Sparse matrices in computer algebra when using distributed memory: theory and applications

G. Malaschonok, E. Ilchenko [1]

Jerusalem, ACA-2017, 17-21 July, 2017

[1] National University Kiev-Mogila Academy, Kiev, Ukraine and Tambov State University, Tambov, Russia

Outline of the talk

History

Outline of the talk

History

Applications

Outline of the talk

History

Applications

Algorithms

Outline of the talk

History

Applications

Algorithms

Experiments

History: block-recursive matrix algorithms

The generalization of Strassen's matrix inversion algorithm (1969) with additional permutations of rows and columns by J. Bunch and J. Hopkroft (1974) is not a block-recursive algorithm.

History: block-recursive matrix algorithms

The generalization of Strassen's matrix inversion algorithm (1969) with additional permutations of rows and columns by J. Bunch and J. Hopkroft (1974) is not a block-recursive algorithm.

Block-recursive algorithms were not so important as long as the calculations were performed on computers with shared memory...

History: block-recursive matrix algorithms

The generalization of Strassen's matrix inversion algorithm (1969) with additional permutations of rows and columns by J. Bunch and J. Hopkroft (1974) is not a block-recursive algorithm.

Block-recursive algorithms were not so important as long as the calculations were performed on computers with shared memory...

... but very impotant for sparse super large matrices on a supercomputer with distributed memory.

Generalisation of Strassen's inversion and algorithm for the solution of systems in commutative domains was discraibed at 1997-2006 ([7], [8], [10], [9]) with strong restriction: the leading minors should not be zero.

History: block-recursive matrix algorithms

Algorithms for solution of a system of linear equations of size n in an integral domain, which served as the basis for creating recursive algorithms:

(1983) Forward and backward algorithm ($\sim n^3$) [4].
(1989) One pass algorithm ($\sim \frac{2}{3}n^3$) [5].
(1995) Combined algoritm with upper left block of size r ($\sim \frac{7}{12}n^3$ for $r = \frac{n}{2}$) [6].

History: block-recursive matrix algorithms

Recursive algorithms for solution of a system of linear equations and for adjoint matrix computation in an integral domain without permutations:

(1997) Recursive algorithm for solution of a system of linear equations [7].
(2000) Adjoint matrix computation (with 6 levels) [8].
(2006) Adjoint matrix computation alternative algorithm (with 5 levels) [10].

History: block-recursive matrix algorithms

This restriction was removed at (2008- 2015):

Main recursive algorithms for sparse matrices:

(2008) The algorithm that computes the adjoint matrix, the echelon form, and the kernel of the matrix operator for the commutative domains was described in [11] .

(2010) The block-recursive algorithm for the Bruhat decomposition and the LEU decomposition for the matrix over the field was obtained in [12] ,

(2013, 2015) and these algorithms were generaized to the LDU and Bruhat decomposition for the matrices over commutative domains in [14], [15] .

New achivements:

It is proved that the LEU algorithm has the complexity $O(n^2 r^{\beta-2})$ for matrices of rank r. [19] (2013).

It is proved that the LEU algorithm has the complexity $O(n^2 s^{\beta-2})$ for quasiseparable matrix, if any it's submatrix which entirely below or above the main diagonal has small rank s [20] (2017) .

1) Applications: Calculation of electronic circuits.

The behavior of electronic circuits can be described by Kirchhoff's laws. The three basic approaches in this theory are direct current, constant frequency current and a current that varies with time. All these cases require the compilation and solution of sparse systems of equations (numerical, polynomial or differential). The solution of such differential equations by the Laplace method also leads to the solution of polynomial systems of equations [16].

2) Applications: Control systems.

In 1967 Howard H. Rosenbrock introduced a useful state-space representation and transfer function matrix form for control systems, which is known as the Rosenbrock System Matrix [17]. Since that time, the properties of the matrix of polynomials being intensively studied in the literature of linear control systems.

3). Applications: Computation of Groebner basis.

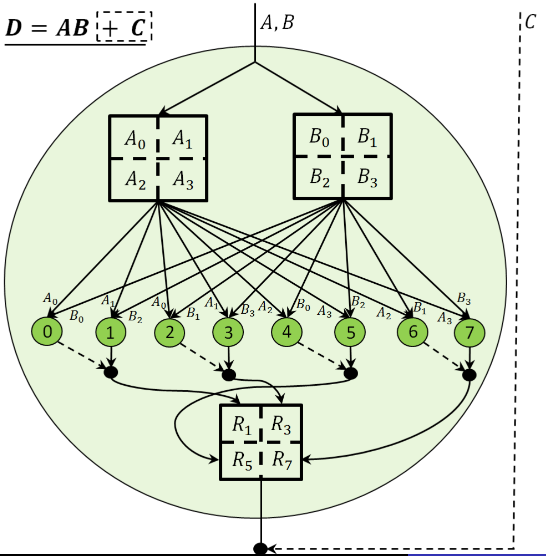
A matrix composed of Buchberger S-polynomials is a strongly sparse matrix. Reduction of the polynomial system is performed when calculating the echelon and diagonal forms of this matrix. The algorithm F4 [18] was the first such matrix algorithm.

4) Applications: Solving ODE's and PDEs.
Solving ODE's and PDE's is often based on solution of leanear systems with sparse matrices over numbers or over polynomials. One of the important class of sparse matrix is called quasiseparable. Any submatrix of quasiseparable matrix entirely below or above the main diagonal has small rank. These quasiseparable matrices arise naturally in solving PDEs for particle interaction with the Fast Multi-pole Method (FMM). The efficiency of application of the block-recursive algorithm of the Bruhat decomposition to the quasiseparable matrices is studied in [20].

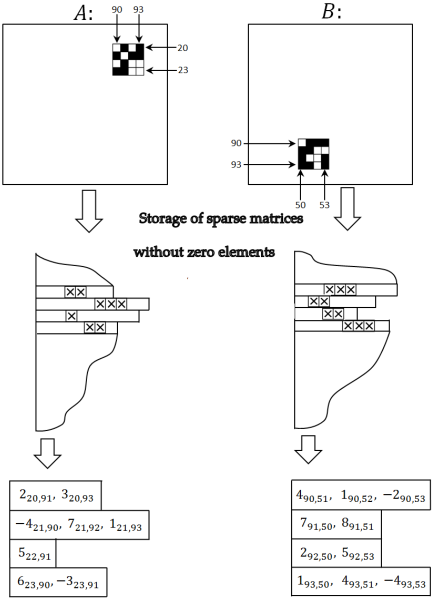
Recursive matrix multiplication for tree trunk and branches

$$\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} \times \begin{pmatrix} B_0 & B_1 \\ B_2 & B_3 \end{pmatrix} + \begin{pmatrix} C_0 & C_1 \\ C_2 & C_3 \end{pmatrix} = \begin{pmatrix} D_0 & D_1 \\ D_2 & D_3 \end{pmatrix}$$

$\boldsymbol{D = AB + C}$



Recursive sparse matrix multiplication on the leaf's block



Recursive Strassen matrix inversion

If $\mathcal{A} = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix}$, $\det(\mathcal{A}) \neq 0$ and $\det(A_0) \neq 0$ then

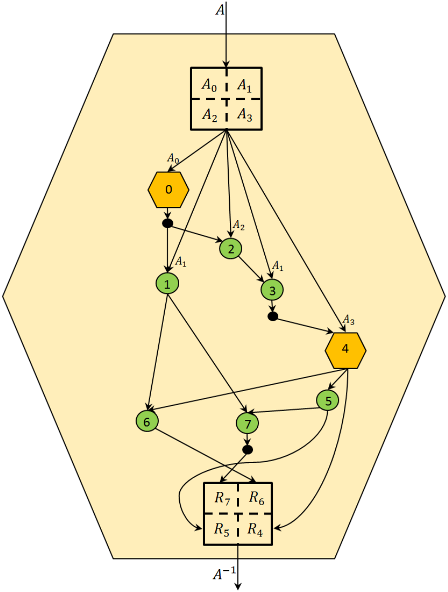
$$\mathcal{A}^{-1} = \begin{pmatrix} \mathbf{I} & -A_0^{-1}A_1 \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & 0 \\ 0 & (A_3 - A_2A_0^{-1}A_1)^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & 0 \\ -A_2 & \mathbf{I} \end{pmatrix} \begin{pmatrix} A_0^{-1} & 0 \\ 0 & \mathbf{I} \end{pmatrix}$$

$$= \begin{pmatrix} A_0^{-1} + A_0^{-1}A_1(A_3 - A_2A_0^{-1}A_1)^{-1}A_2A_0^{-1} & -A_0^{-1}A_1(A_3 - A_2A_0^{-1}A_1)^{-1} \\ -(A_3 - A_2A_0^{-1}A_1)^{-1}A_2A_0^{-1} & (A_3 - A_2A_0^{-1}A_1)^{-1} \end{pmatrix}.$$

If $M_0 = -A_0^{-1}$, $M_1 = M_0A_1$, $M_2 = A_2M_0$, $M_3 = M_2A_1$, $M_4 = (A_3 + M_3)^{-1}$, $M_5 = -M_4M_2$, then

$$\mathcal{A}^{-1} = \begin{pmatrix} M_1M_5 - M_0 & M_1M_4 \\ M_5 & M_4 \end{pmatrix}.$$

Recursive matrix inversion Strassen



Recursive computation of the adjoint and kernel: 1 of 2

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

$$A_{ext}(M_{11}, d_0) = (A_{11}, S_{11}, E_{11}, d_{11}).$$

$$M_{12}^1 = \frac{A_{11} M_{12}}{d_0}, M_{21}^1 = -\frac{M_{21} Y_{11}}{d_0}, M_{22}^1 = \frac{M_{22} d_{11} - M_{21} E_{11}^T M_{12}^1}{d_0}.$$

$$A_{ext}(\bar{I}_{11} M_{12}^1, d_{11}) = (A_{12}, S_{12}, E_{12}, d_{12}), A_{ext}(M_{21}^1, d_{11}) = (A_{21}, S_{21}, E_{21}, d_{21}).$$

$$M_{22}^2 = -\frac{A_{21} M_{22}^1 Y_{12}}{(d_{11})^2}, \ d_s = \frac{d_{21} d_{12}}{d_{11}}.$$

$$A_{ext}(\bar{I}_{21} M_{22}^2, d_s) = (A_{22}, S_{22}, E_{22}, d_{22}).$$

$$M_{11}^2 = -\frac{S_{11} Y_{21}}{d_{11}}, M_{12}^2 = \frac{\left(\frac{\frac{S_{11} E_{21}^T A_{21}}{d_{11}} M_{22}^1 - I_{11} M_{12}^1 d_{21}}{d_{11}} \right) Y_{12} + S_{12} d_{21}}{d_{11}}, M_{12}^3 = -\frac{M_{12}^2 Y_{22}}{d_s},$$

$$M_{22}^3 = S_{22} - \frac{I_{21} M_{22}^2 Y_{22}}{d_s},\ A^1 = A_{12} A_{11},\ A^2 = A_{22} A_{21},$$

$$L = \left(\frac{A^1 - \frac{I_{11} M_{12}^1 E_{12}^T A^1}{d_{11}}}{d_{11}} \right) d_{22}, \quad P = \frac{A^2 - \frac{I_{21} M_{22}^2 E_{22}^T A^2}{d_s}}{d_{21}},$$

$$F = -\frac{\left(\frac{S_{11} E_{21}^T A_{21}}{d_{11}} \right) d_{22} + \frac{M_{12}^2 E_{22}^T A^2}{d_s}}{d_{21}}, \quad G = -\frac{\left(\frac{M_{21} E_{11}^T A_{11}}{d_0} \right) d_{12} + \frac{M_{22}^1 E_{12}^T A^1}{d_{11}}}{d_{11}},$$

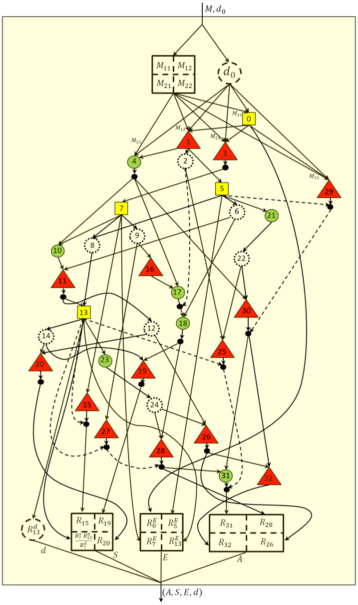
$$A = \begin{pmatrix} \frac{L+FG}{d_{12}} & F \\ \frac{PG}{d_{12}} & P \end{pmatrix}, S = \begin{pmatrix} \frac{M_{11}^2 d_{22}}{d_{21}} & M_{12}^3 \\ \frac{S_{21} d_{22}}{d_{21}} & M_{22}^3 \end{pmatrix}, E = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix},\ d = d_{22}.$$

Then

$$A_{ext}(M, d_0) = (A, S, E, d_{22}).$$

$$I_{ij} = E_{ij} E_{ij}^T, \bar{I}_{ij} = \mathbf{I} - I_{ij}, Y_{ij} = E_{ij}^T S_{ij} - d_{ij} \mathbf{I}, \quad i, j \in 1, 2.$$

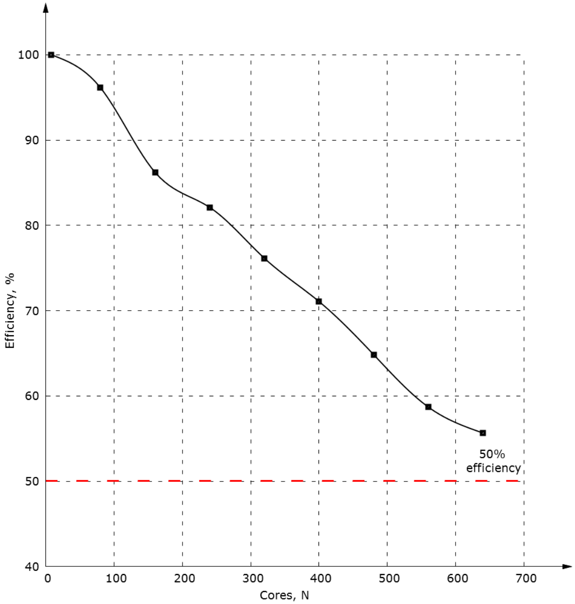
Recursive adjoint matrix computation



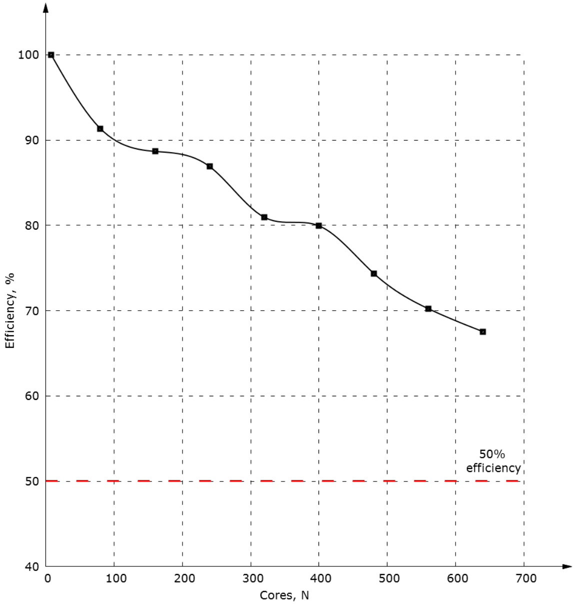
Sparse matrices when using distributed memory

The block-recursive matrix algorithms for sparse matrix require a special approachs to managing parallel programs. One approach to the cluster computations management is a scheme with one dispatcher (or one master).
We consider another scheme of cluster menagement. It is a scheme with multidispatching, when each involved computing module has its own dispatch thread and several processing threads [21], [22].
We demonstrate the results of experiments with parallel programms on the base of multidispatching.

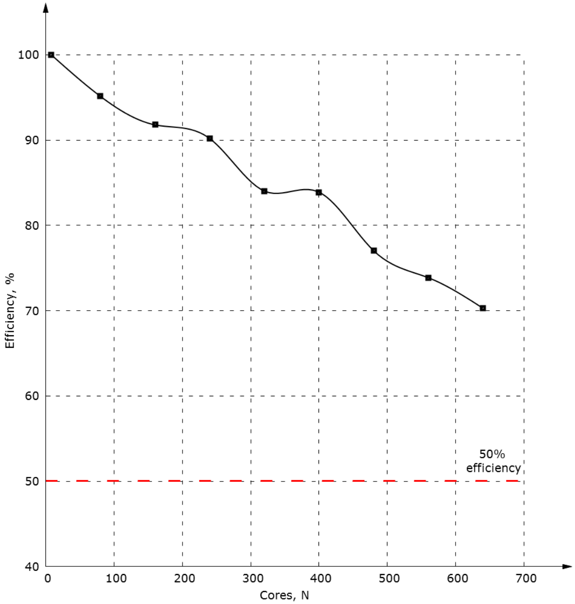
Recursive matrix multiplication (dence, n=8000, Z, 15b)



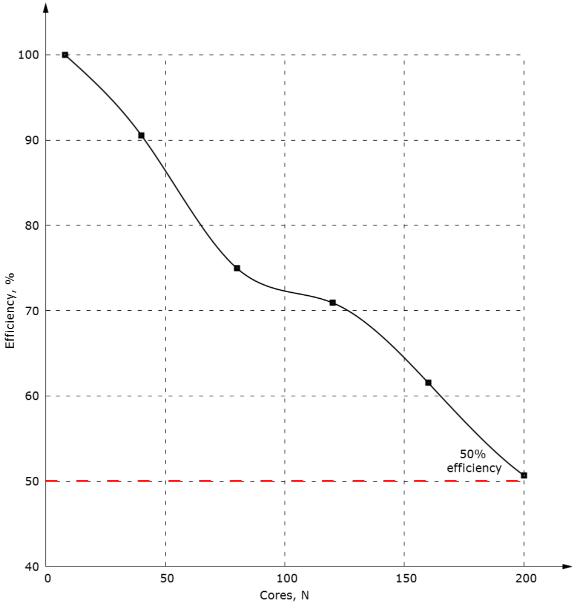
Recursive matrix multiplication (dence, n=12000,Z, 15b)



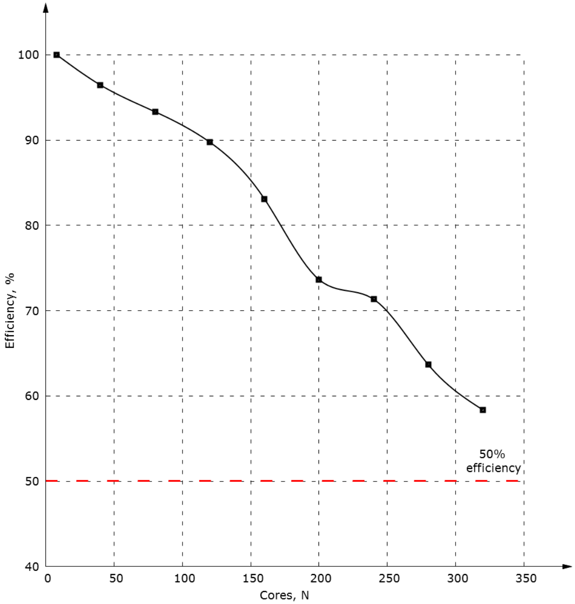
Recursive matrix multiplication (dence, n=14000, Z, 15b)



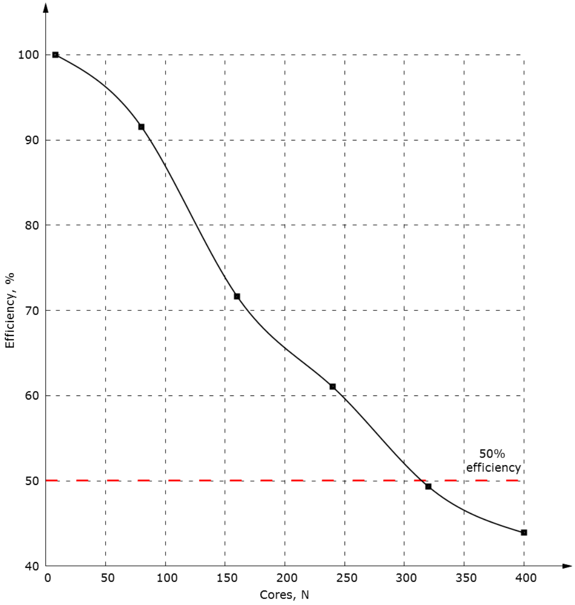
Recursive inversion Strassen (dence, n=8000, double)



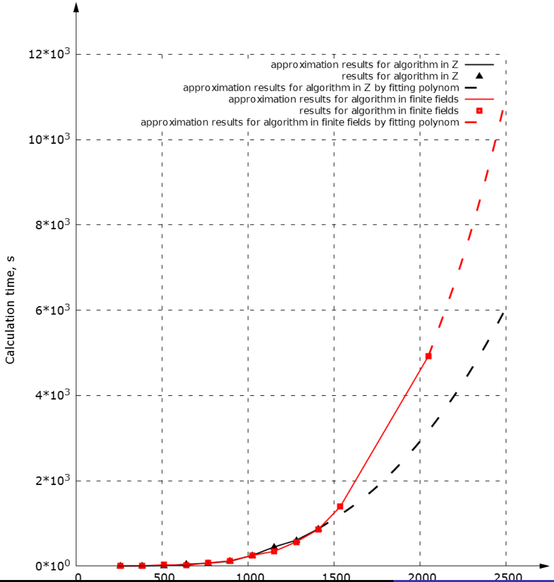
Recursive inversion Strassen (dence, n=16000, double)



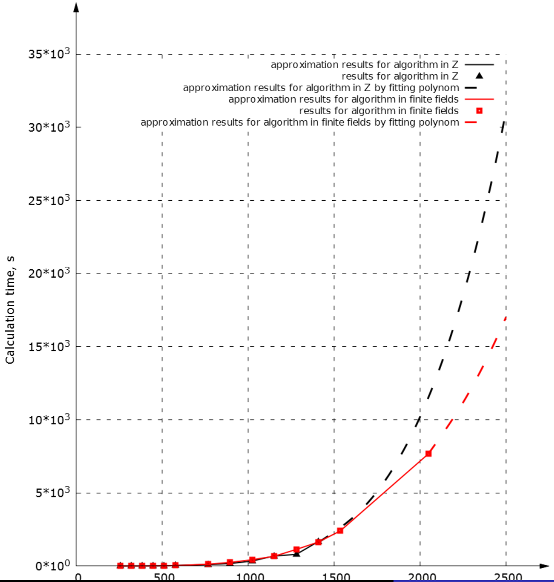
Recursive adjoint and kernel (dence, n=8000, Z)



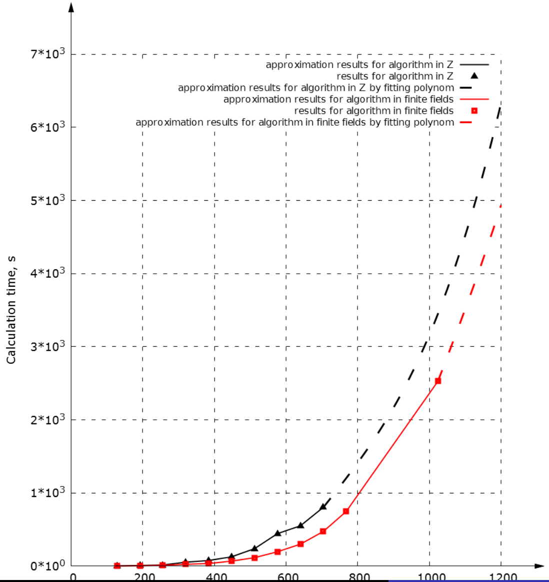
Recursive adjoint and kernel (d=100%, n=100, Z, 15b, CRT+P)



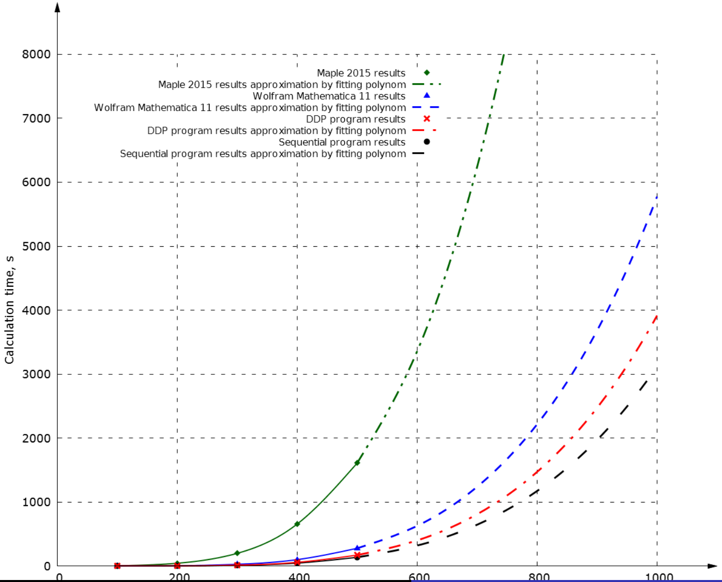
Recursive adjoint and kernel (d=1%, n=10000, Z, 15b, CRT+P)



Recursive adjoint and kernel (d=1%, n=10000, Z, CRT+P)



Comparing sequantional program with Mathematica and MAPLE



Dongarra J. *With Extrim Scale Computing the Rules Have Changed.* In Mathematical Software. ICMS 2016, 5th International Congress, Procdistributed memoryeedings (G.-M. Greuel, T. Koch, P. Paule, A. Sommese, eds.), Springer, LNCS, volume 9725, pp. 3-8, (2016)

Strassen V. Gaussian Elimination is not optimal. Numerische Mathematik. V. 13, Issue 4, 354–356 (1969)

Bunch J., Hopkroft J. *Triangular factorization and inversion by fast matrix multiplication.* Mat. Comp. V. 28, 231-236 (1974)

Malaschonok G.I. *Solution of a system of linear equations in an integral domain,* Zh. Vychisl. Mat. i Mat. Fiz. V.23, No. 6, 1983, 1497-1500, Engl. transl.: USSR J. of Comput. Math. and Math. Phys., V.23, No. 6, 497-1500. (1983)

G.I. Malaschonok. *Algorithms for the solution of systems of linear equations in commutative rings.* Effective methods in Algebraic Geometry, Progr. Math., V. 94, Birkhauser Boston, Boston, MA, 1991, 289-298, (1991) Malaschonok G. *Recursive Method for the Solution of Systems of Linear Equations.* Computational Mathematics. A. Sydow Ed, Proceedings of the 15th IMACS World Congress, Vol. I, Berlin, August 1997), Wissenschaft & Technik Verlag, Berlin, 475-480. (1997)

Malaschonok G. *Effective Matrix Methods in Commutative Domains*, Formal Power Series and Algebraic Combinatorics, Springer, Berlin, 506-517. (2000)

Malaschonok G. *Matrix computational methods in commutative rings.* Tambov, TSU, 213 p. (2002)

Akritas A.G., Malaschonok G.I. *Computation of Adjoint Matrix.* Computational Science, ICCS 2006, LNCS 3992, Springer, Berlin, 486-489.(2006)

Malaschonok G. *On computation of kernel of operator acting in a module* Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], vol. 13, issue 1,129-131 (2008)

Malaschonok G. *On fast generalized Bruhat decomposition in the domains.* Tambov University Reports. Series: Natural and Technical Sciences. V. 17, Issue 2, P. 544-551. (http://parca.tsutmb.ru/src/MalaschonokGI17_2.pdf) (2012)

Malaschonok G. *Generalized Bruhat decomposition in commutative domains.* Computer Algebra in Scientific Computing. CASC'2013. LNCS 8136, Springer, Heidelberg, 2013, 231-242. DOI 10.1007/978-3-319-02297-0_20. arxiv:1702.07248 (2013)

Malaschonok G., Scherbinin A. *Triangular Decomposition of Matrices in a Domain.* Computer Algebra in Scientific Computing. LNCS 9301, Springer, Switzerland, 2015, 290-304. DOI 10.1007/978-3-319-24021-3_22. arxiv:1702.07243 (2015)

Paul, Clayton R. *Fundamentals of Electric Circuit Analysis.* John Wiley & Sons. (2001). ISBN 0-471-37195-5.

Rosenbrock, H.H. *Transformation of linear constant system equations.* Proc. I.E.E. V.114, 541544. (1967)

- Faugere, J.-C. *A new efficient algorithm for computing Gröbner bases (F4)* . Journal of Pure and Applied Algebra. Elsevier Science. Vol. 139, N.1, 61-88. (1999)
- Dumas, J.-G., Pernet, C., Sultan, Z. *Simultaneous computation of the row and column rank profiles.* In: Kauers, M. (Ed.), Proc. ISSAC13. ACM Press, pp. 181-188. (2013)
- Pernet C., Storjohann A. *Time and space efficient generators for quasiseparable matrices.* arXiv:1701.00396 (2 Jan 2017) 29 p. (2017)
- Ilchenko E.A. *An algorithm for the decentralized control of parallel computing process.* Tambov University Reports. Series: Natural and Technical Sciences, Vol. 18, No. 4, 1198-1206 (2013)
- Ilchenko E.A. *About effective methods parallelizing block recursive algorithms.* Tambov University Reports. Series: Natural and Technical Sciences, Vol. 20, No. 5, 1173-1186 (2015)