Comparative Analysis of the Efficiency of Parallel Computing on CPU and GPU for Calculating the Physical Characteristics of Superconducting Quantum Interference Devices

A.R. Rahmonova
O.I. Streltsova, M.I. Zuev, I.R. Rahmonov

The work was performed with the support of the Russian Science Foundation within the framework of project No. 22-71-10022

11th International Conference "Distributed Computing and Grid Technologies in Science and Education"

7-11 July, Dubna

Main Tasks

Within the framework of the joint project of MLIT and BLTP an ecosystem is being developed for the tasks of studying a system based on Josephson junctions using Python in the form of Jupyter Book.

The following tasks were solved:

1. Development of algorithm for calculation of current-voltage characteristics of Superconducting Quantum Interference Device (SQUID) with two Josephson junctions and its critical current dependence on external magnetic field.
2. Realization of parallel calculation of magnetic field dependence of critical current.
3. Comparative analysis of calculations in CPU and GPU

Ecosystem ML/DL/HPC of HybriLIT platform

Component for HPC and data analysis

VM with JupyterHub and SLURM [https://jlabhpc.jinr.ru]

- Intel Xeon Gold 6126 (24 Cores @ 2.6 GHz)
- 32 GB RAM

Development component

JupyterLab Server [https://studhub.jinr.ru]
[https://studhub2.jinr.ru]

- 2x Intel Xeon Gold 6152 (22 Cores @ 2.1 GHz)
- 512 GB RAM

Component for carrying out resource-intensive calculations

Server with NVIDIA Volta [https://jhub1.jinr.ru]
[https://jhub2.jinr.ru]

- 2x Intel Xeon Gold 6148 (20 Cores @ 2.4 GHz)
- 4x NVIDIA Tesla V100 SXM2 32 GB HBM2
- 512 GB RAM

Python Libraries for Accelerating Computing

Numba

```
# import library Numba
import numba as nb
# import library numbalsoda
from numbalsoda import lsoda_sig, lsoda,
dop853
from numba import njit, cfunc
```

- The Numba library allows you to study JIT compilation (Just-In-Time compilation, compilation "at the right time"). JIT compilation translates Python and NumPy code into fast machine code during program execution.
- The @njit decorator is a tracking function for optimizations by the Numba JIT compiler.
- The @njit(parallel=True) decorator. Adding the Parallel=True parameter specifies the connection of the electrode, parallelization, and other optimizations for calculations on the CPU.

Numba + CUDA

The combination of Numba and CUDA enables parallel execution of operations on GPUs.

Some of the capabilities of Numba and CUDA:

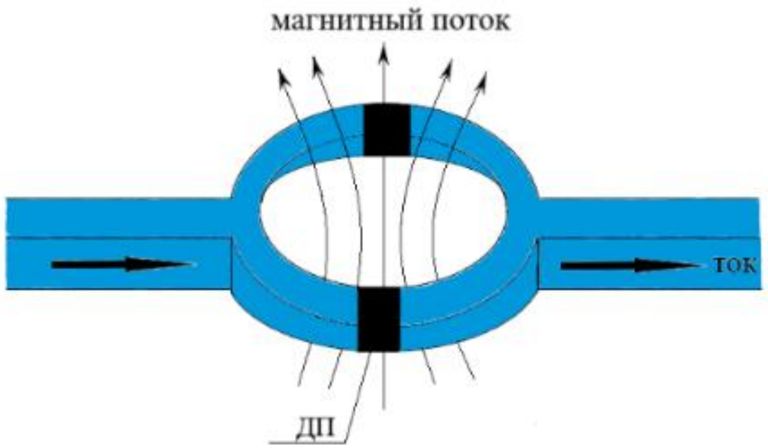
Compile Python functions to run on NVIDIA GPUs using CUDA.

Write GPU kernels using Python syntax that is similar to CUDA C.

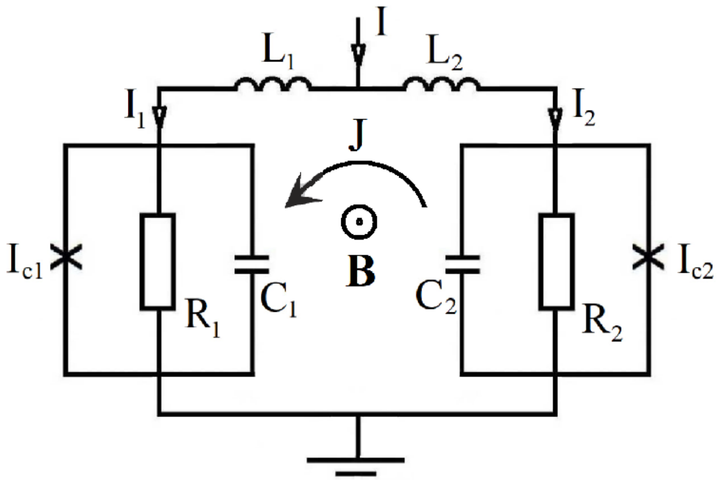
Automatically move NumPy arrays between CPU and GPU.

Control CUDA data transfers and threads, and more.

Modeling of DC-SQUID



Schematic view of DC-SQUID



Equivalent scheme of DC-SQUID

Superconducting Quantum Interference Device (DC-SQUID) is consists of superconducting ring with two Josephson junctions. DC-SQUID can be used as a device for measuring of weak magnetic fields.

Equations system for description of DC-SQUID

$$\begin{cases} \dfrac{dV_1}{dt} = \dfrac{1}{\beta_c}\left\{\dfrac{I}{2} - \dfrac{d\varphi_1}{dt} - \sin(\varphi_1) + \dfrac{1}{2\beta_L}[\varphi_2 - \varphi_1 + 2\pi(n - \varphi_{ext})]\right\} \\ \dfrac{d\varphi_1}{dt} = V_1 \\ \dfrac{dV_2}{dt} = \dfrac{1}{\beta_c}\left\{\dfrac{I}{2} - \dfrac{d\varphi_2}{dt} - \sin(\varphi_2) - \dfrac{1}{2\beta_L}[\varphi_2 - \varphi_1 + 2\pi(n - \varphi_{ext})]\right\} \\ \dfrac{d\varphi_2}{dt} = V_2 \end{cases}$$

Unknown functions

$\varphi 1$ - phase difference of first JJ

$\varphi 2$ - phase difference of second JJ

V_1 - Voltage in first JJ

V_2 - Voltage in second JJ

Model parameters

I - external current

β_c - McCumber parameter

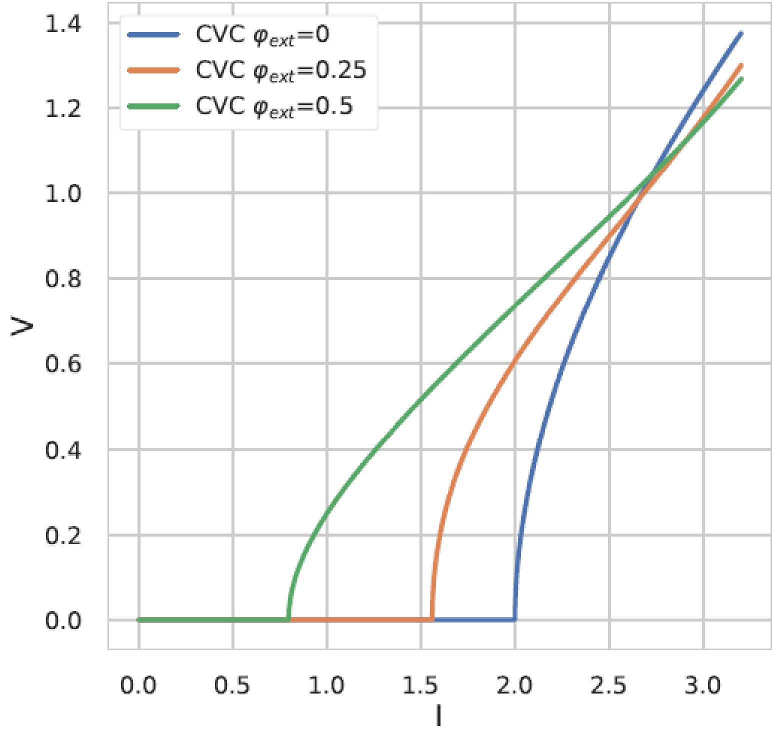
β_L - normalized inductance

φ_{ext} - normalized external magnetic flux

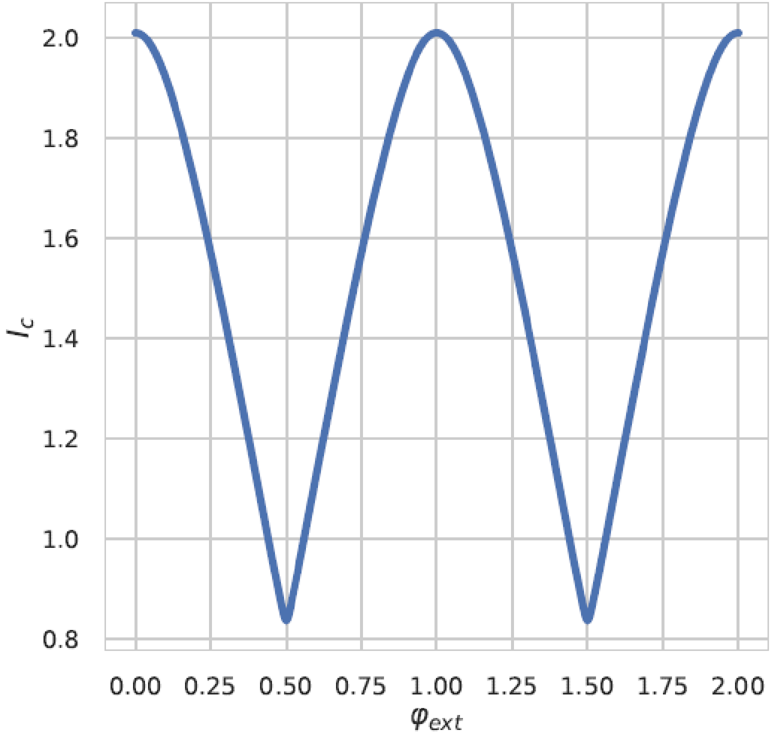
n - number of magnetization quantization

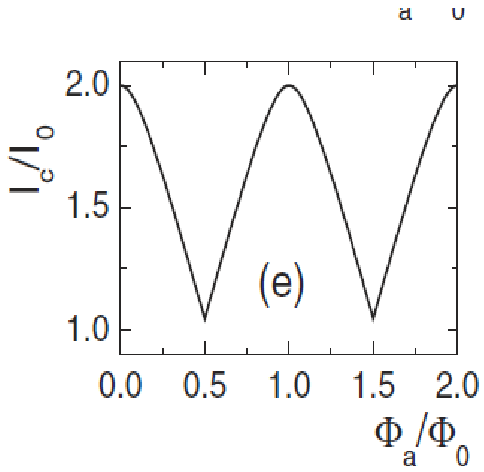
Initial conditions $\varphi 1$ =0, $\varphi 2$ =0, V_1 =0, V_2 =0

Dependence of SQUID critical current on the external magnetic field flux



Calculated volt-ampere characteristics for external magnetic field flux values: 0, 0.25 and 0.5

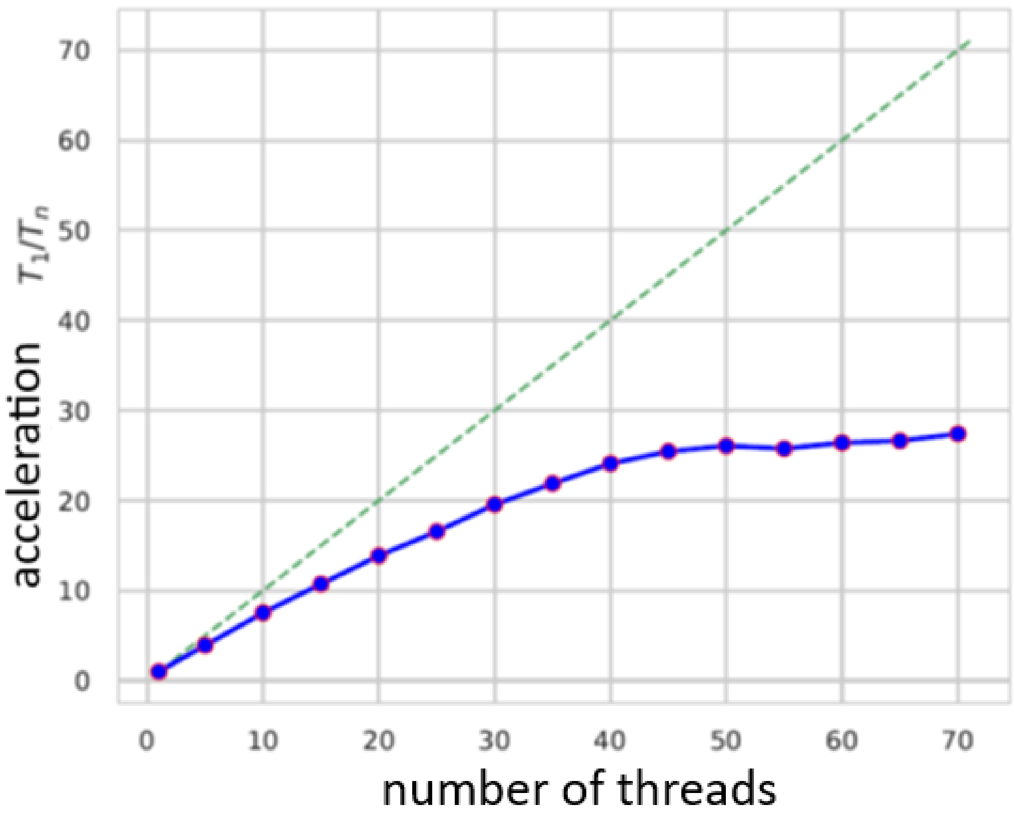


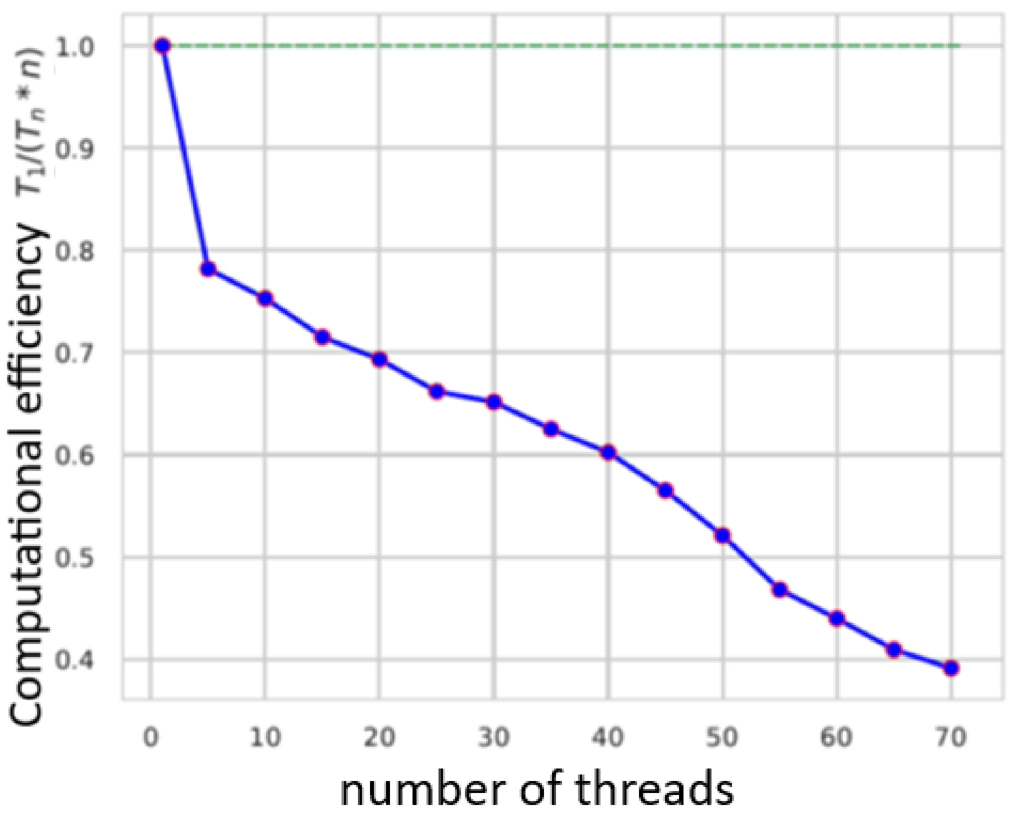


Dependence of the critical current on the magnitude of the external magnetic field flux

J. Clarke, A.I. Braginski, The SQUID Handbook: Applications of SQUIDs and SQUID Systems, Wiley-VCH Verlag GmbH \& Co. KGaA, 2006.

Results of acceleration and efficiency of calculations Using NUMBA in CPU





In case of parallel calculations **27x speedup** was achieved

Result of acceleration with NUMBA in GPU

2048 points

	Threads per block	Calculation time	Copy time from GPU to CPU
1	32	0.930580	91.29
2	64	1.113715	89.30
3	128	0.857186	103.93
4	256	0.979928	123.23
5	512	0.920108	191.78
	Numba (40 proc)	52	

4096 points

	Threads per block	Calculation time
1	32	0.476355
2	64	0.920817
3	128	0.425945
4	256	1.188718
5	512	1.044558
	Numba (40 proc)	101

8192 points

	Threads per block	Calculation time
1	32	0.433546
2	64	0.386040
3	128	0.445090
4	256	0.971825
5	512	0.406400
	Numba (40 proc)	204

Conclusion

- A toolkit for modeling the physical properties of a superconducting quantum interferometer with two Josephson junctions has been developed.

- Algorithms for calculating the current-voltage characteristic of a SQUID under the influence of an external magnetic field and the dependence of the critical current of a SQUID on external magnetic field have been developed and realized in the Python programming language.

- The software realization of the developed algorithms is performed using the Numba library function, including the parallelization mechanism when calculating the dependence of the critical current on the external magnetic field flux.

- The comparative analysis of calculation time in CPU and GPU is performed, and advantage of GPU for big number of calculations is demonstrated.

![jupyter book]

HLIT Jupyter book

Search this book...

Welcome to HLIT Jupyter Book

Select the section you are interested in

- Основы работы с Python: инструментарий на Python для решения научных и прикладных задач
- Численное решение задачи Коши: библиотека SciPy
- Задача 1: Линеаризованное уравнение на магнитный момент
- Задача 2. Периодичность появления интервалов переворота намагниченности в \(ϕ_0\) джозефсоновском переходе под воздействием импульса тока

Next
Основы работы с Python: инструментарий на Python для решения научных и прикладных задач ›

By A.R. Rahmonova, A.S. Vorontsov, A.V. Nechaevskiy, I.R. Rahmonov, M.V. Bashashin, M.I. Zuev, O.I. Streltsova, Y.A. Butenko.

A toolkit based on Python libraries and the Jupyter ecosystem for solving of scientific and applied problems