# Parallel modular algorithm for computing an adjoint matrix in the ring of polynomials of several variables[1]

Andrey Betin

**Abstract.** There is produced a parallel algorithm for computing an adjoint matrix in the ring of polynomials of several variables . The algorithm is based on the method of homomorphic images in the ring of integers and in the ring of polynomials.

Let $n$ be the number of processors in cluster. $0,\ 1,\ 2, \ldots,\ (n-1)$ is the number of processor. Zero processor has matrix $A$ with elements in the ring of polynomials of several variables.

The parallel algorithm consists of four steps.

**Step 1. Computation the number of modules**

**The zero processor sends the matrix $A$ to all other processors. Each processor calculates a quantity of polynomial modules and a quantity of numerical modules. As a result each processor has the input matrix $A$, the quantity of polynomial modules and the quantity numerical modules. We denote $r$ quantity of all modules.**

**Step 2. Solution of the problem over finite fields**

**We denote by $m$ any remainder of $r$ by $n$. Each processor with number $0,\ 1,\ 2, \ldots,\ m-1$ calculates the adjoint matrix in the finite field in $\left\lfloor \frac{r}{n} \right\rfloor + 1$ points, each other processor calculates the adjoint matrix in $\left\lfloor \frac{r}{n} \right\rfloor$ points.**

**Then $i$-th processor computes quantity $r_i$ of tasks over finite fields where $r_i$ is equal $\left\lfloor \frac{q}{k} \right\rfloor + 1$ for $i = 0, \ldots, j-1$ or $r_i$ is equal $\left\lfloor \frac{q}{k} \right\rfloor$ for $i = j, \ldots, k-1$. Consequently, the task is distributed to all processors proportionally. As a result each processor has an array of matrices with elements over a finite field.**

**We use an algorithm for computing adjoint matrix over a finite field from work [1, 2].**

[1] This work was partially supported by the Russian Foundation for Basic Research (grant No. 12-01-06020).

**Step 3. Recovery elements of adjoint matrixes into initial field**

**Each processor splits each matrix from the obtained array into $n$ parts with equal count of elements. The processors share the parts. So, on all processors is provided a uniform load at this stage. Each processor recovers his own part of elements of the adjoint matrix. Recovering of elements of the adjoint matrix is carried out using solutions of Lagrange and solutions of Newton of the Chinese remainder theorem [3].**

**Step 4. Result gathering at the root processor**

**Each processors sends a part of recovered elements to the zero processor. The zero processor obtains adjoint matrix in the initial field as a result.**

**For matrix which size more then $256 \times 256$ elements we can use parallel algorithm for computing adjoint matrix over a finite field [4, 5]. Now algorithm is testing on the supercomputer MVS100k of Joint Supercomputer Center of the RAS. Result of experiments will be reported on the conference.**

## References

[1] Malaschonok G.I. Matrix methods for computing in commutative rings. Monograph. 2002. P.78-82 (Russian) Pereslavtseva O.N. Computing of characteristic polynomials of polynomial matrix. Tambov University Reports. Series: Natural and Technical Sciences. V. 14. Issue 1, 2009. P. 274-277. (Russian)

[2] Malaschonok G.I. About computing of a kernel of an operator. Tambov University Reports. Series: Natural and Technical Sciences. V. 13. Issue 1, 2008. P. 129-131. (Russian)

[3] Buchberger, B., J. Kalma, E. Kaltofen, and other. Computer Algebra: Symbolic and algebraic computations. 1986.

[4] Betin A.A. Parallel recursive algorithm for computing the adjoint matrix. Tambov University Reports. Series: Natural and Technical Sciences. V. 14. Issue 1, 2009. P. 265-269. (Russian)

[5] Betin A.A. Experiments with parallel recursive algorithm for computing the adjoint matrix. Tambov University Reports. Series: Natural and Technical Sciences. V. 15. Issue 1, 2012. P. 341-345. (Russian)

**Andrey Betin**
**e-mail:** andrey.betin@gmail.com