *Hold* phase represents a period where no external field is applied, allowing the magnets to keep their current state. Since each clock signal is phased 90° from one another, we can operate with up to four clock zones.

3 RELATED WORK

Several micromagnetic simulation tools incorporate the formalism of the LLG equation, like *mumax*^{3}[13], LLG System Simulator [14] and the Object-Oriented Micromagnetic Framework (OOMMF) [15]. However, these tools are not suitable for simulating high-density NML because the resultant magnetization of each particle in the

model is composed by several interacting spins. In this particular study, we are going to use OOMMF (1.2 beta 0) as a comparing tool to validate our simulation results and also define some metrics around execution time and memory usage.

In another line of NML simulation, some tools seek the construction of an HDL model to simulate the circuit based only in the expected behavior from the dipolar coupling. The ToPoliNano tool [16] allows a high-speed simulation through a VHDL model, which correctly presents the logic behavior of the circuit. However, HDL models are very high-level models, and they neglect the design and the physical behavior of the circuit. The MagCAD tool [17] was created to allow design in a low-level and then export the circuit to ToPoliNano. When using MagCAD and ToPoliNano, we are limited to use a predefined magnet shape.Also, one must care with the direction of propagation since the side faces of the magnets have an "input" and "output" property used to scan and update the states in the circuit. Such scheme limits the creation of personalized gate structures.

NMLSim 1.0 also uses a predefined nanomagnet shape [11, 18]. The geometry and spacing of the particles were decided through studies of FM and AF coupling to guarantees that both have the same weight when interacting with neighbor particles. The coupling energy was computed for the set of configurations in a grid given the neighborhood radius. Next, the circuit is converted into a graph where the edge is weighted with the C_{yy} element of the coupling tensor, N_c. Therefore, it was able to account the interaction in the Y-axis, and truncation in the magnetization is done to carry out the condition $|m| = 1$. Although NMLSim 1.0 is a potent tool for behavioral simulation with backup in physical properties, the simulation is not very precise since it relies on an empirical model. Another issue is the fixed shape and spacing of the particles, restricting geometry alterations. At last, we are not allowed to change the clocking system for simulations of circuits that require unique clocking schemes.

4 METHODOLOGY

To create a high-performance NML simulator, all the critical elements in an NML device are modeled into data structures for proper manipulation in a C++ program. This section explains in details the LLG equation basics to the new engine, the data structure of the magnets, the clock modeling and the particles interaction in the circuit.

4.1 LLG Engine

As stated before, some physical aspects of the magnetization precession in time was not covered by the first version of NMLSim. In this 2.0 version we incorporate an engine that is based in the equation that physically governs the precession of the magnetization, called the Landau-Lifshitz-Gilbert (LLG) equation,

$$\frac{d\mathbf{M}}{dt} = -\gamma \mathbf{M} \times \mathbf{H}_{eff} + \frac{\alpha}{M_s}\mathbf{M} \times \frac{d\mathbf{M}}{dt}, \tag{1}$$

where $\gamma = 1,76 \times 10^{11} \times \mu_0\ m/(sA)$ is the gyromagnetic factor of the electron, α is the Gilbert damping constant, $\mathbf{M}$ is the magnetization and $\mathbf{H}_{eff}$ is the effective field. The first term in the right-hand side of (1) represents the magnetic torque from the interaction of $\mathbf{M}$ with $\mathbf{H}_{eff}$, and the second one is perpendicular to $\mathbf{M}$ and models Gilbert's damping effect. The effective field, $\mathbf{H}_{eff}$, is the sum of all magnetic fields, such that,

$$\mathbf{H}_{eff} = \mathbf{H}_k + \mathbf{H}_{ex} + \mathbf{H}_d + \mathbf{H}_c + \mathbf{H}_0 \tag{2}$$

where $\mathbf{H}_k$ is the magnetic anisotropy field, $\mathbf{H}_{ex}$ is the exchange field, $\mathbf{H}_d$ is the demagnetizing field, $\mathbf{H}_c$ is the coupling field and $\mathbf{H}_0$ is the Zeeman or external field.

In its current version, NMLSim is able to deal with in-plane nanomagnetic logic (iNML), this means that particle's magnetization has to lie in the sample's plane to code binary information. Permalloy ($Ni_{80}Fe_{20}$) is a material that presents this property since it has a low uniaxial anisotropy constant (K_u) [19]. From that, we can drop modeling the magnetic anisotropy field, $\mathbf{H}_k$. Also, magnet's size is in the order of 100 nm, which makes the magnetic moments to behave in unison throughout the sample, condition knows a macrospin. This last assumption allows us to ignore the effects of the exchange energy in the form of the exchange field, $\mathbf{H}_{ex}$. Therefore, all specimens referred in this work are made of Permalloy and considered to operate in a monodomain regime.

For a single particle, its geometry dictates the easy and hard axis for the orientation of $\mathbf{M}$, and allows the definition of what is a '0', '1' or 'null' state. The field responsible for characterizing this condition is the demagnetizing field, $\mathbf{H}_d$,

$$\mathbf{H}_d = -N_d \mathbf{M} \tag{3}$$

where N_d is the demagnetizing tensor,

$$N_d = \begin{bmatrix} N_x & N_{xy} & N_{xz} \\ N_{yx} & N_y & N_{yz} \\ N_{zx} & N_{zy} & N_z \end{bmatrix}. \tag{4}$$

When more than one particle is near each other, they can process and propagate information through the dipolar coupling. This interaction is modeled through the coupling field, $\mathbf{H}_c$,

$$\mathbf{H}_{c_{i,j}} = -N_{c_{j,i}} \mathbf{M}_j, \tag{5}$$

where $i \neq j$ corresponds to the particles indexes and $N_{c_{j,i}}$ the coupling tensor,

$$N_{c_{i,j}} = \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix}. \tag{6}$$

More information about the computation of the coupling and demagnetizing tensors can be found in [3] and [20].

In order to control the information propagation in a circuit, a clock system must be employed. In this version of the tool, we implement that via external field, through the Zeeman field, $\mathbf{H}_0$. This field can have components on all three directions and is modeled following the waveform and phases presented earlier in Sec. 2.

To update the magnetization at each time step we use an explicit and normalized version of Eq. (1),

$$d\mathbf{m} = -\frac{1}{(1+\alpha^2)}[\mathbf{m} \times \mathbf{h}_{eff} + \alpha \mathbf{m} \times (\mathbf{m} \times \mathbf{h}_{eff})]dt \tag{7}$$

where $\mathbf{m} = \mathbf{M}/M_s$, $\mathbf{h}_{eff} = \mathbf{H}_{eff}/M_s$, M_s is the saturation magnetization. We also use a new time step where $dt := dt' = dt\gamma\mu_0 M_s$. Since $\mathbf{m} = [m_x\ m_y\ m_z]$, equation (7) corresponds to a system of

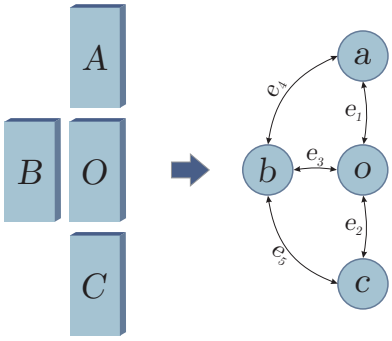

Figure 3: A magnetic circuit and its graph representation. The e_n values are the weights to model the interaction between a pair of particles.

coupled equations, and to obtain its solution we use the explicit Runge-Kutta method of fourth order [21]. As stated in [22], $|\mathbf{m}| = 1$, so the magnetization is allowed to assume values only at the shell of the unity sphere.

4.2 Data Structures and Modeling

The NML circuit's grid is modeled as a weighted bidirectional graph, where each node is a magnet and each edge is the coupling tensor between two nodes. As a consequence, the NML circuit can be represented by the relation $G = (V, E)$, where V is the group of nodes and E is the group of edges. This graph is represented through an adjacency list. Consider, for example, the transcription in Fig. 3. The magnets A, B, C and O are converted into nodes forming the group $V = \{a,\ b,\ c,\ o\}$. The coupling tensor is computed for neighbor nodes, creating the group $E = \{e_1,\ e_2,\ e_3,\ e_4,\ e_5\}$.

Each node of the graph encapsulates several data from a single magnet in the circuit. The node data structure is divided into two classes that communicate with one another, those are called the main class and the auxiliary class. The auxiliary class carries the information regarding the shape of the magnet, and it's responsible for computing the demagnetization and coupling tensors. The main class handles the adjacency list of a magnet as well as the magnetization status of the particle, and it's responsible for updating the magnetization over time by solving equation 7.

The nodes of a circuit connect with others using a neighborhood rule, where a magnet will only connect with others in a certain radius. This user configurable radius limitation enhances the performance of the simulator by ignoring coupling between magnets that are far apart in the circuit. We use a default value of 500 nm. Considering the previous example in Fig. 3, the neighborhood radius captures only the adjacent magnets, which makes the simulation faster by ignoring, for example, the interaction between nodes a and c. Since the graph is bidirectional, each pair of neighbor magnets have a copy of the coupling tensor in their adjacency list.

The process of converting a grid of magnets into the graph $G = (V, E)$ consists of two parts. The first part is the node's building process, where each node information is loaded into the main and the auxiliary classes, and the adjacency list is started with no elements. The final step is the edge creation, where each node computes the distance to the other nodes and add the ones in the neighborhood radius in his adjacency list. Since the graph is bidirectional, as soon as node a adds a node b as its neighbor, the node b adds the node a into its neighbor list as well. Once all magnets added their neighbors, the graph is complete and ready for the simulation process.

4.3 Circuit Simulation

The clocking system in the new version of NMLSim is highly configurable. The user can define an initial and final value for a phase signal, similar to the `Oxs_UZeeman Hrange` available on OOMMF. Clock phases in the NMLSim have linear variations only, since it's the most commonly used in literature and would make the modeling pretty straightforward. Also, they have different duration and different field variation. The user has total control in configuring the number of clock zones and the phase sequence each zone goes through.

We recommend using a time step of at least 500 fs to minimize convergence problems. In each time step, the clock phases update their current signal based on the variation defined. All magnets in a given clock phase perceive the same external control field. After the phases update their clock signal, every magnet of the circuit computes its future magnetization. After all particles have their future magnetization ready, they update their current value and the system time step advances.

5 RESULTS AND ANALISYS

All simulation parameters for OOMMF and NMLSim are presented in Table 1. They were kept constant for all comparison. OOMMF cell size is kept $2\times2\times10$ to minimize the stair effect on representing cut particles. A drawback on that is that to assure convergence the maximum time step must be decreased to 50 fs. Since OOMMF uses error minimization routines while solving the LLG equation, one might say that we could be more flexible and let him decide the actual step size on each iteration before it completes a stage computation. However, in NML circuits we need the ability to precisely control the application of the clock field as a function of time, to keep devices synchronized.

The first analysis compares the precession of magnetization in time for different geometries, configurations, with and without the presence of an external field as a clock signal. Figure 4 shows the results where colors indicate the components $\mathbf{m} = [m_x\ m_y\ m_z]$, solid lines are NMLSim curves, dashed lines with circles are OOMMF's results, and the insets represent particles geometry and orientation in the XY plane. All particles are $50 \times 100 \times 10\ nm^3$, and we point out whenever a particle has a cut. In 4a, the initial magnetization for a rectangular magnet is $\mathbf{m}_i = [0.1\ 0.98999\ 0.1]$. It starts slightly off the plane, tilted to the right and mostly along the Y-axis, so when it starts to precess it moves towards its longer axis, pointing to the right. Both results are similar, and the phasing we observe after 0.5 ns is a consequence of rouding/cumulative errors during calculation of tensor coefficients. In Fig. 4b we have a single particle in the vertical with a cut of 25 nm on the top right corner. Its initial magnetization is $\mathbf{m}_i = [0.98999\ -0.1\ 0.1]$, starting at a metastable state and slightly downwards. As expected, after 2 ns its final state shows its magnetization lying in the diagonal, m_x slightly positive,

caused by the cut inserted on the edge. Both simulations presented a similar precession and the same final state.

The second round of comparisons comprehends the dipolar interaction between particles. The dotted rectangles on the insets of Fig. 1c and 1d represents an input with fixed magnetization, and the blue magnets are the particles of interest (PoI) that have their $\mathbf{m}$ plotted. In Fig. 1c we show an FM coupling, particles 25 nm apart, the input is fixed at $\mathbf{m}_i = [-1\ 0\ 0]$ and the PoI starts with $\mathbf{m}_i = [0.01\ 0.9999\ 0]$, slightly to the right. Since this configures an FM wire, the PoI couples in the same direction as the input. In Fig. 1d we simulate an AF structure where the PoI has a cut of 25 nm on its bottom-right corner. The input has $\mathbf{m}_i = [0\ -1\ 0]$ and the PoI $\mathbf{m}_i = [0.98\ -0.199\ 0]$. It fully accomplishes the AF coupling in both simulators. Although OOMMF represents a micro structure, its cell size still modifies sharp corner definitions. Also, NMLSim has its variability for tensor calculation. Both conditions influence the overall precession and interaction of the particles. Hence, phasing observed in Fig. 1d is a combination of those factors. Furthermore, these variations may account for physical imperfections on the deposition of the nanomagnets in a substrate, but more research is needed to better describe that.

The third comparison involves the application of an external field of 125 mT, following the phase sequence of Clock 1 from Fig. 2, where each phase lasts 1 ns. Figure 4e shows a reset operation in a rectangular particle where H_0 is applied in the +x-direction. After 2.5 ns we see that the particle flips its orientation. Since there is no input to interact with the particle, this behavior becomes random, and the flipping depends on numerical errors. Figure 4f shows a reset operation on a slanted particle (25 nm cut in the bottom-left corner). In this configuration H_0 did not provide enough energy so that the magnetization lies in a metastable state

Our final results compare execution time and memory usage by both simulators when analyzing a grid of $N \times N$ rectangular particles throughout 2 ns. Table 2 has the mean of 30 runs (except when indicated) for each of the metrics. The tests were performed in a machine with an Intel® Core™ i5-2400 CPU @ 3.1 GHz and 7.7 GiB of RAM. For a OOMMF cell size of $2 \times 2 \times 30\ nm^3$ we have an improvement of 193.88 up to 716.23 fold in execution time, and 7.6 up to 41.27 fold in memory usage. A similar result were reported in [23] where a shape-dependent AND gate, during a 20 ns experiment, took 10-20h to be simulated in OOMMF with a grid of $2\times2\times30\ nm^3$. Since all particles on the grid are rectangular, we also simulated the circuits in OOMMF with a greater cell size ($5 \times 5 \times 5\ nm^3$) and a time step of 500 fs. In such case, there is a gain of 4.33 up to 26.29 fold in execution time and 5.53 up to 20.11 fold in memory usage. We notice that the time required by the NMLSim to simulate a 4 times greater grid is comparable to OOMMF's. Such improvement comes from the simplifications considered in the model, mainly because of the monodomain approach which reduces the number of cells to be computed at each time step and diminish the amount of memory required to store the overall state of the system. Also, OOMMF was able to run with four threads while NMLSim can still be optimized through parallel libraries, which might power up its performance.

Table 1: NMLSim and OOMMF Simulation Parameters

Variable	Symbol	NMLSim	OOMMF
Sat. magnetization (A/m)	M_s	8×10^5	8×10^5
Damping constant	α	0.05	0.05
Exchange constant (J/m)	A	-	1.3×10^{-10}
Anisotropy constant (J/m^3)	$K1$	-	0
Time step (fs)	dt	500	50
Reporting time (ps)	-	5	5
Cell size (nm^3)	-	-	$2 \times 2 \times 10$
Method	-	Runge-Kutta 4$^{\text{th}}$ Order	

Table 2: Execution Time and Memory Usage

Cell Size	N	NMLSim		OOMMF		Improvement	
		ET (s)	M (MB)	ET (s)	M (MB)	ET (×)	M (×)
$2 \times 2 \times 10\ nm^3$	2	2.13	4.38	422.58	43.83	198.24	10.00
	4	9.15	4.50	1,774.31	34.22	193.88	7.60
	8	30.29	4.69	8,138.81†	118.42†	268.68	25.25
	16	50.87	6.03	36,432.00*	248.68*	716.23	41.27
$5 \times 5 \times 5\ nm^3$	2	2.13	4.38	14.09	43.92	6.61	10.02
	4	9.15	4.50	39.60	24.89	4.33	5.53
	8	30.29	4.69	156.30	94.33	5.16	20.11
	16	50.87	6.03	694.82	105.25	13.66	17.47
	32	128.94	12.13	2,992.99	151.97	23.21	12.53
	64	488.21	38.06	12,835.25*	638.42*	26.29	16.77
	128	2,212.27	145.30	–	–	–	–
	256	12,332.00*	581.02*	–	–	–	–

Markers indicate conditions where we performed a reduced number of runs: *5, †15.

6 CONCLUSION AND FUTURE WORK

In this work, a CAD and simulation tool for NML circuit is presented, which uses the LLG equation to compute the precession of the magnetization over time. NML circuits are converted into a data structure for efficient manipulation, which is a bidirectional weighted graph. The tool is high-configurable when dealing with clocking systems and magnets geometry.

The monodomain approach chosen for simulation efficiency was validated through several comparisons with the well established micromagnetic simulator OOMMF. In terms of physical behaviour, both tools present similar results. However, they severely differ in terms of execution time and memory usage, where a four times bigger circuit than a given circuit simulated by OOMMF can be simulated by the proposed tool with the same computational resources.

As ongoing research, there is a new version of the tool in development featuring a new engine that accounts for thermal effects and enables the usage of spin Hall effect for implementing the clocking scheme. An user interface that allows an easier manipulation when building NML circuits is also in development.

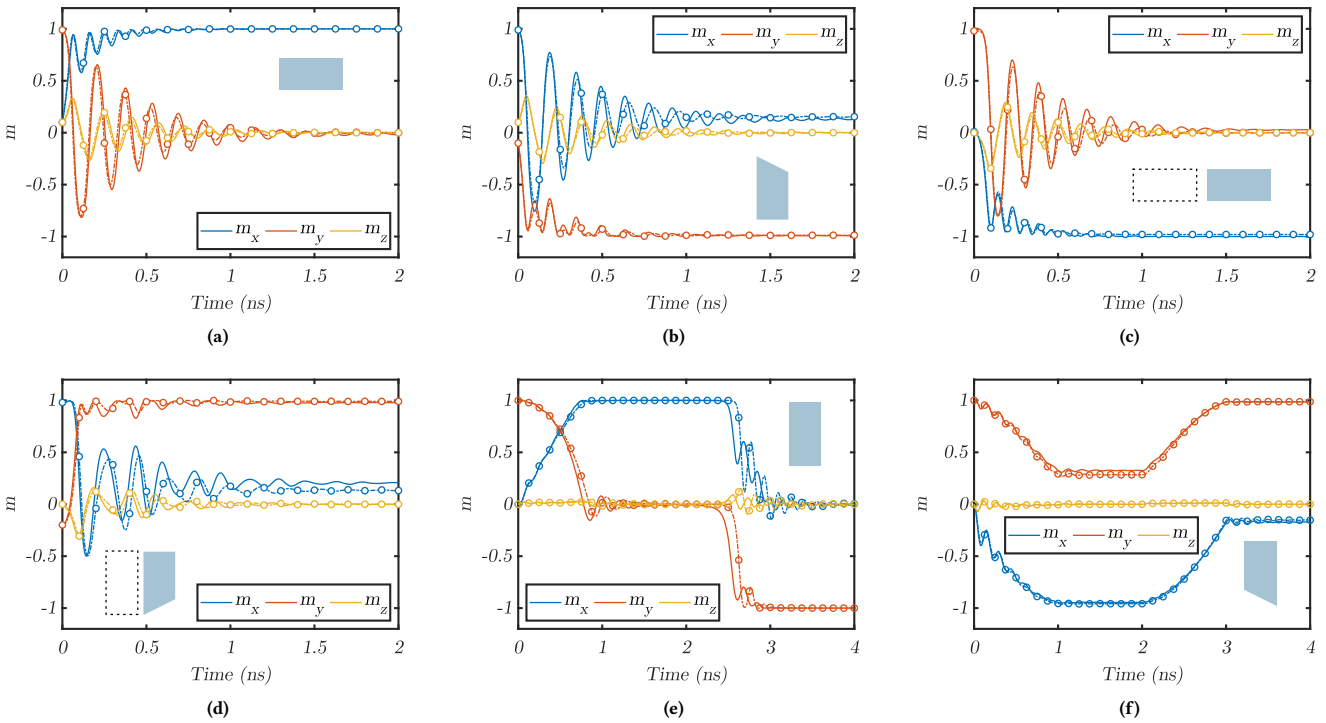

Figure 4: Comparison results between magnetization components (colors) from NMLSim (solid lines) and OOMMF (dashed-circled lines). Blueish insets represent particles geometries and allocation. Dashed particles represent fixed inputs used to polarize the particles of interest. See text for a detailed explanation of each configuration.

REFERENCES

[1] M Mitchell Waldrop. The chips are down for moore's law. *Nature*, 530(7589): 144–147, 2016.

[2] Ralph K Cavin, Paolo Lugli, and Victor V Zhirnov. Science and engineering beyond moore's law. *Proceedings of the IEEE*, 100(Special Centennial Issue): 1720–1749, 2012.

[3] Luiz G. C. Melo, Thiago R. B. S. Soares, and Omar P. Vilela Neto. Analysis of the Magnetostatic Energy of Chains of Single-Domain Nanomagnets for Logic Gates. *IEEE Transactions on Magnetics*, 53(9):1–10, sep 2017. ISSN 0018-9464. doi: 10.1109/TMAG.2017.2704913.

[4] MT Niemier, Gary H Bernstein, G Csaba, A Dingler, XS Hu, S Kurtz, S Liu, J Nahas, W Porod, M Siddiq, et al. Nanomagnet logic: progress toward system-level integration. *Journal of Physics: Condensed Matter*, 23(49):493202, 2011.

[5] Alexandra Imre, G Csaba, L Ji, A Orlov, GH Bernstein, and W Porod. Majority logic gate for magnetic quantum-dot cellular automata. *Science*, 311(5758):205–208, 2006.

[6] Edit Varga, Alexei Orlov, Michael T Niemier, X Sharon Hu, Gary H Bernstein, and Wolfgang Porod. Experimental demonstration of fanout for nanomagnetic logic. *IEEE Transactions on Nanotechnology*, 9(6):668–670, 2010.

[7] Edit Varga, G Csaba, GH Bernstein, and W Porod. Implementation of a nanomagnetic full adder circuit. In *2011 11th IEEE International Conference on Nanotechnology*, pages 1244–1247. IEEE, 2011.

[8] Alex Hubert and Rudolf Schäfer. *Magnetic domains: the analysis of magnetic microstructures*. Springer Science & Business Media, 2008.

[9] M. Graziano, A. Chiolerio, and M. Zamboni. A technology aware magnetic qca ncl-hdl architecture. In *2009 9th IEEE Conference on Nanotechnology (IEEE-NANO)*, pages 763–766, July 2009.

[10] Mariagrazia Graziano, Marco Vacca, Alessandro Chiolerio, and Maurizio Zamboni. An ncl-hdl snake-clock-based magnetic qca architecture. *IEEE Transactions on Nanotechnology*, 10(5):1141–1149, 2011.

[11] Thiago R. B. S. Soares, João G. Nizer Rahmeier, Vitor C. de Lima, Lucas Lascasas, Luiz G. Costa Melo, and Omar P. Vilela Neto. NMLSim: a Nanomagnetic Logic (NML) circuit designer and simulation tool. *Journal of Computational Electronics*, 17(3):1370–1381, sep 2018. ISSN 1569-8025. doi: 10.1007/s10825-018-1215-8.

[12] György Csaba and Wolfgang Porod. Behavior of nanomagnet logic in the presence of thermal noise. In *2010 14th International Workshop on Computational Electronics*, pages 1–4. IEEE, 2010.

[13] A Vansteenkiste, J Leliaert, M Dvornik, F Garcia-Sanchez, and B Van Waeyenberge. The design and verification of mumax3, aip advances 4, 107133 (2014). *CAS Article*.

[14] Michael R Scheinfein. Llg micromagnetics simulator. 18:25, 1997. URL http://llgmicro.home.mindspring.com.

[15] M. J. Donahue and D. G. Porter. OOMMF user's guide, version 1.2 beta 0, 2016. URL http://math.nist.gov/oommf.

[16] F. Riente, G. Turvani, M. Vacca, M. R. Roch, M. Zamboni, and M. Graziano. Topolinano: A cad tool for nano magnetic logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(7):1061–1074, July 2017. ISSN 0278-0070. doi: 10.1109/TCAD.2017.2650983.

[17] Fabrizio Riente, Umberto Garlando, Giovanna Turvani, Marco Vacca, Massimo Ruo Roch, and Mariagrazia Graziano. Magcad: tool for the design of 3-d magnetic circuits. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 3:65–73, 2017.

[18] Thiago RBS Soares, Isaias F Silva, Luiz GC Melo, and Omar P Vilela Neto. A new methodology for design and simulation of nml circuits. In *2016 IEEE 7th Latin American Symposium on Circuits & Systems (LASCAS)*, pages 259–262. IEEE, 2016.

[19] L. F. Yin, D. H. Wei, N. Lei, L. H. Zhou, C. S. Tian, G. S. Dong, X. F. Jin, L. P. Guo, Q. J. Jia, and R. Q. Wu. Magnetocrystalline anisotropy in permalloy revisited. *Physical Review Letters*, 97(6):1–4, 2006. ISSN 00319007. doi: 10.1103/PhysRevLett.97.067203.

[20] R. Moskowitz and E. Della Torre. Theoretical aspects of demagnetization tensors. *IEEE Transactions on Magnetics*, 2(4):739–744, dec 1966. ISSN 0018-9464. doi: 10.1109/TMAG.1966.1065973.

[21] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer New York, 2002. doi: 10.1007/978-0-387-21738-3.

[22] Ivan Cimrák. A Survey on the Numerics and Computations for the Landau-Lifshitz Equation of Micromagnetism. *Archives of Computational Methods in Engineering*, 15(3):277–309, sep 2008. ISSN 1134-3060. doi: 10.1007/s11831-008-9021-2.

[23] S Kurtz, E Varga, M J Siddiq, M Niemier, W Porod, X S Hu, and G H Bernstein. Non-majority magnetic logic gates: a review of experiments and future prospects for 'shape- based' logic Non-majority magnetic logic gates: a review of experiments and future prospects for 'shape-based' logic. *J. Phys.: Condens. Matter*, 23:53202–13, 2011. doi: 10.1088/0953-8984/23/5/053202.