Journal of
Applied Physics

Tomorrow's micromagnetic simulations

Cite as: J. Appl. Phys. **125**, 180901 (2019); https://doi.org/10.1063/1.5093730
Submitted: 24 February 2019 . Accepted: 25 April 2019 . Published Online: 10 May 2019

J. Leliaert, and J. Mulkers

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Perspective: Magnetic skyrmions—Overview of recent progress in an active research field
Journal of Applied Physics **124**, 240901 (2018); https://doi.org/10.1063/1.5048972

Potential clinical applications of terahertz radiation
Journal of Applied Physics **125**, 190901 (2019); https://doi.org/10.1063/1.5080205

Frontiers of magnetic force microscopy
Journal of Applied Physics **125**, 060901 (2019); https://doi.org/10.1063/1.5050712

AIP
Publishing

Journal of Applied Physics PERSPECTIVE scitation.org/journal/jap

Tomorrow's micromagnetic simulations

Cite as: J. Appl. Phys. **125**, 180901 (2019); doi: 10.1063/1.5093730
Submitted: 24 February 2019 · Accepted: 25 April 2019 ·
Published Online: 10 May 2019

View Online Export Citation CrossMark

J. Leliaert[a),b)] and J. Mulkers[a),c)]

AFFILIATIONS

Department of Solid State Sciences, Ghent University, 9000 Ghent, Belgium

[a)]**Contributions:** J. Leliaert and J. Mulkers contributed equally to this work.
[b)]**Electronic mail:** jonathan.leliaert@ugent.be
[c)]**Electronic mail:** jeroen.mulkers@ugent.be

ABSTRACT

Micromagnetic simulations are a valuable tool to increase our understanding of nanomagnetic systems and to guide experiments through parameter spaces that would otherwise be difficult and expensive to navigate. To fulfill this task, simulations have always pushed the limits of what is possible in terms of software and hardware. In this perspective, we give an overview of the current state of the art in micromagnetic simulations of ferromagnetic materials followed by our opinion of what tomorrow's simulations will look like. Recently, the focus has shifted away from exclusively trying to achieve faster simulations, toward extending pure micromagnetic calculations to a multiphysics approach. We present an analysis of how the performance of the simulations is affected by the simulation details and hardware specifications (specific to the graphics processing unit-accelerated micromagnetic software package mumax[3]), which sheds light on how micromagnetic simulations can maximally exploit the available computational power. Finally, we discuss how micromagnetic simulations can benefit from new hardware paradigms like graphics cards aimed at machine learning.

I. INTRODUCTION

Micromagnetism is the framework used to study magnetization phenomena on the intermediate scale between the quantum mechanical scale of individual atoms and the macroscale.[1] The nanometer length scale is relevant for applications such as nanomagnetic logic,[2,3] sensors,[4] spintronic logic devices,[5,6] and memory devices.[7,8] Within the micromagnetic framework, the magnetization is approximated by a continuous vector field with a constant magnitude, whose dynamics, governed by the phenomenological Landau–Lifshitz–Gilbert (LLG) equation,[9,10] take place on the picosecond time scale.

In order to model spintronic applications, the LLG equation is extended by additional torques describing the effect of spin currents on the magnetization.[11] These torques can be exerted by a spin-polarized current injected perpendicularly through a thin film,[12,13] or by a current running in the plane of the film where it gets polarized by the magnetic structure of the film itself.[14] Next to these spin-transfer torques, there also exist spin–orbit torques, named after the spin–orbit interactions in which they originate. The latter are especially relevant today because they are generated at the same interfaces[15] that give rise to the Dzyaloshinskii–Moriya interaction (DMI), making them a useful way to drive skyrmions through a thin film.

This practical importance of micromagnetism also recurs in the four motivations to perform micromagnetic research identified in Ref. 16 (1) to confirm or interpret experimental results, (2) to test or optimize a device design, (3) to predict new phenomena, and (4) to validate (approximate) analytical theories. Although the latter two show that micromagnetism is also a research topic in its own right, the former two motivations underscore its practical utility. For instance, when designing an expensive or time-consuming experiment, micromagnetic simulations need to be performed beforehand in order to navigate the parameter space that would otherwise be impossible to investigate in detail. An additional advantage is that simulations allow one to monitor quantities that are not accessible experimentally, like individual micromagnetic energy terms, which provides for a deeper understanding of the phenomena under study.

Before the advent of modern computers, micromagnetic research was limited to analytic derivations of equilibrium magnetization states under strong assumptions and to the simplification of the LLG dynamics to effective equations of motion, applicable to one specific system, like domain wall motion[9,17] or ferromagnetic resonance.[18] However, this approach is limited to a few very specific cases and is not sufficiently general to complement the results of complex experimental systems. The alternative is to use numerical techniques to minimize the free magnetic energy and to solve the LLG equation to simulate the magnetization dynamics. The first steps in this direction were taken by LaBonte and co-workers[19,20] by using a CDC1604 computer with only 192 kb of memory (and costing a whopping 1 million dollar in 1964) to find the equilibrium magnetization distribution of a domain wall. In these references, it was not stated how long this calculation took, but this following quote from the first dynamical micromagnetic simulation from Schryer and Walker[21] in 1974 shows that this was a very ambitious project:

> *At the outset it was not at all clear how much could be accomplished with the computer simulation with a reasonable investment of effort. The numerical integration of the equations of motion presented in itself an interesting and difficult problem. It quickly became apparent that unless the character of the motion was reasonably simple, the integration would not be economically feasible.*

To solve the LLG equation numerically, one has to approximate the continuum theory as a discrete problem. There are two well-developed approaches to realize this: (i) the finite difference[22] (FD) method in which field quantities are approximated by discrete values on a spatial grid and the derivatives by finite differences, (ii) the finite element[23] (FE) method in which the domain is subdivided into elements, and the field quantities are approximated using nodal basis functions.

Numerical micromagnetism is challenging, mainly because of complexities in calculating the effective magnetic field, which is derived from the different energies contributing to the magnetization dynamics. Landau and Lifshitz, who initiated the field of micromagnetism in their seminal paper,[9] took into account the magnetostatic, exchange, and magnetocrystalline anisotropy field contributions. Over the years, the LLG equation has been extended with additional contributions due to, e.g., the magnetoelastic interaction, thermal fluctuations, and higher-order exchange fields like the Dzyaloshinskii-Moriya interaction (DMI). For a complete overview of the many different contributions, we refer to reviews in the literature.[16,24–26] The magnetostatic interaction is computationally the most challenging contribution, because it is a long range interaction, taking into account the influence of the magnetization at any point on the magnetization at every other point. In a naive brute-force calculation, a simulation consisting of N discretization cells would require a number of calculations scaling as N^2. Depending on the implementation, this can be reduced to an $\mathcal{O}(N \log N)$ (Fast Fourier transform) or even $\mathcal{O}(N)$ (Fast Multipole Expansion) algorithm, although the latter suffers from such a large prefactor that it is less performant than the former for all but the largest simulations.

When modeling experimentally accessible systems, one faces two restrictions: first, individual simulations can become too time-consuming when the physical system under study is large (even though it may be no larger than a few microns), or displays dynamics over longer time scales. Second, the required number of simulations can be too large to finish in a reasonable amount of time, as is the case when considering stochastic dynamics, where many realizations are necessary to study the average behavior (or the outliers); or when modeling a system with multiple free parameters, in which the amount of simulations necessary to investigate the parameter space quickly becomes very large. Because a large share of simulations are performed to complement experimental work, there thus has always been a need to push the boundary of what is computationally feasible.

Despite the enormous advances in computer technology since Schryer and Walker's domain wall study in 1974, researchers still face the challenge of learning as much as possible from the simulations that can be performed on the finite computational resources available. Today, however, they have access to a number of user-friendly micromagnetic software packages, making this task easier than ever.

II. TODAY'S MICROMAGNETIC SIMULATIONS

Micromagnetic simulations, as we are used to them today, first appeared in 1998 with the release of the freely available Object Oriented MicroMagnetic Framework[27] (OOMMF). This package drastically lowered the threshold to perform micromagnetic simulations and still remains one of the most widely used and trusted packages, providing the golden standard in benchmark solutions to verify other codes.

A. General-purpose micromagnetic codes

Since the early 2000s,[28] numerical micromagnetism has advanced a lot, as evidenced by the development of numerous new micromagnetic codes. A (nonexhaustive) overview of the several codes with their main characteristics is presented in Table I. At their core, all of these software packages solve the LLG equation. They differ, however, in the algorithms applied to solve it, the hardware they run on, and the different physical phenomena they take into account (spin-transfer torques, spin–orbit torques, anisotropic magnetoresistance, thermal fluctuations, magnetostriction, spin diffusion, electric fields in multiferroics, etc.).

B. Specialized simulation packages

Next to general-purpose micromagnetic simulators shown in Table I, there also exist closely related, but more specialized packages, aimed at a narrower research domain. A first example are the atomistic spin simulation packages, such as Vampire[47] and Spirit.[48] In contrast to micromagnetic software packages, atomistic simulation packages take into account the magnetic moments of each individual atom or lattice site. However, similar to the magnetization field in the micromagnetic framework, the dynamics of these magnetic moments are governed by the LLG equation. The biggest benefit of using micromagnetic simulations over the more accurate atomistic simulations is that it uses less computational resources. If, however, the magnetization is not well described by a continuous field, e.g., when a Bloch point is involved, one is obliged to use the more computationally heavier atomistic approach. The atomistic approach is also well suited to the simulation of antiferromagnetic materials, or materials with nonstandard exchange interactions. Such materials can also be simulated using *ad hoc* adaptations to standard micromagnetic packages, e.g., by approximating the sublattices by two separate continuous fields in the case of antiferromagnets. Alternatively, there also exists (often nonshared) software developed with this purpose in mind but the field is much more fragmented and there exists no broadly adopted package yet.

TABLE I. List of general-purpose micromagnetic codes. The largest distinction between these packages is how they discretize space, which is either using a FD or FE approach. Next, we indicate whether they run exclusively on CPU, or are capable of using graphics cards (GPU's). Finally, it is indicated whether the codes are commercial or free software. Note that here *free* not only means that the codes can be used free of charge but also that the users have the freedom to view and change the source code, i.e., that it is open source software.

Name	Release	FE/FD	GPU capable?	Free?	References
LLG micromagnetics simulator	1997	FD	No	Commercial	29
OOMMF	1998	FD	No	Free	27
micromagus	2003[a]	FD	No	Commercial	30
magpar	2003	FE	No	Free	31
Nmag	2007	FE	No	Free	32
GPMagnet	2010	FD	Yes	Commercial	26
FEMME	2010	FE	No	Commercial	33
tetramag[b]	2010	FE	Yes	Commercial	34
finmag[c]	2011	FE	No	Free	35
Fastmag	2011	FE	Yes	Commercial	36
Mumax	2011	FD	Yes	Free	37
micromagnum	2012	FD	Yes	Free	38
magnum.fd[d]	2014	FD	Yes	Free	39
magnum.fe	2013	FE	No	Commercial	40
mumax^3	2014	FD	Yes	Free	41
LLG micromagnetics simulator v4.	2015	FD	Yes	Commercial	29
Grace	2015	FD	Yes	Free	42,43
OOMMF (GPU version)	2016	FD	Yes	Free	44
fidimag	2018	FD	No	Free	45
commics	2018	FE	No	Free	46

[a] v5.0.
[b] Will be succeeded by tetmag2.
[c] Succeeds Nmag.
[d] Succeeds micromagnum.

A second example are studies considering the longitudinal magnetization dynamics, e.g., for temperatures close to the Curie temperature. This case requires a more general framework than that of the LLG equation as the norm of the magnetization is no longer conserved, and Brown's stochastic field does not suffice to take temperature effects into account. Such studies have technological importance to correctly model heat-assisted magnetic recording[49] (HAMR). This is a technology that could greatly enhance the storage density of a magnetic hard drive by locally heating up a tiny surface to decrease its switching barrier and subsequently write data on smaller areas than in classical hard drives. There are a few different frameworks which can be used to investigate these dynamics like using renormalized cell sizes,[50] the Landau–Lifshitz–Bloch[51] equation and the Landau–Lifshitz–Baryakhtar equation.[52,53] Simulation using the latter approach can be performed using the graphics processing unit (GPU)-accelerated package, called hotspin,[54] developed to simulate the longitudinal magnetization dynamics at finite temperatures.

A third example of a related research field is magnetic nanoparticle dynamics. This topic, relevant for several biomedical applications,[55] differs from thin film micromagnetic simulations in several ways. First, due to their small size, nanoparticles are typically considered to have a uniform (monodomain) magnetization structure, thus removing the need to calculate the exchange interactions. Next, because the nanoparticles are typically suspended in a liquid, the simulated space is only sparsely filled, which makes calculating the magnetostatic interaction between particles very inefficient with FFT methods. Moreover, suspended particles also show mechanical dynamics, like rotations[56] and Brownian motion, which couple to the magnetization dynamics. Finally, due to the exponential size dependence of the switching rate of the nanoparticles, it can become necessary to implement the thermal switching as a jump noise process instead of considering the stochastic LLG equation. Both Magpy[57] and Vinamax[58] are CPU-based specialized packages with similar functionality, aiming to efficiently model such magnetic nanoparticle dynamics. Also, a custom-built GPU-code exists[59] but has not been publicly released.

C. Hardware platforms

Different hardware platforms have different (dis)advantages that should be taken into account when weighing the cost and total simulation time for a computational study. Generally, micromagnetic CPU codes are vastly slower than similar GPU codes and require expensive hardware to achieve decent performance levels.[24] This contrast with mumax^3 and other similar GPU codes, which already achieve high performance on relatively cheap gaming cards.[41]

GPUs have been developed since the 1980s and are designed specifically to generate visual images on a display. Current GPU's have far surpassed this original goal as their strength in performing many operations in parallel is exploited for general-purpose computing. The threshold to use GPU's in scientific calculations was significantly lowered in 2007 when NVIDIA released CUDA: an interface to make the accessibility of the GPU from software (relatively) easy. In the years following the CUDA release, the first GPU-accelerated micromagnetic software packages developed: it was first adopted in the commercial micromagnetic codes GPMagnet and shortly afterwards by tetramag. The first free GPU-accelerated micromagnetic code was mumax,[37] making the double-digit speedup offered by GPU calculations (as compared to their CPU counterparts) accessible to a broad userbase, including casual users trying to complement an experiment. As a side remark, it is interesting to note that there exist multiple free GPU-accelerated finite difference packages, but there still is no free GPU-accelerated finite element code, despite the fact that two of the first GPU-capable packages were finite element codes.

One restriction of the CUDA formalism is that it is tied to NVIDIA hardware, although this is not necessarily a disadvantage as NVIDIA is aware of their large user base in research and started to specifically target scientific computing with the development of specialized hardware. Nevertheless, vendor independent codes, based on OPENCL, are also available.[60]

If one is interested in only a few, relatively large simulations, the best option is probably to use GPU software. However, *most*

GPU clusters, accessible in an academic setting, contain at most a few tens of GPU's, whereas getting access to hundreds, if not thousands of CPU cores in a CPU cluster is relatively easy. This means that, despite the significantly longer time that each individual simulation will take, the total simulation time necessary for studies consisting of a large number of simulations can be optimized by running all of them in parallel on a CPU cluster. A third option that should be considered, especially when simulations are only sporadically used to complement experimental data is cloud computing, where one can use remote hardware on-demand. A provider, e.g., Amazon Web Services (AWS)[61] shares resources between all its users, who pay only for the computation time they use. Such a service has the advantage that one does not face the task of installing and maintaining a GPU workstation (let alone a cluster), and only spend time and resources on the actual calculations. Furthermore, using cloud services gives one access to several GPU's simultaneously, so that a batch of simulations can be run efficiently in parallel. A free tool (mucloud[62]) was developed, which allows users to run mumax^3 simulations in the cloud. In 2016, it turned out to be favorable to use cloud services as long as one needed less than 480 ± 30 h of the simulation time.[62] Today, one can use an NVIDIA Tesla K80 for 1 h for 0.90$ on AWS, offering a similar performance as a GTX 1080Ti, currently priced at 649 ± 50$. This means that it is favorable for users to use cloud services as long as they need less than 720 ± 60 h of the simulation time. Although this is significantly longer than in 2016, it remains a relatively small amount of simulation time. For now, we therefore conclude that the option of cloud computing is only economically interesting for sporadic users.

D. User interfaces

The last important property next to the speed, cost (of both the necessary hardware and software) and the capabilities of the different micromagnetic simulators, is their user-friendliness. This is mainly determined by the interface: the way in which the users interact with the program. Generally, micromagnetic packages run simulations based on input files, which gives users the freedom to program very complex simulations. Furthermore, this approach allows the simulation to be repeated in the future using the exact same input files, and to run the software on remote servers that are only accessible via a text-based terminal. The disadvantage of this approach is that users have to get familiar with the (sometimes complex) syntax of a particular software package. Nonetheless, with the rising popularity of the easy-to-learn language python, this became a lot simpler as almost all recent micromagnetic packages use a python interface. Moreover, it is possible to provide a python interface as an additional layer over the existing software. For instance, the realization of JOOMMF[63] that integrates OOMMF in Jupyter Notebook as a part of the OpenDreamKit[64] project is one example.

mumax^3 has no python interface as its input files have to be written in its own scripting language or in the Go[65] programming language. However, input files are not the only way to interact with software. Some packages, including mumax^3 and spirit,[48] also offer an intuitive graphical user interface (GUI), in which users can get a deeper understanding of their system by varying different parameters on the fly and following the magnetization dynamics as the simulation runs. Moreover, the quick feedback offered by such an interface allows one to efficiently optimize input files. In particular, researchers who are not primarily focusing on numerical studies, but use them as a tool to complement theoretical or experimental work can greatly benefit from the user-friendliness of a GUI. The main drawback of GUIs is the lack of repeatability, as each command command is executed manually. In mumax^3, this was addressed by saving all executed commands in a log file that can later serve as an input file to repeat the simulation.

III. PERFORMANCE

In this section, the performance of micromagnetic simulations running on GPUs will be discussed. Here, we focus on the performance as a function of the used GPU, the simulation size, and improvements in the NVIDIA libraries for GPU computing. As the authors are affiliated with the DyNaMat group at Ghent University, where mumax^3 was developed, this section will focus purely on the performance of this package. We do believe, however, that the trends in performance are representative for any finite difference GPU-accelerated micromagnetic software package and that the insights learned, e.g., on the influence of specific hardware properties on performance can be used to guide design choices in future software development.

A. GPU specifications

Many mumax^3 users have run the same performance test simulation (containing 2^{22} finite difference cells) on their machine and have send us the corresponding benchmarks.[66] Based on these benchmarks, we can compare the performance of mumax^3 for a wide range of NVIDIA GPUs. Because practically, all numerical operations are executed on the GPU, we only make a distinction by the GPU and do not focus on other specifications of the system when discussing the benchmark results. Figure 1 shows the GPU performance quantified by the achieved throughput in mumax^3 (number of cells evaluated per second). From this figure, it is immediately clear that the performance of mumax^3 heavily depends on the used GPU. The achieved throughput ranges from $\sim 10 \times 10^6$ cells/s for small mobile GPUs, such as the GeForce GT650M and the GeForce GT 755M, up to $\sim 600 \times 10^6$ cells/s for the ultra high-end Tesla V100.

Figure 1 also shows the memory bandwidth and the raw processing power as found in the manufacturer's specification reports. The clearly noticeable correlation between the throughput and the memory bandwidth shows that the memory bandwidth is the limiting factor for the performance, even more so than the raw processing power, as already clarified in Ref. 24. This is especially apparent for the Tesla P40 and the GeForce GTX 1080, which have a strong raw processing power compared to the memory bandwidth.

B. GPU prices

The prices of GPUs range from a few hundred dollars up to more than 10 000 dollars for the ultra high-end Tesla V100, the most expensive GPU on the list. For micromagnetic simulations, we see that the achieved throughput per dollar strongly differs from GPU to GPU. For example, the Tesla V100 has the best

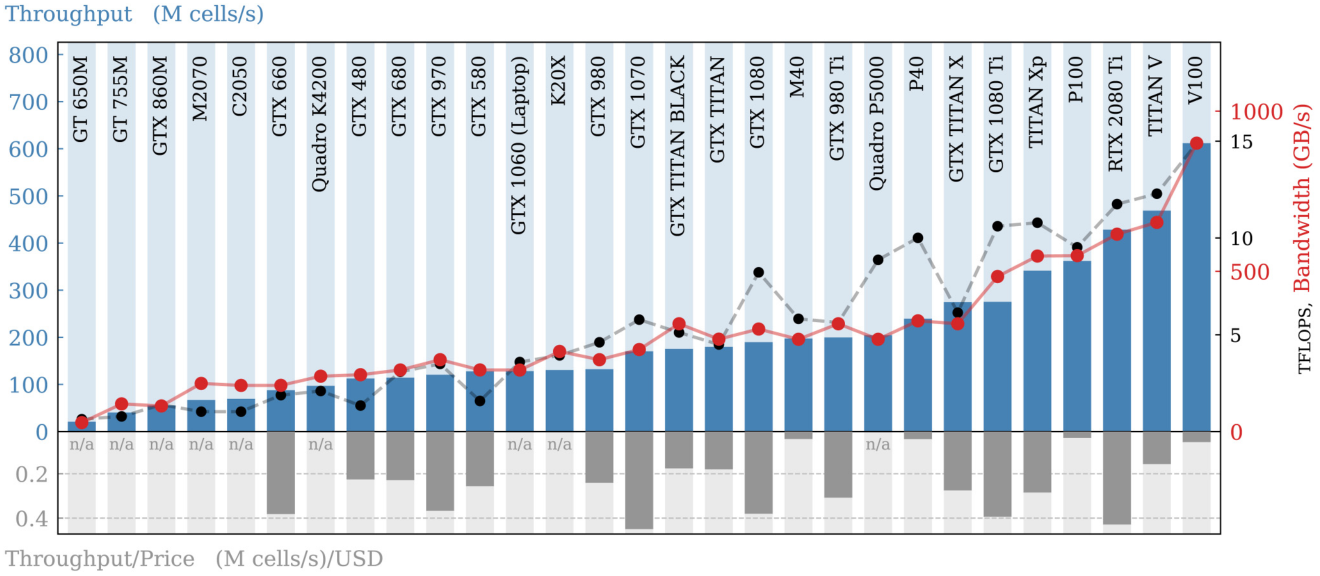

FIG. 1. mumax^3 throughput[67] (in 10^6 cells/s) for different NVIDIA GPU benchmarks provided by the mumax^3 community. The black and red curves show the processing power in TFLOPS (Tera Floating Point Operations Per Second) and the memory-bandwidth of these GPUs, respectively. The mumax^3 throughput per dollar is shown (gray bars, the bigger, the better) for the GPUs with a public manufacturer's suggested release price.

performance but scores rather low if we look at the performance per dollar. More recently released GPUs that offer good value for money include the GeForce RTX 2080 Ti, the GeForce GTX 1080 (Ti), and the GeForce GTX 1070. especially the latter can often found at discounted prices due to the release of the newer RTX series. Figure 1 also shows that the GPUs with the lowest performance are the GPUs found in mobile devices.

C. GPU performance over time

The performance of mumax^3 has increased rapidly over the last years due to the fact that GPUs are undergoing a rapid evolution. Figure 2 shows the evolution of the performance (again quantified by the achieved throughput in mumax^3) over the last eight years. On average, we see that the performance increases with 30% per year, a trend that is also followed when considering only the high-end cards.

D. Simulation size

The achieved throughput depends on the simulation size. To demonstrate this, we show in Fig. 3 the throughput as a function of the total number of finite difference cells for three different GPUs. We see that the throughput increases with the number of cells and reaches a maximum at $\sim 2^{16}$ cells for the Geforce GTX 1060/1080 and $\sim 2^{18}$ cells for the Titan V. For even larger simulations, we see that the throughput is slightly lower and, eventually, converges to a constant value. The appearance of a maximum throughput for a certain simulation size indicates again that the bandwidth is the predominant factor for large simulations.

E. Software improvements

Besides the increased hardware performance of the GPUs, the achieved throughput of mumax^3 has also increased due to improvements in third-party software. Let us, e.g., investigate the increased efficiency of the NVIDIA CUDA Fast Fourier Transform (cuFFT)

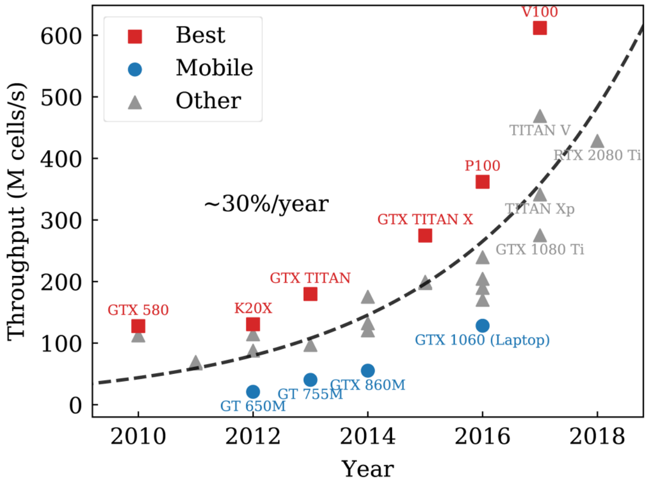

FIG. 2. mumax^3 throughput[67] as a function of the GPU release date. On average, the throughput increases with 30% every year. For the high-end GPUs (red squares), there is a similar increase of the throughput with a difference factor of approximately 1.6 compared to the average throughput.

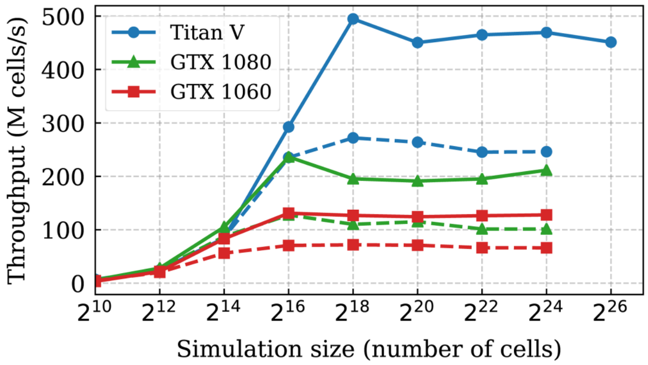

FIG. 3. The achieved throughput as a function of the simulation size for three different GPUs. In general, 2D simulations (full line) have a much larger throughput than 3D simulations (dashed line) with the same total number of cells.

library. mumax^3 uses the cuFFT library for the computation of the demagnetization field, the most time-consuming part of the simulation. Figure 4 shows the elapsed (wall) time of a single simulation step of a large simulation consisting out of $\sim 2^{16}$ cells for different CUDA (and cuFFT library) versions. Here, a distinction is made between a simulation with optimal grid dimensions ($2^8 \times 2^8 \times 1$) and ill-chosen grid dimensions ($241 \times 271 \times 1$) consisting out of large primes. Despite the fact that both grids contain a similar total number of cells, we see that the wall time[68] per step is almost three times lower for the optimal grid dimensions than for the ill-considered dimensions, explained by a much more efficient FFT algorithm for grid dimension with low prime factors. Furthermore, we see that the FFT algorithms have become more efficient with the release of new CUDA versions.

IV. TOMORROW'S MICROMAGNETIC SIMULATIONS

A. Ongoing trends

It is clear from the scope of the recently released micromagnetic software packages that the field is continuing its evolution toward a multiphysics approach, where an increasing number of physical phenomena, coupling to the magnetization dynamics, can be taken into account. Some packages, like FinMag (uses the dolFIN interface to FEniCS)[35] or OMNes (Open source Multiphysics Numerical Simulation)[69] are specifically aimed at multiphysics modeling, but the same trend is also visible in the addition of new capabilities to the existing packages. Notable examples for mumax^3 are electric field-driven domain wall motion,[70] magnetic friction,[71] and exchange bias.[72,73] Moreover, this multiphysics approach will extend itself beyond the scope of ferromagnets toward the research of dynamics in antiferromagnets or ferrimagnets, for which the current simulation capabilities are still limited.

A second trend that we believe will continue is the decreasing threshold to use scientific software thanks to the increased user-friendliness of the interfaces (e.g., the very clean GUI of spirit[48] and possibilities to integrate the numerical results within one's workflow. The current state-of-the art in this respect is JOOMMF,[63] allowing to combine an entire scientific study, including simulation details, output, data processing, plotting, and annotations into one web-based document.

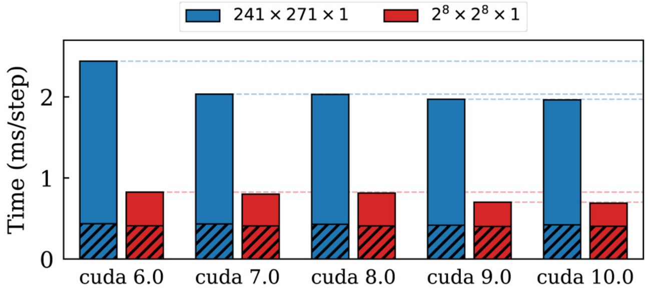

FIG. 4. The execution time per step for different CUDA (and cuFFT) versions and two different grid dimensions. The hatched part denotes the time per step unrelated to the FFT.

B. Simulation topics

Numerical micromagnetic research will likely remain an important tool within current research topics like (i) mobility (including diffusion and creep) studies of quasiparticles like vortex cores domain walls and skyrmions through (chiral) thin films, (ii) spin waves and magnonics, and (iii) spintronics and spinorbitronics.

Within these mature topics and following the trend toward multiphysics approaches, we expect that the gap between numerical modeling and experiment will be further reduced. This can be achieved by including all necessary effects to be able to compare numerical results quantitatively to their experimental counterparts. For instance in spintronics research, one could further generalize or increase the complexity of the already existing models[74] which take into account different dynamics at play, and a few of which even solve the electric currents self-consistently using Ohm's law, e.g., magnum.fe.[75] Other studies might benefit more by the inclusion of realistic disorder[76,77] or thermal fluctuations[53,78] in their models.

Next to a continuation of today's topics of interest, the evolution of tomorrow's micromagnetic simulations is guided by emerging trends in magnetism research. One example that currently draws a lot of attention is neuromorphic computing.[79]

C. Neuromorphic computing

Neuromorphic circuits mimick biological architectures, like the human brain, to perform computational tasks. Typically, they consist of a large number of small circuits that simulate a simple neuron (processing unit) or synapse (memory). In applications like machine learning, software implementations of neuromorphic algorithms have already proven to work. However, this approach is not optimal because such software runs on hardware that is optimized for classical, von Neumann computing, where memory and processing units are much less connected. In order to operate in a fast and energy-efficient way, neuromorphic circuits should be built on dedicated hardware, like spin-torque oscillators[80] or memristors.[81] Because of the inherently stochastic nature[82] of the spin-torque oscillators, the development of such devices requires high throughput micromagnetic simulations. A single spin-torque oscillator can be as small as 100 nanometer in diameter (requiring as little as $\approx 64 \times 64 = 2^{12}$ finite difference cells). However, considering Fig. 3, it is clear that this makes very inefficient use of the GPU, as only 10% of the maximal simulation throughput is used.

D. Parallel simulations

As discussed in Sec. III, the performance of mumax^3 is mainly limited by the memory bandwidth of the GPU. Extrapolating from the trend of the past 10 years, we predict that this property will continue to improve by a factor of about 30% per year, leading to another expected quadrupling of the performance in the coming 5 years. However, most of these performance gains are realized at very large numbers of cells. Therefore, it can be more efficient to simulate several devices at once. In the method presented in Ref. 83, the magnetostatic field was cut off at a distance to uncouple different devices in one simulation. This approach provides a speedup, but even though the demagnetizing kernel is cut short, it still has the drawback that an FFT [scaling as $\mathcal{O}(N \log N)$] has to be taken over the entire simulated volume. A different approach that would offer a significantly larger speedup is to split the simulation into multiple FFT's, each taken over different magnetostatically uncoupled regions, in order to simulate multiple spin-torque oscillators. This would allow to exploit the full computational power of the GPU by choosing the total simulation size at its peak performance.

E. New hardware developments

The trend toward non von Neumann computing is of course not restricted to magnetic research and neuromorphic computing, but has been ongoing in artificial intelligence (AI) research as well. An exciting new hardware development are tensor cores[84] that are specifically developed to massively speedup deep learning[85] calculations. These tensor cores perform a matrix multiply and accumulate (MAC) operation where it multiplies two half precision 4×4 matrices into a single precision result, to which it can add another 4×4 single precision matrix. The MAC operation is the work horse of deep learning. Compared to the single precision implementation in CUDA, its mixed precision implementation offers increased performance because it allows to reduce both the memory bandwidth and the the arithmetic bandwidth.[86]

Next to their main application in AI, they also show potential speedups in other calculations.[87] The main drawback of tensor cores is that they work in mixed precision, whereas mumax^3 works in single precision. Although single precision suffices for almost any micromagnetic calculation, it is clear that half precision calculations do not provide the necessary accuracy.

However, it has been shown that a single precision matrix multiplication can be split up into two half precision calculation in order to perform an FFT on tensor cores without loss of accuracy.[88] The presented proof of concept was slower than the cuFFT implementation in CUDA, but hopefully this will be mitigated in the future by a more efficient implementation so micromagnetic calculations can also benefit from the computational capabilities of tensor cores.

V. CONCLUSIONS

Over the last 20 years, the field of numerical micromagnetics was (and will remain) evolving toward a multiphysics approach with the Landau–Lifshitz–Gilbert equation at its core. Currently, there exist many different software packages which each contain one or more extensions to include different physical phenomena, but there still is not one general-purpose true multiphysics package which has a very broad scope, yet offers the same performance as current true micromagnetic packages, although efforts to address this challenge are underway. A second ongoing trend is that micromagnetic software packages become increasingly user-friendly with nice GUI's and input files with an easy syntax. This is especially helpful in situations where micromagnetic simulations are used to support or bridge theory and experiment in fundamental of technologically driven applied research.

Regarding the simulation topics, we see that next to the continuation of the classical research fields, the broader trend toward non-von Neumann computing bring both exciting new research due to the application of, e.g., spin-torque oscillators but also new hardware, like tensor cores, that could offer large performance increases.

Additionally, we presented data that show that mumax^3 performance is mainly limited by the memory-bandwidth of the graphics cards. Over the last decade, this specification improved at an an exponential rate of about 30% per year over the last decade, and we expect this trend to continue.

On the whole, one can conclude that numerical micromagnetics will continue its role as the tool of choice to propel nanoscale magnetic research.

ACKNOWLEDGMENTS

This work was supported by the Fonds Wetenschappelijk Onderzoek (FWO-Vlaanderen) through Project No. G098917N (J.M.) and with a postdoctoral fellowship (J.L.). We sincerely thank everyone who shared benchmark figures for the different GPU's, and we gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU's used for this research.

REFERENCES

[1] W. F. Brown, *Micromagnetics* (Interscience Publishers, New York, 1963).
[2] P. Gypens, J. Leliaert, and B. Van Waeyenberge, Phys. Rev. Appl. **9**, 034004 (2018).
[3] A. Imre, G. Csaba, L. Ji, A. Orlov, G. Bernstein, and W. Porod, Science **311**, 205 (2006).
[4] D. Suess, A. Bachleitner-Hofmann, A. Satz, H. Weitensfelder, C. Vogler, F. Bruckner, C. Abert, K. Prügl, J. Zimmer, C. Huber *et al.*, Nat. Electron. **1**, 362 (2018).
[5] D. A. Allwood, G. Xiong, C. C. Faulkner, D. Atkinson, D. Petit, and R. P. Cowburn, Science **309**, 1688–1692 (2005). http://www.sciencemag.org/cgi/reprint/309/5741/1688.pdf.
[6] D. Pinna, F. Abreu Araujo, J.-V. Kim, V. Cros, D. Querlioz, P. Bessiere, J. Droulez, and J. Grollier, Phys. Rev. Appl. **9**, 064018 (2018).
[7] S. Parkin and S.-H. Yang, Nat. Nanotechnol. **10**, 195 (2015).
[8] A. Fert, V. Cros, and J. Sampaio, Nat. Nanotechnol. **8**, 152 (2013).
[9] L. Landau and E. Lifshitz, Phys. Z. Sowietunion **8**, 14 (1935).
[10] T. Gilbert, IEEE Trans. Magn. **40**, 3443 (2004).

[11] C. Abert, "Spintronics in micromagnetics," in *Handbook of Materials Modeling: Methods: Theory and Modeling*, edited by W. Andreoni and S. Yip (Springer International Publishing, Cham, 2019), pp. 1–24.
[12] L. Berger, Phys. Rev. B **54**, 9353 (1996).
[13] J. Slonczewski, J. Magn. Magn. Mat. **159**, L1 (1996).
[14] S. Zhang and Z. Li, Phys. Rev. Lett. **93**, 127204 (2004).
[15] A. Brataas and K. M. Hals, Nat. Nanotechnol. **9**, 86 (2014).
[16] D. Kumar and A. O. Adeyeye, J. Phys. D Appl. Phys. **50**, 343001 (2017).
[17] W. Döring, Z. Naturforsch. A **3**, 373 (1948).
[18] C. Kittel, Phys. Rev. **73**, 155 (1948).
[19] W. F. Brown, Jr. and A. E. LaBonte, J. Appl. Phys. **36**, 1380 (1965).
[20] A. LaBonte, J. Appl. Phys. **40**, 2450 (1969).
[21] N. L. Schryer and L. R. Walker, J. Appl. Phys. **45**, 5406 (1974).
[22] T. Schrefl, G. Hrkac, S. Bance, D. Suess, O. Ertl, and J. Fidler, "Numerical methods in micromagnetics (finite element method)," in *Handbook of Magnetism and Advanced Magnetic Materials* (American Cancer Society, 2007).
[23] J. E. Miltat and M. J. Donahue, *Handbook of Magnetism and Advanced Magnetic Materials* (Wiley-Interscience, 2007), Vol. 2, p. 14.
[24] J. Leliaert, M. Dvornik, J. Mulkers, J. De Clercq, M. V. Milošević, and B. V. Waeyenberge, J. Phys. D Appl. Phys. **51**, 123002 (2018).
[25] H. Kronmüller and M. Fähnle, *Micromagnetism and the Microstructure of Ferromagnetic Solids*, Cambridge Studies in Magnetism (Cambridge University Press, 2003).
[26] L. Lopez-Diaz, D. Aurelio, L. Torres, E. Martinez, M. A. Hernandez-Lopez, J. Gomez, O. Alejos, M. Carpentieri, G. Finocchio, and G. Consolo, J. Phys. D Appl. Phys. **45**, 323001 (2012).
[27] M. J. Donahue, Interagency report NISTIR 6376, NIST, Gaithersburg, MD, 1999.
[28] J. Fidler and T. Schrefl, J. Phys. D Appl. Phys. **33**, R135 (2000).
[29] See http://llgmicro.home.mindspring.com/LLGEnvFrame.htm for more information about LLG Micromagnetic Simulator.
[30] See http://www.micromagus.de for more information about MicroMagus.
[31] W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, and V. Tsiantos, Comput. Mater. Sci. **28**, 366 (2003).
[32] T. Fischbacher, M. Franchin, G. Bordignon, and H. Fangohr, IEEE Trans. Magn. **43**, 2896 (2007).
[33] See http://www.suessco.com/simulations/femme/ for more information about FEMME.
[34] A. Kakay, E. Westphal, and R. Hertel, IEEE Trans. Magn. **46**, 2303 (2010).
[35] M.-A. Bisotti, M. Beg, W. Wang, M. Albert, D. Chernyshenko, D. Cortés-Ortuño, R. A. Pepper, M. Vousden, R. Carey, H. Fuchs, A. Johansen, G. Balaban, L. Breth, T. Kluyver, and H. Fangohr, FinMag: Finite-element micromagnetic simulation tool, 2018.
[36] R. Chang, S. Li, M. V. Lubarda, B. Livshitz, and V. Lomakin, J. Appl. Phys. **109**, 07D358 (2011).
[37] A. Vansteenkiste and B. V. de Wiele, J. Magn. Magn. Mater. **323**, 2585 (2011).
[38] C. Abert, G. Selke, B. Kruger, and A. Drews, IEEE Trans. Magn. **48**, 1105 (2012).
[39] See http://micromagnetics.org/magnum.fd for more information about magnum.fd.
[40] C. Abert, L. Exl, F. Bruckner, A. Drews, and D. Suess, J. Magn. Magn. Mater. **345**, 29 (2013).
[41] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. Van Waeyenberge, AIP Adv. **4**, 107133 (2014).
[42] R. Zhu, SoftwareX **3–4**, 27 (2015).
[43] R. Zhu, AIP Adv. **6**, 056401 (2016).
[44] S. Fu, W. Cui, M. Hu, R. Chang, M. J. Donahue, and V. Lomakin, IEEE Trans. Magn. **52**, 1 (2016).
[45] M.-A. Bisotti, D. Cortés-Ortuño, R. Pepper, W. Wang, M. Beg, T. Kluyver, and H. Fangohr, J. Open Res. Softw. **6**, 22 (2018).
[46] C.-M. Pfeiler, M. Ruggeri, B. Stiftner, L. Exl, M. Hochsteger, G. Hrkac, J. Schöberl, N. J. Mauser, and D. Praetorius, e-print arXiv:1812.05931 (2018).
[47] R. F. L. Evans, W. J. Fan, P. Chureemart, T. A. Ostler, M. O. A. Ellis, and R. W. Chantrell, J. Phys. Condens. Matter **26**, 103202 (2014).
[48] G. P. Müller, M. Hoffmann, C. Disselkamp, D. Schürhoff, S. Mavros, M. Sallermann, N. S. Kiselev, H. Jónsson, and S. Blügel, e-print arXiv:1901.11350 [cond-mat.mes-hall] (2019).
[49] H. F. Hamann, Y. C. Martin, and H. K. Wickramasinghe, Appl. Phys. Lett. **84**, 810 (2004).
[50] G. Grinstein and R. Koch, Phys. Rev. Lett. **90**, 207201 (2003).
[51] U. Atxitia, D. Hinzke, and U. Nowak, J. Phys. D Appl. Phys. **50**, 033003 (2016).
[52] V. Baryakhtar, Zh. Eksp. Teor. Fiz. **87**, 1501 (1984).
[53] M. Dvornik, A. Vansteenkiste, and B. V. Waeyenberge, Appl. Phys. Lett. **105**, 162411 (2014).
[54] See http://github.com/godsic/hotspin for more information about hotspin.
[55] Q. Pankhurst, N. Thanh, S. Jones, and J. Dobson, J. Phys. D Appl. Phys. **42**, 224001 (2009).
[56] K. D. Usadel and C. Usadel, J. Appl. Phys. **118**, 234303 (2015).
[57] O. Laslett, J. Waters, H. Fangohr, and O. Hovorka, preprint arXiv:1801.06073 (2018).
[58] J. Leliaert, A. Vansteenkiste, A. Coene, L. Dupré, and B. Van Waeyenberge, Med. Biol. Eng. Comput. **53**, 309 (2015).
[59] D. Cabrera, A. Lak, T. Yoshida, M. Materia, D. Ortega, F. Ludwig, P. Guardia, A. Sathya, T. Pellegrino, and F. Teran, Nanoscale **9**, 5094 (2017).
[60] F. G. Khan, B. Montrucchio, B. Jan, A. N. Khan, W. Jadoon, S. Shamshirband, A. T. Chronopoulos, and I. A. Khan, Concurrency Comput. Pract. Exp. **29**, e3981 (2017).
[61] See https://aws.amazon.com/ for "Amazon web services."
[62] C. Jermain, G. Rowlands, R. Buhrman, and D. Ralph, J. Magn. Magn. Mater. **401**, 320 (2016).
[63] M. Beg, R. A. Pepper, and H. Fangohr, AIP Adv. **7**, 056025 (2017).
[64] See https://opendreamkit.org/index.html for "Opendreamkit project."
[65] See https://golang.org/ for "The go programming language."
[66] See http://github.com/mumax/3/tree/master/bench for more information about the mumax3 benchmark simulation script.
[67] mumax^{3}calculates in single precision.
[68] "Wall time" or "elapsed real time" is the total time spent between the start and completion of a computational task. It is more general than, e.g., CPU time, which only monitors the time that the CPU spent on the task and does not take any overhead due to, e.g., memory transfer into account.
[69] See https://www.omnes.uni-mainz.de/ for more information about OMNeS.
[70] B. Van de Wiele, J. Leliaert, K. J. A. Franke, and S. van Dijken, New J. Phys. **18**, 033027 (2016).
[71] I. Rissanen and L. Laurson, Phys. Rev. E **97**, 053301 (2018).
[72] J. De Clercq, A. Vansteenkiste, M. Abes, K. Temst, and B. V. Waeyenberge, J. Phys. D Appl. Phys. **49**, 435001 (2016).
[73] J. De Clercq, J. Leliaert, and B. Van Waeyenberge, J. Phys. D Appl. Phys. **50**, 425002 (2017).
[74] C. Abert, preprint arXiv arXiv:1810.12365 (2018).
[75] See http://micromagnetics.org/magnum.fe/spin_diffusion.html for more information about self-consistent calculations of electric current flow using magnum.fe.
[76] J. Lau, R. McMichael, and M. Donahue, J. Res. Natl. Inst. Stand. Technol. **114**, 57–67 (2009).
[77] J. Leliaert, B. Van de Wiele, A. Vansteenkiste, L. Laurson, G. Durin, L. Dupré, and B. Van Waeyenberge, J. Appl. Phys. **115**, 17D102 (2014).
[78] J. Leliaert, J. Mulkers, J. De Clercq, A. Coene, M. Dvornik, and B. V. Waeyenberge, AIP Adv. **7**, 125010 (2017).
[79] J. Grollier, S. Guha, H. Ohno, and I. K. Schuller, J. Appl. Phys. **124**, 151801 (2018).
[80] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, Nature **547**, 428 (2017).
[81] S. Lequeux, J. Sampaio, V. Cros, K. Yakushiji, A. Fukushima, R. Matsumoto, H. Kubota, S. Yuasa, and J. Grollier, Sci. Rep. **6**, 31510 (2016).
[82] K. Roy, A. Sengupta, and Y. Shim, J. Appl. Phys. **123**, 210901 (2018).
[83] D. Berkov and E. Semenova, "Massively parallel micromagnetic simulations of small-size nanoelements with applications to STT-MRAM devices," in *JEMS conference, Book of Abstracts*, presented at the 9th Joint European Magnetic symposia (JEMS 2018).

[84] See https://www.nvidia.com/en-us/data-center/tensorcore/ for "Nvidia tensor cores."
[85] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
[86] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," e-print arXiv:1710.03740 [cs.AI] (2017).
[87] S. Markidis, S. W. D. Chien, E. Laure, I. B. Peng, and J. S. Vetter, in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (IEEE, 2018), pp. 522–531.
[88] A. Sorna, X. Cheng, E. D'Azevedo, K. Won, and S. Tomov, in *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)* (IEEE, 2018), pp. 3–7.