High-performance version of the RPB code based on graphic processors for determination of the plasma boundary in tokamaks

I V Zotov¹,² and L I Vysotsky¹,³

¹ Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, 119991 Russia
² E-mail: iv-zotov@cs.msu.ru, ³ E-mail: vysotskylev@yandex.ru

Abstract. The operation of modern tokamaks is impossible without an effective system for controlling the plasma boundary during the discharge. In this paper, we consider a method for determining the plasma boundary on the basis of integral equations. A high-speed parallel code using GPUs was developed for this purpose. Different methods of parallelization of the algorithm have been considered. The possibility of processing the magnetic measurements in real time during experiments is shown. The effect of the technical parameters of GPUs (the number of cores, the data bus width, the amount of internal memory) on the performance is also shown. The simulation results for the T-15MD tokamak, currently under construction at National Research Center Kurchatov Institute, are presented.

1. Introduction
The operation of modern tokamaks is impossible without an effective plasma boundary control system. The problem of determining the plasma boundary on the basis of discrete magnetic measurements is the subject of a number of papers [1–8]. From the mathematical point of view, the problem is reduced to a Cauchy problem for a two-dimensional homogeneous elliptic equation for the plasma MHD equilibrium (the homogeneous Grad–Shafranov equation). Different computational approaches are applied to solving the resulting ill-conditioned problem, namely, the method of harmonic expansion in special functions (EFIT code, etc. [1–2]), the method of point filaments [3], the method of distributed filaments [4], and the integral equation method (RPB code [5–6]). Each of these methods has its advantages and disadvantages. In this paper, a method for determining the plasma boundary based on integral equations is considered. A high-speed parallel code named RPB-GU is developed. Different methods of parallelization of the algorithm are considered. The most efficient of them show that processing magnetic measurements in real time is possible. Comparison of serial and parallel implementation of the code is carried out. We also present the results of computational experiments on different GPUs for the T-15MD tokamak [9].

2. Formulation of the problem
We solve the inverse problem of magnetohydrodynamic equilibrium (the problem of finding a free boundary magnetic surface). It is set by a two-dimensional elliptic Grad-Shafranov equation with additional Cauchy-type conditions at the boundary. To find the plasma boundary Γₚₜ, the following problem is posed:

\[
\begin{cases}
\Delta^* \psi = 0, (r, z) \in \Omega_{\psi}, \\
\partial \psi / \partial n_{\text{L}} = F_1, \partial \psi / \partial n_{\text{L}} = F_2.
\end{cases}
\]

Here, \( \Delta^* \psi \equiv r \frac{\partial}{\partial r} \left( \frac{1}{r} \frac{\partial \psi}{\partial r} \right) + \frac{\partial^2 \psi}{\partial z^2} \) is the adjoint Laplace operator. The \( \Omega_{\psi} \) region is the vacuum region surrounding the plasma in the shape of a ring bounded from the outside by the measurement contour \( \text{L} \) and from the inside by the plasma boundary \( \Gamma_{\text{pt}} \). The plasma boundary is free and is determined either at the limiter from the condition of being tangential to the diaphragm, or at the stationary stage from the condition of passing through the X-point of the separatrix. Functions \( F_1 \) and \( F_2 \) are the experimentally measured normal and tangential components of the magnetic field, respectively.
3. Solution Method

Two fixed contours are selected: one inside the plasma \((l_1)\), the other \((l_2)\) outside the observation contour \(L\). The current in the plasma and the poloidal currents are approximated by surface currents in the carrier circuits \((l_1, l_2)\). The solution is sought as the sum of simple-layer potentials. Taking into account boundary conditions, we obtain a system of Fredholm integral equations of the first kind. To solve the system numerically, the contours \(l_1\) and \(l_2\) are divided into parts of the equal length, and the densities of potentials of a simple layer \(v_1\) and \(v_2\) are calculated at the nodes of this division. Thus, we obtain a system of linear algebraic equations for the variables \(v_{k,j}, k \in \{1,2\}, j \in \{1, ..., N\}\). The system is ill-conditioned (the condition number is of the order \(10^{18}–10^{27}\)) and therefore, regularization is necessary. The functional that is minimized has the form

\[
J(v_1, v_2) = \|B_t - B_t^f\|_{L_2}^2 + \|B_n - B_n^f\|_{L_2}^2 + \alpha \left(\|v_1\|_{W_1}^2 + \|v_2\|_{W_2}^2\right),
\]

where \(B_t\) is the tangential and \(B_n\) is the normal component of the poloidal magnetic field, \(L_2\) is the space of functions having a quadratic norm, \(\alpha\) is a regularization parameter, and \(W_1^2\) is the space of functions having a first derivative integrable with a square.

The finite-dimensional analogue of this functional is convex like a quadratic form with a symmetrical positively defined matrix. Therefore, the necessary and sufficient condition for a global minimum [10] is its gradient being equal to zero or, equivalently,

\[
M_{\alpha} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} h_1 A^T A + a(h_1 + h_2)I + a(h_1 + h_2) \begin{bmatrix} D_1^T & 0 \\ 0 & D_2^T \end{bmatrix} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = A^T F,
\]

Here, \(A \in \mathbb{R}^{2N \times 2N}\) is the operator of the system of integral equations, \(A^T\) is transposed matrix, \(F \in \mathbb{R}^{2M}\) is the vector of measurements, \(D_k, k \in \{1,2\}\) are the matrices of the differentiation operators in the stabilizer of the functional \(J\), \(h_1\), \(h_1\), \(h_2\) are steps of partitioning the contours \(L, l_1, l_2\), and \(I\) is unit matrix. Consequently, the solution to the system of equations is \(M_{\alpha}^{-1} A^T F\).

A quasi-optimal choice of the regularization parameter \(\alpha\) is used, \(\alpha \|d f_{opt}(\alpha)/d \alpha\|^2 \rightarrow \min_{\alpha}\). The optimal solution is sought numerically by sweeping the values of \(\alpha\) on a logarithmic grid. For an effective solution on a GPU, the \(M_{\alpha}\) matrices for a series of different values of \(\alpha\) are inverted, multiplied by \(A^T\), and concatenated into one “high” matrix \(\hat{M}\). Then, to find the unknown current densities \(v_1\) and \(v_2\), it suffices to multiply this matrix by the measurement vector \(F\).

4. Parallel algorithm

The algorithm for determining the plasma boundary consists of the following steps:

1. The matrix \(\hat{M}\) is formed and loaded onto a GPU, then multiplied by \(F\) using the CuBLAS linear algebra library for graphic processors.
2. From the obtained set of solutions \(v(\alpha_1), ..., v(\alpha_k)\), the best solution is selected by the criterion of quasi-optimality (in parallel for all \(\alpha\)).
3. In parallel, for each node of the grid, the value of the flow of the poloidal field \(\psi\) is calculated.
4. If it is necessary to visualize the plasma boundary, then the parallel “marching squares” method is launched to find the level line with the specified value.
5. It is also possible to determine the deviation of the boundary from the position specified in the scenario at individual points (gaps).

The matrix of the system of integral equations depends on the location of the carrier contours \(l_1\) and \(l_2\) and the measurement contour \(L\). The external contour \(l_2\) is fixed, and the internal contour \(l_1\) can vary depending on the discharge stage (initial, rise, or stationary). With the number of sensors \(M = 200\), the amount of memory necessary for storing 40 matrices (for different \(\alpha\)) is approximately 50 MB. For a GPU with 3–12 GB of video memory we can store about 80–250 matrices and update the “current” every 10 ms of a 2.5-s-long discharge without using the main memory.

5. Calculation results

To compare the characteristics of the graphics processors, the peak performance on operations with single (sf) and double (df) precision was used (Table 1). These values can be obtained using the CUDA runtime API calls and depend only on the characteristics of the graphics cards. The performance of the program using the GPU is estimated using the effective (achieved) performance (GFlops): \(P_{\text{eff}} = N_{\text{op}}/t\), where \(N_{\text{op}}\) is the total number of operations and \(t\) is the total running time. The calculations were carried out on different
Nvidia GPUs: from low-end (GT730M, GTX970M) to high-end ones (Tesla C2075, Tesla K20c, Tesla K40c).

Comparison of CPU and GPU implementation is presented in Table 2 for Intel Xeon E5620 and Nvidia Tesla K40c (double precision). The parallel version of the RPB code exceeds the standard version in speed 24–60 times. This confirms the high efficiency of the proposed algorithm. The dependence of the solution time on the number of sensors for different GPUs is shown in Fig. 1 for single (Fig. 1a) and double (Fig. 1b) precision.

Figure 2 shows a two-zero configuration of the future T-15MD tokamak at the time of discharge $t = 2.5 \, \text{s}$ [7, 8]. In this configuration, the dotted line shows the accuracy of the recovery of the boundary when the relative error of the signals on the sensors is of 1%. In this case, the accuracy of determining the boundary as a whole is of approximately 0.5 cm.

### Table 1. Parameters of graphic devices.

| GPU Youtube | GT 730M | GTX 970M | Tesla C2075 | Tesla K20c | Tesla K40c |
|-------------|---------|----------|-------------|------------|------------|
| Peak (Gflops),single | 557 | 2365 | 1030 | 3524 | 4291 |
| Peak (Gflops),double | 139 | 73.9 | 515 | 1175 | 1430 |
| Effective (Gflops),single | 23.2 | 79.8 | 84.1 | 114 | 123 |
| Effective (Gflops),double | 9.0 | 33.2 | 63.7 | 92.8 | 103 |
| Total Cuda Cores | 384 | 1280 | 448 | 2496 | 2880 |
| GPU Clock Rate(Ghz) | 0.725 | 0.924 | 1.15 | 0.702 | 0.745 |
| Total Cuda DPU | 192 | 640 | 172 | 1248 | 1440 |

### Table 2. CPU and GPU solution time (ms) vs. the number of magnetic sensors (df).

| N | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 | 200 |
|---|----|----|----|-----|-----|-----|-----|-----|-----|
| CPU time, ms | 26.98 | 36.91 | 46.95 | 57.27 | 67.69 | 78.35 | 89.17 | 90.41 | 112.01 |
| Tesla K40c time, ms | 1.12 | 1.13 | 1.24 | 1.30 | 1.35 | 1.54 | 1.58 | 1.72 | 1.8 |

Fig. 1. (a, b) solution time (ms) vs. number of magnetic sensors: (a) sf, (b) df.

Fig. 2. Reconstructed and exact plasma boundaries in T-15MD tokamak (solid blue and dotted red lines, respectively) for 36 two-component magnetic sensors ($B_t$ and $B_n$) (blue circles) at a measurement error of 1%. 
6. Conclusions
The calculations show that the effective parallel implementation of the RPB code on GPUs allows one to process magnetic measurements in real time, even when the regularization parameter is chosen by the criterion of quasi-optimality at each moment of time.

Acknowledgments
This work was supported by RFBR grants nos. 17-07-544-a and 17-07-883-a.

References
[1] Ferron J R, Walker M L, Lao L L, Jonh H E S T, Humphreys D A, Leuer J A 1998 Nuclear Fusion 38 1055
[2] Faugeras B 2016 Fusion Science and Technology 69 (2) 495
[3] Zotov I V, Persiyanov I S, Sychugov D Yu 2004 Problems of Atomic Science and Technology. Ser. Thermonuclear Fusion 27 (4) 44
[4] Tsaun S, Jhang H 2007 Fusion Engineering and Design 82 163
[5] Belov A G, Zotov I V, Sychugov D Yu 2012 SCET2012 Spring World Congress on Engineering and Technology (Xi’an, China, 2012) 278 (http://www.scirp.org).
[6] Zotov I V, Belov A G 2014 Problems of Atomic Science and Technology. Ser. Thermonuclear Fusion 37 (1) 97
[7] Zotov I V, Melnikov A V, Sychugov D Yu, Lukash V E, Khayrutdinov R R 2016 43rd EPS Conf. on Plasma Phys. and Contr. Fias. (Leuven, Belgium, 2016) P2.035
[8] Zotov I V, Belov A G, Sychugov D Yu, Lukash V E, Khayrutdinov R R 2015 Problems of Atomic Science and Technology. Ser. Thermonuclear Fusion 38 (2) 51
[9] Melnikov A V et al 2015 Fusion Engineering and Design 96–97 306
[10] Fang K-T, Zhang Y-T 2016 Generalized multivariate analysis (Beijing: Science Press, Berlin: Springer-Verlag)