

ML/DL/HPC Ecosystem of the HybriLIT Heterogeneous Platform (MLIT JINR): New Opportunities for Applied Research

Yu. Butenko,^a M. Ćosić,^b A. Nechaevskiy,^a D. Podgainy,^a I. Rahmonov,^a A. Stadnik,^a O. Streltsova^a and M. Zuev^{a,*}

^a*International Intergovernmental Organization Joint Institute for Nuclear Research,
6, Joliot-Curie st., Dubna, Moscow Region, 141980, Russia*

^b*Vinča Institute Of Nuclear Sciences, University of Belgrade,
12-14 Mike Petrovića Alasa, Vinča, Beograd, 11351, Srbija*

*E-mail: butenko@jinr.ru, mcosic@vinca.rs, nechav@jinr.ru,
podgainy@jinr.ru, rahmonov@theor.jinr.ru, stadnik@jinr.ru, strel@jinr.ru,
zuevmax@jinr.ru*

The work presents the possibilities for using the ML/DL/HPC ecosystem deployed on the HybriLIT Heterogeneous Platform (Meshcheryakov Laboratory of Information Technologies JINR) on top of JupyterHub, which provides opportunities for solving tasks not only in the field of machine learning and deep learning, but also for the convenient organization of calculations and scientific visualization. The ecosystem allows one to develop and implement program modules in Python, as well as to carry out methodical computations. The relevance of deploying such an environment is primarily associated with the great demand for software modules that are provided to a group of researchers or the scientific community, when all stages of the study can be reproduced; the code has been modified and used by the scientific community. Using the example of solving a specific problem to study the dynamics of magnetization in a Phi-0 Josephson Junction (Superconductor-Ferromagnet-Superconductor structure), a methodology for developing software modules is presented; it enables not only to carry out calculations, but also to visualize the results of the study and accompany them with the necessary formulas and explanations. The possibility of parallel implementation of the algorithm for performing computations for various values of parameters of the model based on the Joblib Python library is shown, and the results of computational experiments demonstrating the efficiency of parallel data processing are presented.

*** *The 6th International Workshop on Deep Learning in Computational Physics (DLCP2022)* ***

*** *6-8 July 2022* ***

*** *JINR, Dubna, Russia* ***

*Speaker

1. Introduction

To accelerate the study of systems based on Josephson junctions, which are of critical importance for modern microelectronics, parallel implementations of algorithms are developed to solve the corresponding systems of nonlinear differential equations using parallel programming technologies. To provide a convenient research environment, information systems that allow calculations to be carried out on computing platforms with a hybrid architecture and in cloud environments are elaborated. [1–4] A distinctive feature of such information systems is the aggregation of the developed models, which allow users to change various parameters and then perform calculations in batch mode. However, at the stage of research or for the formation of laboratory work for students, it is of interest to be able to both modify the equations and sequentially reproduce all stages of calculations: to visually see the equations, to build all intermediate graphs and display the necessary intermediate calculations. This opportunity is provided by an environment built on top of the Jupyter Notebook ecosystem [5, 6], with the Python programming language, which is actively developed and includes a wide range of mathematical libraries, libraries for scientific visualization, and approaches to algorithm parallelization. It should be noted that Jupyter Notebook is now actively used for machine and deep learning tasks, and the accumulated tools can be used more widely for scientific research that requires, among other things, numerical computations. This approach is demonstrated in the work by an example of solving the task of studying the possibility of reversing the magnetic moment in superconductor-ferromagnet-superconductor structures, in which spin-orbit interaction in a ferromagnet provides a mechanism for a direct connection between the magnetic moment and the Josephson current; such Josephson junctions are called φ_0 -junctions. [7] It is noteworthy that the study of such systems is associated with numerous computational experiments for various models and for a large number of model parameters. To speed up calculations, the development of a parallel implementation in Python is of particular relevance.

2. ML/DL/HPC Ecosystem

The HybriLIT Heterogeneous Computing Platform [2] is part of the Multifunctional Information and Computing Complex of the Meshcheryakov Laboratory of Information Technologies of JINR, Dubna. [8] The platform consists of the "Govorun" supercomputer [9], the HybriLIT education and testing polygon [10], and the ML/DL/HPC ecosystem [11] for machine learning, deep learning and data analysis.

The HybriLIT heterogeneous computing platform ecosystem includes the following computing components:

- jhub.jinr.ru, a virtual machine designed to develop models and algorithms based on Jupyter-Hub, a multi-user platform for working with Jupyter Notebook: KVM vCPU (2.6 Ghz, 24 cores), 128 GB RAM;
- jhub2.jinr.ru, for resource-intensive, massively parallel calculations, including those using NVIDIA graphics accelerators: $2 \times$ Intel Xeon Gold 6148 (2.4 Ghz, 20 cores), $4 \times$ NVIDIA Volta V100, 512 GB RAM;

- jlabhpc.jinr.ru, a virtual machine for performing calculations on the compute nodes of the HybriLIT heterogeneous platform using the SLURM task scheduler, for developing applications, for analyzing data and visualizing the results obtained: Intel Xeon Gold 6126 (2.6 Ghz, 24 cores), 32 GB RAM;
- studhub.jinr.ru, for conducting lectures, tutorials and master classes for JINR students, post-graduates and researchers: $2 \times$ Intel Xeon Gold 6152 (2.1 Ghz, 22 cores), 512 GB RAM.

On all components of the ecosystem, the main libraries and frameworks for machine and deep learning tasks are installed, the development of algorithms and research are carried out on the basis of Jupyter Notebook.

Jupyter Notebook is a Python programming environment with the ability to work with individual Jupyter notebooks, text files, images of various formats, datasets. In the notebook, it is possible to break the code into separate fragments, written in the form of separate cells, and execute them in any order. The result of running each individual cell is displayed in this notebook. The Jupyter Notebook interface opens in the default browser. Any additional libraries not included with the base version of Jupyter can be installed while working with notebooks.

The prepared notebook file can be saved in a convenient format, such as HTML or PDF. If one needs to recalculate with new initial parameters, it is necessary to install the Jupyter environment through the Anaconda [12] package manager on a working machine, to open the resulting file in it and run all the cells or individual parts of the code in it.

It is also possible to generate notebook reports containing, in addition to cells with the code, cells in Markdown [13] format comprising a description of the problem being solved with formulas, accompanying figures and graphs. In addition to separate files, notebooks can be organized into separate Internet resources with a unique link, by clicking on which the user can read the description of the problem, copy the necessary code and run it on his working machine. [14]

Within the ecosystem, one can run calculations in the MATLAB package using the Jupyter-Hub interface. [15]

Access to the components of the ML/DL/HPC ecosystem is provided to registered users of the HybriLIT heterogeneous platform.

3. Statement of the physical problem

As an example of working in the Jupyter Notebook environment, let us consider the problem of calculating time dependences of magnetic moment components for various values of the φ_0 -junction parameters, on the basis of which it is possible to set the parameter intervals where it flips from $m_z = 1$ to $m_z = -1$.

The basic equations are presented in [7]. Below is the Cauchy problem for the system of equations in a dimensionless form.

The dynamics of the magnetic moment of the system under consideration is described by the

Landau-Lifshitz-Gilbert equation:

$$\begin{aligned}\frac{dm_x}{dt} &= -\frac{1}{1+M^2\alpha^2} \{m_y H_z - m_z H_y + \alpha [m_x (M, H) - H_x]\}, \\ \frac{dm_y}{dt} &= -\frac{1}{1+M^2\alpha^2} \{m_z H_x - m_x H_z + \alpha [m_y (M, H) - H_y]\}, \\ \frac{dm_z}{dt} &= -\frac{1}{1+M^2\alpha^2} \{m_x H_y - m_y H_x + \alpha [m_z (M, H) - H_z]\},\end{aligned}\quad (1)$$

where $M = [m_x, m_y, m_z]$ are the components of the magnetic moment; the components of the effective field $H = [H_x, H_y, H_z]$ depend on the Josephson phase difference φ and are defined as follows:

$$\begin{aligned}H_x(t) &= 0, \\ H_y &= Gr \sin(\varphi(t) - rm_y(t)), \\ H_z(t) &= m_z(t),\end{aligned}$$

where the component y depends on the Josephson phase difference. The equation for the Josephson phase difference φ is determined within the resistive model from the expression for the electric current I through the Josephson junction measured in the units of the critical current I_c :

$$I = \omega \left(\frac{d\varphi}{dt} - r \frac{dm_y}{dt} \right) + \sin(\varphi - rm_y),$$

namely:

$$\frac{d\varphi}{dt} = -\frac{1}{\omega} \left(\sin(\varphi - rm_y) + r \frac{dm_y}{dt} \right) + \frac{1}{\omega} I, \quad (2)$$

with the following model parameters: G is the ratio of the Josephson energy to the magnetic anisotropy energy, r is the spin-orbit coupling constant, α is the Gilbert dissipation, ω is the ratio of the ferromagnetic resonance frequency to the characteristic frequency of the Josephson junction. In our calculations, it is assumed to be equal to 1.

The initial conditions presuppose that all components of the magnetic moment, except for m_z , are equal to zero:

$$m_x(0) = 0, m_y(0) = 0, m_z(0) = 1, \varphi(0) = 0. \quad (3)$$

In the problem under consideration, the possibility of reversing the magnetic moment m_z under the action of the rectangular current pulse $I = I(t)$ included in equation 2 is studied:

$$I = I(t) = \begin{cases} 0, & t \in [0, T_f - \Delta t/2] \\ A_s, & t \in [T_s - \Delta t/2, T_s + \Delta t/2] \\ 0, & t \in [T_s + \Delta t, T_f] \end{cases} \quad (4)$$

where A_s is the pulse amplitude, Δt is its duration, T_s is the time of the beginning of its localization, and T_f is the computation time, which is chosen to be large enough to enter stable mode.

4. Implementation of the computational scheme

To study the possibility of the magnetic reversal depending on the parameters of the model, it is necessary to numerically solve the Cauchy problem for the system of nonlinear equations (1-2) with initial conditions (3) for a given time dependence of the current (4).

The numerical solution of the initial problem was carried out using the `scipy.integrate.solve_ivp` function of the SciPy library, in which it is possible to choose the solution method: for nonstiff problems, explicit Runge-Kutta methods "RK45" or "RK23" of orders 5(4) and 3(2), respectively, are available, while methods for solving stiff problems are "Radau", i.e., the implicit Runge-Kutta method of the Rado family IIA of the 5th order, the implicit multi-step method of the variable order "BDF", etc. [16] The results presented in the article were obtained using the "RK45" and "BDF" methods.

The numerical solution of the Cauchy problem was carried out for various values of the model parameters. Fig. 1 shows the dynamics of the reversal of the magnetic moment component m_z for two values of the ratio of the Josephson energy to the magnetic anisotropy energy G : at $G = 9$ there is a reversal of the magnetic moment m_z , and at $G = 8$ there is no reversal. Fig. 1 also demonstrates a plot of the applied pulsed current. The calculations were performed with the values of the parameters $r = 0.1$, $\alpha = 0.1$.

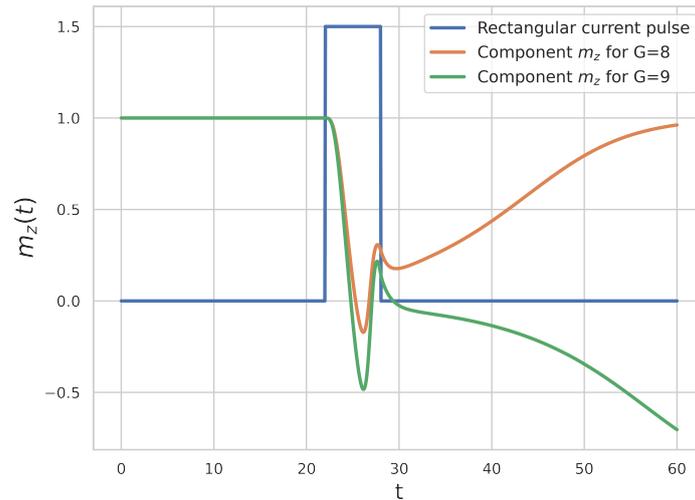


Figure 1: Dynamics of the reversal of the magnetic moment m_z for two values of the ratio of the Josephson energy to the magnetic anisotropy energy G and plot of the pulsed current.

The study shows that the reversal of the magnetic moment is periodic, which can be demonstrated by performing calculations with small steps in the parameter plane, for example, (G, α) , which is fully consistent with the results obtained in [7].

Fig. 2 presents a graph of the frequency of the reversal of the magnetic moment m_z with an increase in the value of the parameter G in the plane (G, α) . The results were obtained for the values of the parameters: $\alpha = 0.01 \dots 0.11$ with a step 0.001 and $G = 1 \dots 101$ with a step 1.

It should be noted that carrying out numerous calculations for various parameters requires a lot of time, for example, to obtain the results presented in Fig. 2, it is necessary to solve 100^2 problems of the same type and write the result (positive or negative value of the component m_z

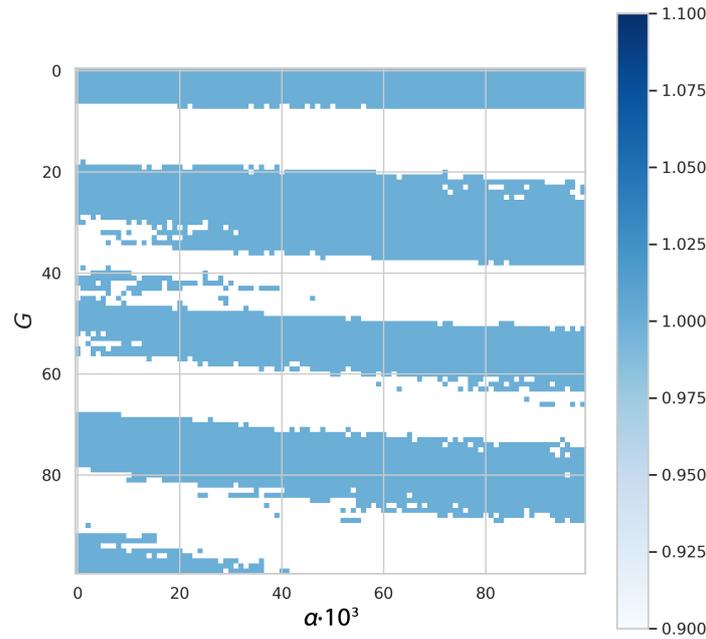


Figure 2: Graph of the frequency of the reversal of the magnetic moment m_z in the plane of the parameters (G, α) .

at the computation time T_f) into the final matrix of values. The uniformity of the problems being solved makes it possible to parallelize the data by distributing solutions to Cauchy problems for different values of the parameters among different processes or threads. One of possible parallel implementations is presented below.

5. Parallel implementation of the algorithm

To speed up computations, in the Python language there are both separate libraries that allow performing calculations on central processors and graphics accelerators: multiprocessing [17], Joblib [18], mpi4py [19], and popular libraries and frameworks with the possibility of parallel computing: TensorFlow [20], Keras [21].

In this work, a parallel implementation was realized using the Joblib library. The parallelization scheme is shown in Fig. 3.

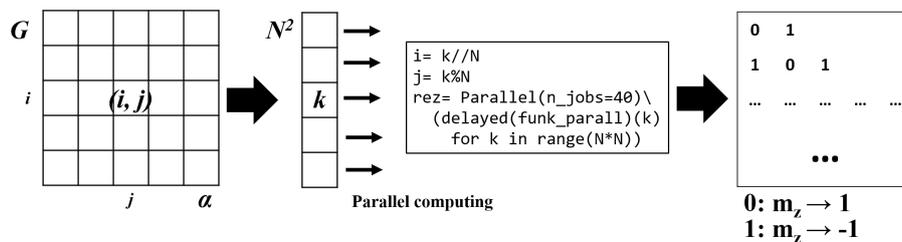


Figure 3: Task parallelization scheme using the Joblib library

In the process of parallelization, a computational area of the dimension $N \times N = 100 \times 100$ was distributed among computational threads, the value of which is specified by the n_jobs parameter. For parallelization, the *Parallel* class of the Joblib library was used (Fig. 3).

The calculations of the developed algorithm were carried out on two components of the ML/DL/HPC ecosystem, which are real computing servers. In the process of calculations, all the obtained computation times were averaged over three runs.

Figs. 4-5 demonstrate graphs of the dependence of the computation time and speedup on the number of threads on the jhub2.jinr.ru component. The graphs show values of up to 50 threads, due to the fact that with their further increase, the computation time does not change significantly. It can be seen from the graphs that in the process of calculations, a speedup of 21 times was achieved on 42 threads.

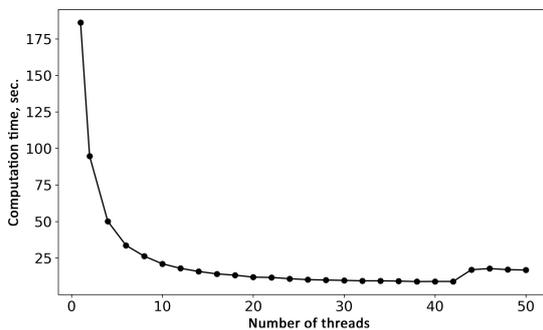


Figure 4: Graph of the dependence of the task computation time on the number of threads on jhub2.jinr.ru.

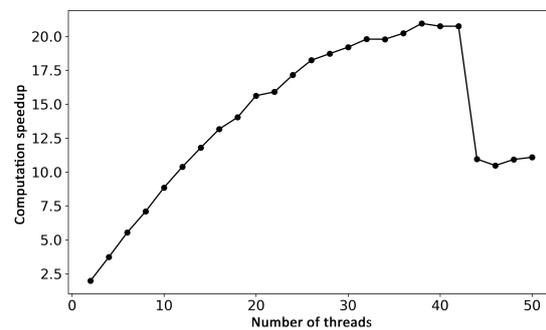


Figure 5: Graph of the dependence of the task computation speedup on the number of threads on jhub2.jinr.ru.

Figs. 6-7 illustrate graphs of the dependence of the computation time and speedup on the number of threads on the studhub.jinr.ru component. The graphs show values up to 60 threads, due to the fact that with their further increase, the computation time does not change significantly. It can be seen from the graphs that in the process of calculations, a speedup of 22 times was achieved on 50 threads.

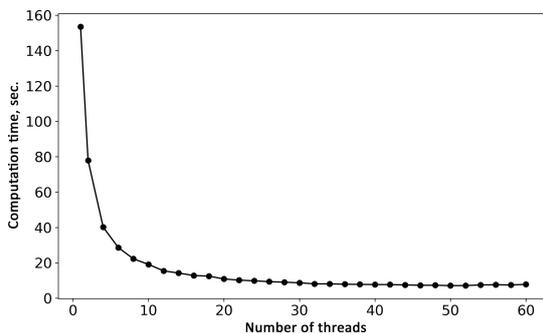


Figure 6: Graph of the dependence of the task computation time on the number of threads on studhub.jinr.ru.

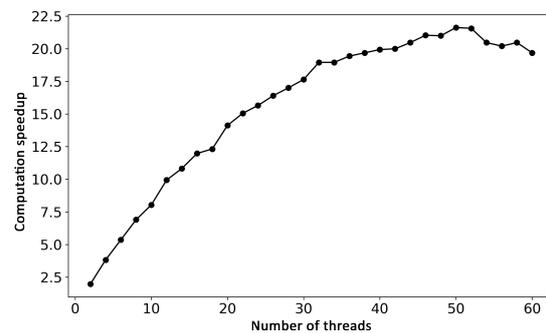


Figure 7: Graph of the dependence of the task computation speedup on the number of threads on studhub.jinr.ru.

6. Conclusion

The use of the Python language and the Joblib library to parallelize calculations simplified the process of writing a program to solve the task of studying the possibility of reversing the magnetic moment in superconductor-ferromagnet-superconductor structures. Good indicators of computation speedup by more than 20 times were obtained on both components of the ML/DL/HPC ecosystem. The finished Jupyter notebook is freely available on the resources of the HybriLIT heterogeneous platform at <http://hlit.jinr.ru/spintronics>. It will be used in the process of the laboratory and research work of both students and researchers of the institute.

Acknowledgement

The work was supported by the Russian Science Foundation, project no. 22-71-10022.

References

- [1] Virtual research environment for modeling physical processes in hybrid nanostructures comprising superconductors and magnets [Electronic resource]. URL: <http://sconduct.jinr.ru> (accessed 01.08.2022)
- [2] HybriLIT heterogeneous platform [Electronic resource]. URL: <http://hlit.jinr.ru/en/> (accessed 01.08.2022)
- [3] JINR Cloud Infrastructure [Electronic resource]. URL: <https://micc.jinr.ru/?id=19> (accessed 01.08.2022)
- [4] JINR cloud service for scientific and engineering computations [Electronic resource]. URL: <https://saas.jinr.ru> (accessed 01.08.2022)
- [5] Q. Kong, T. Siau, A. Bayen, *Python Programming and Numerical Methods: A Guide for Engineers and Scientists* [Electronic resource]. URL: <https://pythonnumericalmethods.berkeley.edu> (accessed 01.08.2022)
- [6] Project Jupyter [Electronic resource]. URL: <https://jupyter.org> (accessed 01.08.2022)
- [7] P. Kh. Atanasova, S. A. Panayotova, I. R. Rahmonov et al., *Periodicity in the Appearance of Intervals of the Reversal of the Magnetic Moment of a ϕ_0 Josephson Junction*, *Jetp Letters* **110** 722–726 2019.
- [8] JINR MICC [Electronic resource]. URL: <https://micc.jinr.ru/?id=29> (accessed 01.08.2022)
- [9] "Govorun" supercomputer [Electronic resource]. URL: http://hlit.jinr.ru/en/supercomputer_govorun (accessed 01.08.2022)
- [10] Education and testing polygon [Electronic resource]. URL: http://hlit.jinr.ru/en/education_and_testing_polygon_eng (accessed 01.08.2022)

- [11] ML/DL/HCP Ecosystem [Electronic resource]. URL: http://hlit.jinr.ru/en/access-to-resources_eng/ecosystem-for-ml_dl_bigdataanalysis-tasks_eng (accessed 01.08.2022)
- [12] Anaconda (Python distribution) [Electronic resource]. URL: <https://www.anaconda.com> (accessed 01.08.2022)
- [13] Markdown [Electronic resource]. URL: <https://daringfireball.net/projects/markdown> (accessed 01.08.2022)
- [14] Jupyter Book [Electronic resource]. URL: <https://jupyterbook.org> (accessed 01.08.2022)
- [15] MATLAB in the JupyterHub Environment [Electronic resource]. URL: <http://hlit.jinr.ru/matlab-in-jupyterhub-ru/> (accessed 01.08.2022)
- [16] SciPy. Integration and ODEs [Electronic resource]. URL: https://docs.scipy.org/doc/scipy/reference/genera-ted/scipy.integrate.solve_ivp.html (accessed 01.08.2022)
- [17] Multiprocessing. Process-based parallelism [Electronic resource]. URL: <https://docs.python.org/3/library/multiprocessing.html> (accessed 01.08.2022)
- [18] Joblib. Lightweight pipelining in Python [Electronic resource]. URL: <https://joblib.readthedocs.io> (accessed 01.08.2022)
- [19] MPI for Python [Electronic resource]. URL: <https://mpi4py.readthedocs.io/en/stable> (accessed 01.08.2022)
- [20] TensorFlow: Use a GPU [Electronic resource]. URL: <https://www.tensorflow.org/guide/gpu> (accessed 01.08.2022)
- [21] Keras: the Python deep learning API [Electronic resource]. URL: <https://keras.io> (accessed 01.08.2022)