## Parallel Algorithm for Computing the PRS

Oxana Pereslavtseva[1]

*Tambov State University, Internatsionalnaya, 33, Tambov, Russia*
e-mail: oxana.pereslavtsevagmail.com

In this talk we present a parallel algorithm for computing a polynomial remainder sequence in ring $Z[x_1, \dots, x_t]$. The algorithm is based on a homomorphic images method. The algorithm consists of four steps.

Step 1. Computation the number of modules.

Step 2. Solution of the problem over finite fields. We use an algorithm has been proposed in [1].

Step 3. Recovery coefficients of polynomials over ring $Z[x_1, \dots, x_t]$.

Step 4. Result gathering at the root processor.

We discuss the computation time and experiments with programms in computer algebra system MathPartner [2].

## Acknowledgments

Author is partially supported by RFBR grant No 16-07-00420.

## References

[1] A.G. Akritas, G.I. Malaschonok, P.S. Vigklas. *Sturm Sequences and Modfied Subresultant Polynomial Remainder Sequences.* Serdica Journal of Computing, V.8, No 1, 2014, 101-118.

[2] G. I. Malaschonok. *MathPartner Computer Algebra* ISSN 0361-7688, Programming and Computer Software, 2017, Vol. 43, No. 2, pp. 112-118.



## Параллельный алгоритм обращения целочисленной матрицы: эксперименты на кластере МВС-10П

Сергей Хворов

*Тамбовский государственный университет им. Г.Р. Державина, Интернациональная, 33, RU-392000, Тамбов, Россия*
e-mail: derbist27@gmail.com

### Аннотация

Данная статья посвящена параллельному алгоритму нахождения обратной матрицы через присоединенную матрицу и определитель, его программной реализации, а также экпериментам, которые были произведены на кластере МВС-10П. Параллельный алгоритм основан на использовании Китайской теоремы об остатках и последовательном алгоритме, реализованном в системе компьютерной алгебры MATH PARTNER. Граф алгоритма имеет двухуровневую структуру, достигнуто равномерное распределение между процессорами.

Алгоритм вычисления обратной матрицы сводится к вычислению опеределителя $\det A$ матрицы $A$ и матрицы алгебраических дополнений $A^* = (a^*_{ij}) = ((-1)^{i+j} M_{ij}(A))$. При этом, если полученный определитель не равен нулю, то обратная матрица вычисляется по формуле

$$A^{-1} = (A^*)^T / \det A. \qquad (6)$$

Кроме того, в алгоритме используется китайская теорема об остатках. Она позволяет найти решение задачи в простых полях, а затем получbnm ответ в рациональных числах.

При этом, сложность вычислений в $n$ раз уменьшается и становятся доступны параллельные потоки вычислений: в каждом простом поле вычисления происходят независимо. Общее количество арифметических машинных операций в таком алгоритме будет $\sim (mn^{1+\log_2 7} + n^2m^2)$. Если число процессоров порядка $n(m/M)$, где М - разрядность машинного слова, то общее время вычислений на кластере будет оцениваться величиной $\sim n^{\log_2 7}$.

**Теорема** (китайская теорема об остатках): Пусть числа $m_1, m_2, ..., m_n$ – попарно взаимно простые, $r_1, r_2, ..., r_n$ – целые числа и $M = m_1 m_2 ... m_n$.