GPU Simulation of Electron Beam Current Distribution during Pulsed Heating of a Metal Target

Alexey V Snytnikov\textsuperscript{1,2,3} and Galina G Lazareva\textsuperscript{3,4} and Alexey A Romanenko\textsuperscript{3}

\textsuperscript{1} Institute of Automation and Electrometry SB RAS
\textsuperscript{2} Novosibirsk State Technical University
\textsuperscript{3} Novosibirsk State University
\textsuperscript{4} Peoples’ Friendship University of Russia (RUDN University), Moscow, Russian Federation

E-mail: snytav@gmail.com, lazareva-gg@rudn.ru, arom@nsu.ru

Abstract. The GPU implementation by means of the OpenACC technology of the simulation of the current distribution in tungsten sample is heated by an electron beam pulse is presented. A special case of axial symmetry without taking into account electrical driving forces is considered. The temperature in the sample, calculated from the two-phase Stefan problem, is needed to solve the electrodynamic equations. The calculation time for different GPUs compared to CPU time is given.

1. Introduction

BETA \cite{1} is the experimental facility in the Budker Institute of Nuclear Physics SB RAS. BETA stands for Beam of Electrons for materials Test Applications.

A natural experiment is constantly accompanied by a numerical one \cite{2}. The model of tungsten heating is based on the solution of the equations of electrodynamics and two-phase Stefan problem for temperature.

The inclusion of gas component into the model is necessary, but in present the model contains no gas, it remains the for the future work. The present paper mentions gas where it is necessary for understanding physics.

The work is devoted to the computation of the current in the sample, which is considered as a possible source of rotation of the substance, which is observed in the experiment. The computation was carried out in order to analyze and plan of experiments to determine the influence of the Ampere forces on the dynamics of matter. It is planned to investigate the supposed influence of temperature inhomogeneity on the movement of charge and melt.

Temperature inhomogeneity in the melt leads to the appearance of a current flow. In areas with temperature changes, an electromotive force occurs. It is important that the electromotive force is different in the gas, if it is considered and in the melt. Therefore, a nonzero accelerating voltage can arise on a closed loop through the gas and melt, which generates a current along this loop.

The conductivity value is nonlinearly dependent on temperature. It is possible to give estimates in which areas of the gas and the melt there will be good conductivity and high voltage. But without the use of mathematical modeling, it is difficult to predict in which areas
the greatest current can be observed. The current interacting with the magnetic field leads to the movement of the substance as a whole.

In this paper, a special case is considered when the equations for the fields and currents are written out for a tungsten sample in a cylindrical coordinate system without taking into account the electromotive forces. This is due to the need to compare, in the future course of research, the calculated effect of thermal currents with the effect of the beam current. It is assumed that the characteristic time of change is long in comparison with the time for establishing equilibrium of the equations of electrodynamics on the scale of the problem. Further development of the model assumes the inclusion of the calculation of the electromotive force in gas and melt.

In order to determine the specific conductivity, the temperature distribution in the sample is calculated based on the solution of the Stefan problem. The position and speed of movement of the phase boundary depends on nonlinear coefficients. The condition at the boundary between a free melt and a solid consists in the continuity of temperature and discontinuity of the heat flux due to the absorption or release of a known amount of heat.

The known results cannot be used due to the specificity of the problem statement (temperature and pressure range, spatial and time scale). The novelty and complexity of solving the problem is mainly due to the need for a correct description of nonlinear boundary conditions describing the heating and evaporation of a material on its surface. The practical orientation of the work requires that the formulation of the model problem matches the experimental conditions as closely as possible. The calculation results of the final model will be used for comparison with the experimental data obtained at the BETA experimental facility at the Budker INP SB RAS.

2. Basic equations

2.1. Stefan problem

In order to evaluate the temperature on the heated surface, the temperature distribution in the sample is calculated based on the solution of the Stefan problem [2]. The position and speed of the phase boundary depends on nonlinear coefficients. Stefan problem is used to compute the temperature distribution in the tungsten plate surface:

\[
\begin{align*}
    c(T)\rho^*(T)\frac{\partial T}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} r \lambda(T) \frac{\partial T}{\partial r} + \frac{\partial}{\partial z} \lambda(T) \frac{\partial T}{\partial z} \\
    (n, \nabla T) |_{\gamma} &= \frac{W(t, r) - N(t, r)}{\lambda(T)},
\end{align*}
\]

(1)

\((n, \nabla T) = 0\) at other boundaries, and \(T = T_0\) for \(t = 0\). Here \(T\) is the temperature, \(c(T)\) is the specific heat, \(\rho^*(T)\) is the density, \(\lambda(T)\) is the thermal conductivity, \(W(t, r)\) is the power of the heat flux on the surface, \(N(t, r)\) is the power loss, \(n\) is the normal to the surface, and \(T_0\) is the initial temperature.

2.2. Equation for current distribution

\[
\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{j}, \nabla \times \mathbf{E} = 0, \nabla \cdot \mathbf{B} = 0, \nabla \cdot \mathbf{j} = 0, \mathbf{E} = \rho_c \mathbf{j}
\]

(2)

here \(\rho_c\) is the unit conductivity, \(\mathbf{B}\) - magnetic field, \(\mathbf{E} = (E_r, E_\phi, E_Z)\) - electric field, \(\mathbf{j} = (j_r, j_\phi, j_Z)\) - current.

Let us introduce the vector potential of the current \((F_r, F_\phi, F_Z)\):

\[
\mathbf{j} = \nabla \times \mathbf{F}.
\]

(3)

Then the equation system 2 should be rewritten in cylindrical coordinates. Omitting the mathematical details given in [3], the result is the equation

\[
\nabla^2 \Phi + \nabla G \nabla \Phi = 0
\]

(4)
here \( G = rF_\phi \) is the distribution of current in a circle with radius \( r \) and height \( z \).

3. Numerical method

In the two-dimensional region

\[
\{(r, z)|r \in [r_0, r_{max}], z \in [z_0, z_{max}], r_0 = z_0 = 0, r_{max} = 12\text{mm}, z_0 = 3\text{mm}\},
\]

a uniform rectangular mesh with the nodes \((i, k)\), \(i = 1 \cdots N_R, k = 1 \cdots N_Z\) is introduced. Equation 1 is implemented using the stabilizing correction scheme and the Thomas method [4]. The solution method is described in detail in the works [5], [6], [2].

The use of Successive Over-Relaxation [4] method at each timestep allows to build an economical algorithm, provided the optimal value of the relaxation parameter [7].

At present the size of the mesh is \(1200 \times 300\) nodes. This size is determined by the physics, and not increase of the mesh size is planned.

4. GPU Implementation

With the given parameters (the mesh size etc.), the SOR method makes more than 680000 iterations to converge a single time step. This part of the code takes major part of the execution time and is the main hot spots. The code was ported to GPU using the OpenACC [8] technology. So, we introduced parallel sections with \texttt{acc parallel loop} directive. To keep all the data on GPU and not moving them back and forth, we wrapped the main loop with \texttt{acc data} region. Other parts of the code like initialization of the arrays and boundary conditions were marked with OpenACC directives too.

The profiler (NVidia Nsight Systems) showed that we spend much time copying error from GPU to CPU. Taking into account that we have to complete several hundred thousands operations to converge, we decided to update error value on CPU side after each 1000 iterations. This optimization gave us 5% performance gain. In such a way, there are two synchronisation points: after 1000 iterations and after each time step.

The output data are saved after each time step. In our experiments we’ve been using SSD hard drive. The profiler showed that saving data is not a hot spot. So, there is no reason to perform saving data in an independent parallel thread.

5. Performance

The code marked with OpenACC directives was compiled with NVIDIA HPC SDK 21.5. The advantage of the selected approach is we have a single source code and it can be compiled both for GPU and multi-core CPU. One a target system we have CUDA 10.2 and 11.4 installed allong with CUDA driver 470.57, operating system – Ubuntu-20.04.

Figure 1 shows the performance of the code obtained with Nvidia Tesla VT100 GPU compared to Intel(R) Xeon(R) CPU E5-2698 v4 processor. The processor was used both to run serial code and to launch OpenMP-parallelized code. The best result we got requesting 4 CPU cores. Figure 2 compares the same code on a slower processor and a bit older CPU, namely Intel core i7 and Nvidia Titan X. Obtained results on CPU are well-agreed with [9].

The profiler shows that the performance limiter is a launch latencies for the GPU-enabled code. It’s the result of the mesh size. Selected algorithm and its implementation do not allow us to make kernels bigger (to have longer execution time) and therefore rise the GPU load. It’s possible to restructure algorithm and get more 2-3% for performance but it will result in decreasing the code clarity. Probably, the better approach is to re-think the task and select another algorithm that is more suitable for parallel architectures like multi-core CPUs and GPUs.
Figure 1. Worktime for 2 timesteps, mesh 1200 × 300 nodes.

Figure 2. Worktime for 2 timesteps, mesh 1200 × 300 nodes.

6. Conclusion
The code for computation of the current in tungsten melt under the influence of a pulsed electron beam was ported to GPU. Porting was performed by using the OpenACC directives. Time measurements have shown the advantage of using a GPU for this class of tasks. The limiting factor for further growth in GPU performance and CPU scalability is the small mesh size. On the other hand, increasing the mesh size leads to a dramatic increase in the number of iterations. Due to this reason other methods for solving the above system of differential equations should be considered.

Acknowledgments
The publication has been prepared with the support of RFBR grant 19-07-00446, code development was supported by RFBR grant 19-07-00085. The work of A.A.Romanenko (GPU implementation) and the work of A.V.Snytnikov (design of numerical methods) was supported by the state funding program FSUS-2020-0034.
References

[1] Vyacheslavov L, Arakcheev A, Burdakov A, Kandaurov I, Kasatov A, Kurkuchekov V, Mekler K, Popov V, Shoshin A, Skovorodin D, Trunev Y and Vasilyev A 2016 AIP Conference Proceedings 1771 060004

[2] Arakcheev A, Apushkinskaya D, Kandaurov I, Kasatov A, Kurkuchekov V, Lazareva G, Maksimova A, Popov V, Snytnikov A, Trunev Y, Vasilyev A and Vyacheslavov L 2018 Fusion Engineering and Design 132 13–17 ISSN 0920-3796 URL https://www.sciencedirect.com/science/article/pii/S0920379618304356

[3] Lazareva G, Popov V, Arakcheev A, Burdakov A, Schwab I, Vaskevich V, Maksimova A, Ivashin N and Oksogoeva I 2021 Journal of Applied and Industrial Mathematics (2)

[4] Demmel J W 1997 Applied Numerical Linear Algebra (SIAM)

[5] Lazareva G G, Arakcheev A S, Kandaurov I V, Kasatov A A, Kurkuchekov V V, Maksimova A G, Popov V A, Snytnikov A V, Trunev Y A, Vasilyev A A and Vyacheslavov L N 2018 AIP Conference Proceedings 2025 080005

[6] Lazareva G, Arakcheev A S, Burdakov A, Kandaurov I, Kasatov A, Kurkuchekov V, Maksimova A G, Popov V, Shoshin A, Snytnikov A, Trunev Y A, Vasilyev A and Vyacheslavov L 2018

[7] Yang S and Gobbert M K 2009 Applied Mathematics Letters 22 325–331 ISSN 0893-9659 URL https://www.sciencedirect.com/science/article/pii/S0893965908001523

[8] Openacc, more science, less programming URL https://www.openacc.org

[9] Lazareva G, Korneev V and Maksimova A 2021