Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/092L06

A Fast Method for Computing the Inverse of Symmetric Block Arrowhead Matrices

Waldemar Hołubowski[1], *Dariusz Kurzyk*[1,*] *and Tomasz Trawiński*[2]

[1] Institute of Mathematics, Silesian University of Technology, Kaszubska 23, Gliwice 44–100, Poland
[2] Mechatronics Division, Silesian University of Technology, Akademicka 10a, Gliwice 44–100, Poland

Received: 6 Jul. 2014, Revised: 7 Oct. 2014, Accepted: 8 Oct. 2014
Published online: 1 Apr. 2015

Abstract: We propose an effective method to find the inverse of symmetric block arrowhead matrices which often appear in areas of applied science and engineering such as head-positioning systems of hard disk drives or kinematic chains of industrial robots. Block arrowhead matrices can be considered as generalisation of arrowhead matrices occurring in physical problems and engineering. The proposed method is based on LDL^T decomposition and we show that the inversion of the large block arrowhead matrices can be more effective when one uses our method. Numerical results are presented in the examples.

Keywords: matrix inversion, block arrowhead matrices, LDL^T decomposition, mechatronic systems

1 Introduction

A square matrix which has entries equal zero except for its main diagonal, a one row and a column, is called the arrowhead matrix. Wide area of applications causes that this type of matrices is popular subject of research related with mathematics, physics or engineering, such as computing spectral decomposition [1], solving inverse eigenvalue problems [2], solving symmetric arrowhead systems [3], computing the inverse of arrowhead matrices [4], modelling of radiationless transitions in isolated molecules [5,6], oscillators vibrationally coupled with a Fermi liquid [6], modelling of wireless communication systems [7,8].

In this paper we propose a simple and effective method to find the inverse of the symmetric block arrowhead matrices, which have wide applications in mechatronics. In order to illustrate the significance of arrowhead matrices in process of designing mechatronic systems, we should look at this in more details. Colloquially, the mechatronic system is a combination of different elements. It means that devices are made in different technologies which are strongly coupled to each other. The systems are built from following main subsystems such as: mechanical, electromagnetic, electronic and informatics. Such systems can also have subsystems associated with pneumatics, hydraulics, thermal and many others. Considered subsystems are highly differentiated, hence formulation of uniform and simple mathematical model describing their static and dynamic states becomes problematic. The process of preparing a proper mathematical model is often based on the formulation of the equations associated with Lagrangian formalism [9], which is a convenient way to describe the equations of mechanical, electromechanical and other components. As a result of application of Lagrange formalism, we get the mathematical model describing the dynamic of the system. The obtained model is given by second order differential equation, which can be expressed as matrix equation. In matrix notation of equation of the modelled system, it is possible to distinguish matrix of inertia, whose structure corresponds to the structure of the real object – the mechatronic system. The inertia matrices usually are symmetric. Additionally, in many cases these matrices can be expressed as symmetric arrowhead matrices or symmetric block arrowhead matrices. It is typical of the mathematical models describing the following devices:

– electromechanical transducers [10,11,12,13]. This component is included for instance in squirrel-cage induction motors, where the inertia matrix can be represented as a block arrowhead matrix. The number of blocks in the matrix depends on the number of

* Corresponding author e-mail: Dariusz.Kurzyk@polsl.pl

harmonics of the magnetic field in the airgap and the number of rotor bars in a cage [12,13].
–electromechanical transducers with a double stator [14, 15]. This type of transducers can be successfully used as generators in production of wind energy.
–kinematic chains of industrial robots [16]. Depending on the configuration of the open kinematic chain of an industrial robotic manipulator, the inertia matrix can be expressed as block arrowhead matrix, which represents kinematic relationships between actuators and the elements of the Stewart platform [17].
–head-positioning systems of hard disk drives (HDD) [18,19]. Drivetrain of a head-positioning control system can be analysed as a special case of branched robotic manipulator [18,19,20].

The block arrowhead inertia matrices are widely used in modeling of mechatronic systems. Its inverses have important meaning e.g. in:

–reduction the time of designing and development of mentioned device models,
–increasing efficiency of simulation of the modelled systems,
–eliminating torsional vibrations in the drive systems.

Considered inertia matrices depending on the systems can have large sizes, hence there is a need for improvement of methods for block arrowhead matrices inversion.

Generally, matrix inversion is not harder than matrix multiplication [21,22]. Computational complexity of matrix inversion based on Gauss-Jordan elimination is $\mathcal{O}(n^3)$ [23]. Strassen algorithm, which can be used to inverse of matrix with complexity $\mathcal{O}(n^{2.8074})$[21], is more efficient. Coppersmith and Winograd [24,25] show that matrix multiplication can be obtained in $\mathcal{O}(n^{2.3755})$. Now, the fastest algorithm of matrix multiplication running in $\mathcal{O}(n^{2.3727})$ time was performed by Williams [26]. Two last mentioned algorithms are rarely used in practice.

2 Inverse of block arrowhead matrix

2.1 Arrowhead matrix

Let $\hat{A}$ be a square block matrix given in the following way

$$\hat{A} = \begin{bmatrix} \hat{A}_1 & \hat{B}_1 & \hat{B}_2 & \cdots & \hat{B}_{k-1} \\ \hat{B}_1^T & \hat{A}_2 & 0 & \cdots & 0 \\ \hat{B}_2^T & 0 & \hat{A}_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{B}_{k-1}^T & 0 & 0 & \cdots & \hat{A}_k \end{bmatrix}, \tag{1}$$

where $n_1,n_2,...,n_k$ are dimensions of matrices $\hat{A}_1,\hat{A}_2,...,\hat{A}_k$ and $\sum_{i=1}^k n_i = n$. By use of permutation matrix P, the $\hat{A}$ can be transformed to

$$A = P\hat{A}P^T = \begin{bmatrix} A_1 & 0 & 0 & \cdots & B_1^T \\ 0 & A_2 & 0 & \cdots & B_2^T \\ 0 & 0 & A_3 & \cdots & B_3^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_1 & B_2 & B_3 & \cdots & A_k \end{bmatrix}, \tag{2}$$

where $A_1,A_2,...A_k$ are $n_1,n_2,...,n_k$ dimensional matrices, respectively. In particular if $\hat{P}$ is expressed as

$$\hat{P} = \begin{bmatrix} 0 & 0 & \cdots & \mathrm{I}_k \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathrm{I}_2 & \cdots & 0 \\ \mathrm{I}_1 & 0 & \cdots & 0 \end{bmatrix}, \tag{3}$$

where $\mathrm{I}_1,\mathrm{I}_2,...,\mathrm{I}_k$ are identity matrices with dimensions $n_1,n_2,...,n_k$ then

$$A = \hat{P}\hat{A}\hat{P}^T = \begin{bmatrix} \hat{A}_1 & 0 & 0 & \cdots & \hat{B}_1^T \\ 0 & \hat{A}_2 & 0 & \cdots & \hat{B}_2^T \\ 0 & 0 & \hat{A}_3 & \cdots & \hat{B}_3^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{B}_1 & \hat{B}_2 & \hat{B}_3 & \cdots & \hat{A}_k \end{bmatrix}. \tag{4}$$

Hence, the A consists of the same blocks $\hat{A}_1,\hat{A}_2,...,\hat{A}_k$ and $\hat{B}_1,\hat{B}_2,...,\hat{B}_k$ as matrix $\hat{A}$. If we consider permutation matrix $\tilde{P}$ expressed as

$$\tilde{P} = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{5}$$

then entries of $\tilde{A} = \tilde{P}\hat{A}\tilde{P}^T$ are given by $\tilde{A}_{i,j} = \hat{A}_{n-i+1,n-j+1}$, where n is a size of $\hat{A}$.

The form of a matrix given by (1) occurs many times in different cases e.g. during designing the mechatronic system. Blocks of the matrix are related with the structure of a modelled object. Hence the blocks should be invariant under transformation of the matrix. Expressed in (3) changes the structure of the matrix, but the blocks remain unchanged.

2.2 LDL^* decomposition

Let A be a Hermitian positive definite matrix. Decomposition the A into the product of a lower triangular matrix and its conjugate transpose is called Cholesky decomposition. Thus, every Hermitian matrix can be expressed as

$$A = LL^*, \tag{6}$$

where L is a lower triangular matrix. If A is a symmetric positive definite matrix, then A can be factorized into $A = LL^T$. If we consider (6) in following way

$$A = LL^* = L'D^{\frac{1}{2}}(L'D^{\frac{1}{2}}) = L'D^{\frac{1}{2}}(D^{\frac{1}{2}})^*(L')^* = L'D(L')^*, \quad (7)$$

where L' is a lower triangular matrix and D is a diagonal matrix, then we get LDL^* decomposition. This variant of Cholesky decomposition is also useful for the Hermitian nonpositive matrix A. For real symmetric matrices, the factorization has the form $A = LDL^T$. The computational complexity of Cholesky decomposition is $\mathcal{O}(n^3)$ and the most efficient algorithms used for the factorization require $\frac{1}{6}n^3$ operations. The complexity of LDL^* decomposition is the same as Cholesky decomposition.

Let $A = [a_{ij}] \in \mathbf{C}^{n\times n}$ be the Hermitian matrix. The LDL^* decomposition factorizes A into a lower triangular matrix $L = [l_{ij}] \in \mathbf{C}^{n\times n}$, a diagonal matrix $D = [d_{ij}] \in \mathbf{C}^{n\times n}$ and conjugate transpose of L expressed as $A = LDL^*$, where

$$\begin{aligned} d_{ii} &= a_{ii} - \sum_{k=1}^{i-1} l_{ik}\overline{l_{ik}}d_{kk} \\ l_{ij} &= \frac{1}{d_{jj}}(a_{ij} - \sum_{k=1}^{j-1} l_{ik}\overline{l_{jk}}d_{kk}), \quad \text{for } i > j. \end{aligned} \quad (8)$$

If we consider block Hermitian matrix $A = [A_{ij}] \in \mathbf{C}^{n\times n}$, where $A_{ij} \in \mathbf{C}^{n_i\times n_j}$, then LDL^* decomposition is expressed as $A = LDL^*$, where $L = [L_{ij}] \in \mathbf{C}^{n\times n}$ and $D = [D_{ij}] \in \mathbf{C}^{n\times n}$ are given by

$$\begin{aligned} D_{ii} &= A_{ii} - \sum_{k=1}^{i-1} L_{ik}D_{kk}L_{ik}^* \\ L_{ij} &= (A_{ij} - \sum_{k=1}^{j-1} L_{ik}D_{kk}L_{jk}^*)D_{jj}^{-1}, \quad \text{for } i > j. \end{aligned} \quad (9)$$

2.3 Decomposition of a block arrowhead matrix

Consider an arrowhead matrix $A \in \mathbf{R}^{n\times n}$ given in the following way

$$A = \begin{bmatrix} a_1 & 0 & 0 & \cdots & b_1^T \\ 0 & a_2 & 0 & \cdots & b_2^T \\ 0 & 0 & a_3 & \cdots & b_3^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_1 & b_2 & b_3 & \cdots & a_n \end{bmatrix}. \quad (10)$$

As a result of the factorization, we obtain matrices D and L expressed as

$$D = \begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_n \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_1 & l_2 & l_3 & \dots & 1 \end{bmatrix}, \quad (11)$$

where for $1 \le i \le n-1$

$$d_i = a_i, \ \ l_i = \frac{b_i}{d_i} \quad \text{and} \quad d_n = a_n - \sum_{k=1}^{n-1} l_k^2 d_k. \quad (12)$$

The computational complexity of the factorization is $\mathcal{O}(n)$.

Let $A = [A_{ij}] \in \mathbf{R}^{n\times n}$ be a symmetric block arrowhead matrix expressed as matrix in (2). The A can be decomposed into a products of matrices L, D and L^T, where

$$D = \begin{bmatrix} D_1 & 0 & 0 & \dots & 0 \\ 0 & D_2 & 0 & \dots & 0 \\ 0 & 0 & D_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & D_n \end{bmatrix} \quad L = \begin{bmatrix} F_1 & 0 & 0 & \dots & 0 \\ 0 & F_2 & 0 & \dots & 0 \\ 0 & 0 & F_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_1 & L_2 & L_3 & \dots & F_k \end{bmatrix}. \quad (13)$$

Matrices F_i and D_i are obtained as results of LDL^T decomposition of a matrix A_i and $L_i = B_i(F_i^{-1})^T D_i^{-1}$, where $1 \le i \le k-1$. Next, F_k and D_k are results of the factorization of a matrix $\tilde{A}_k = A_k - \sum_{i=1}^{k-1} L_i D_i L_i^T$. Suppose that $M(n)$ denotes computational complexity of inversion of n dimensional matrix. It is possible to show that the matrix inversion is equivalent to the matrix multiplication [22], thus $M(n)$ denotes also complexity of matrix multiplication. Assume that $n_{max} = \max(n_1, n_2, ..., n_k)$. We need perform k times LDL^T decomposition in $\mathcal{O}(n_i^3)$, which can be evaluated as $\mathcal{O}(k\cdot n_{max}^3)$. Hence complexity of the proposed method of the matrix decomposition is $\mathcal{O}(k\cdot(n_{max}^3 + M(n_{max})))$. So, if $n \gg n_{max}$ then the presented algorithm should be faster.

2.4 Inversion of block arrowhead matrix

Let A be a matrix given as (2) and D, L be matrices received as a result of the decomposition of A. Inverse matrix A^{-1} is given in the following way

$$\begin{aligned} D^{-1} &= \begin{bmatrix} D_1^{-1} & 0 & \dots & 0 \\ 0 & D_2^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_n^{-1} \end{bmatrix}, \\ L^{-1} &= \begin{bmatrix} F_1^{-1} & 0 & \dots & 0 \\ 0 & F_2^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -F_k^{-1}L_1F_1^{-1} & -F_k^{-1}L_2F_2^{-1} & \dots & F_k^{-1} \end{bmatrix}. \end{aligned} \quad (14)$$

Hence, the inverse of A can be expressed as

$$A^{-1} = (L^{-1})^T D^{-1} L^{-1}. \quad (15)$$

The matrix multiplication is a quite computationally expensive operation, hence the way of obtaining the matrix given by (15) is not effective.

It is easy to check that

$$(L^{-1})^T D^{-1} L^{-1} =$$

$$\begin{bmatrix} R_1 & 0 & 0 & \cdots & S_1^T \\ 0 & R_2 & 0 & \cdots & S_2^T \\ 0 & 0 & R_3 & \cdots & S_3^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_1 & S_2 & S_3 & \cdots & R_k \end{bmatrix} + \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1k-1} & 0 \\ U_{21} & U_{22} & \cdots & U_{2k-1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ U_{k-11} & U_{k-12} & \cdots & U_{k-1k-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (16)$$

where

$$\begin{aligned} R_i &= (F_i^{-1})^T D_i^{-1} F_i^{-1} \\ S_i &= -(F_k^{-1})^T D_k^{-1} F_k^{-1} L_i F_i^{-1} \\ U_{ij} &= (F_k^{-1} L_i F_i)^T D_k^{-1} F_k^{-1} L_j F_j. \end{aligned} \quad (17)$$

Blocks R_i, S_i and U_{ij} are obtained in $\mathcal{O}(k^2 M(n_{max}))$. Hence the inversion of block arrowhead can be performed in $\mathcal{O}(k \cdot n_{max}^3 + k^2 M(n_{max}))$.

3 Example

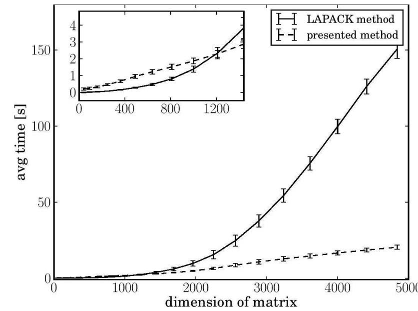
Consider the following example. Let R be a k dimensional random matrix. We create a matrix $B = (R + R^T)/2$ which is symmetric. It is easy to check that $\mathrm{I} \otimes B$ is a symmetric block diagonal matrix, where I is an l dimensional identity matrix. Assume that E is a $k \cdot l$ dimensional zero matrix except last k columns with entries equal to 1. In last step, we create block arrowhead matrix $A = \mathrm{I} \otimes B + E + E^T$. It means that A can be expressed in following way

$$A = \begin{bmatrix} B & 0 & 0 & \cdots & \mathbf{1} \\ 0 & B & 0 & \cdots & \mathbf{1} \\ 0 & 0 & B & \cdots & \mathbf{1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \cdots & B + 2\mathbf{1} \end{bmatrix}, \quad (18)$$

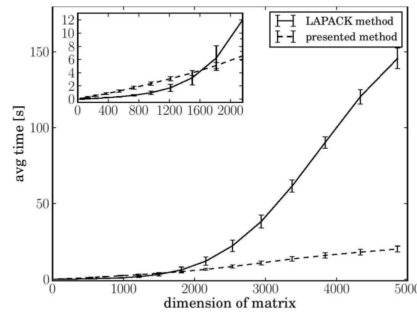
where $\mathbf{1}$ is a matrix with all elements equal 1. The proposed construction of the matrix A provides an easy way to generate a block arrowhead matrix with a size dependent on l. Next, our algorithm will be compared with a fast algorithm for the matrix inverse computations implemented in a library for numerical linear algebra `LAPACK`. A numerical investigation will be performed by the usage of programming language `python`. In each step, we increase the size of matrix A and we take average operating time of algorithms. Results for $k = 10$ and $k = 15$ are illustrated in figure 1. It is easy to check that our algorithms provide better results for a large matrix.

4 Conclusions

In this paper, we propose a variant of LDL^T decomposition of block arrowhead matrices and effective algorithm for computing the inverse of an arrowhead matrix. Presented method need $\mathcal{O}(k \cdot n_{max}^3 + k^2 M(n_{max}))$ time. In case of large sparse matrices, the method is more effective than other wide applied algorithms for matrix inversion. Performed numerical example shows that obtained acceleration of calculations can be significant.



(a)



(b)

Figure 1: Times of computing the inverse of block arrowhead matrix by algorithm implemented in `LAPACK` library (solid line) and proposed in our paper (dash line). a) results for matrices with blocks of size equal to 10, b) results for matrices with blocks of size equal to 15 .

References

[1] F. Diele, N. Mastronardi, M. Van Barel, E. Van Camp, On computing the spectral decomposition of symmetric arrowhead matrices, In Computational Science and Its Applications – ICCSA. Springer Berlin Heidelberg, 932–941 (2004).

[2] Y. Yuan, Generalized inverse eigenvalue problems for symmetric arrow-head matrices, International Journal of Computational and Mathematical Sciences **4**, 268–271 (2010).

[3] G. A. Gravvanis, Solving symmetric arrowhead and special tridiagonal linear systems by fast approximate inverse preconditioning, Journal of Mathematical Modelling and Algorithms **1**, 269–282 (2002).

[4] H. Saberi Najafi, S. Edalatpanah, G. A. Gravvanis, An efficient method for computing the inverse of arrowhead matrices, Applied Mathematics Letters **33**, 1–5 (2014).

[5] M. Bixon, J. Jortner, Intramolecular radiationless transitions, The Journal of Chemical Physics **48**, 715–726 (2003).

[6] J. W. Gadzuk, Localized vibrational modes in Fermi liquids, general theory, Physical Review B **24**, 1651–1663 (1981).

[7] L. Shen, B. W. Suter, Bounds for eigenvalues of arrowhead matrices and their applications to hub matrices and wireless communications, EURASIP Journal on Advances in Signal Processing **58** (2009).

[8] H. T. Kung, B. W. Suter, A hub matrix theory and applications to wireless communications, EURASIP Journal on Advances in Signal Processing, (2007).

[9] P. C. Müller, Modelling and control of mechatronic systems by the descriptor approach, Journal of Theoretical and Applied Mechanics **43**, 593–607 (2005).

[10] A. Mężyk, T. Trawiński, Modeling of drive system with vector controlled induction machine coupled with elastic mechanical system, Recent Advances in Mechatronics, Springer-Verlag Berlin Heidelberg, 248–252 (2007).

[11] T. Trawiński, K. Kluszczyński, W. Kołton, Lumped parameter model of double armature VCM motor for head positioning system of mass storage devices, XXI Sympozjum Środowiskowe PTZE, Lubliniec 2011, Przegląd Elektrotechniczny **87**, 184–187 (2011).

[12] T. Trawiński, Z. Pilch, W. Burlikowski, K. Kluszczyński, Operation fundamentals of the electromagnetic torque generator part I – Properties of the pulsating torque, Technical and Economic Aspect of Modern Technology Transfer in Context of Integration with European Union, **II**, 255–260 (2004).

[13] Z. Pilch, W. Burlikowski, T. Trawiński, K. Kluszczyński, Operation fundamentals of the electromagnetic torque generator part II – Design process, Technical and Economic Aspect of Modern Technology Transfer in Context of Integration with European Union, **II**, 261–265 (2004).

[14] J. Poza, E. Oyarbide, D. Roye, M. Rodriguez, Unified reference frame dq model of the brushless doubly fed machine. Electric Power Applications, IEEE Proceedings **153**, 726–734 (2006).

[15] I. Sarasola, J. Poza, E. Oyarbide, M. A. Rodriguez, Stability Analysis of a Brushless Doubly-Fed Machine under Closed Loop Scalar Current Control, IEEE Industrial Electronics, IECON 2006 – 32nd Annual Conference on, 1527–1532 (2006).

[16] M. Da Silva, O. Bruls, B. Paijmans, W. Desmet, H. Van Brussel, Concurrent simulation of mechatronic systems with variable mechanical configuration, Proceedings of ISMA2006, AMS1 – Multibody Dynamics For Multi-Physics Applications, 69–79 (2006).

[17] P. Cruz, R. Ferreira, J. Sequeira, Kinematic Modeling of Stewart-Gough Platforms, ICINICO 2005 – Robotics and Automation, 93–99 (2005).

[18] D. Słota, T. Trawiński, R. Wituła, Inversion of Dynamic Matrices of HDD Head Positioning System, Applied Mathematical Modelling **35**, 1497–1505 (2011).

[19] T. Trawiński, R. Wituła, Modeling of HDD head positioning systems regarded as robot manipulators using block matrices, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), 129–144 (2010).

[20] S. M. Suh, C. C. Chung, S. H. Lee, Discrete-time LQG/LTR dual-stage controller design in magnetic disk drives, IEEE Transaction on Magnetics **37**, 1891–1895 (2001).

[21] M. D. Petković, P. S. Stanimirović, Generalized matrix inversion is not harder than matrix multiplication, Journal of computational and applied mathematics **230**, 270–282 (2009).

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, **2**, Cambridge: MIT press (2001).

[23] K. E. Atkinson, An introduction to numerical analysis, John Wiley & Sons Inc, (1989).

[24] D. Coppersmith, S. Winograd, On the asymptotic complexity of matrix multiplication, SIAM Journal on Computing **11**, 472–492 (1982).

[25] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, Journal of symbolic computation **9**, 251–280 (1990).

[26] V. V. Williams. Breaking the Coppersmith-Winograd barrier, unpublished results (2011).

Waldemar Hołubowski is a Professor at Faculty of Applied Mathematics at Silesian University of Technology. He obtained his PhD and DSc in pure mathematics at Saint Petersburg State University. His research interest are group theory, graph theory, algebraic structures, especially matrices and their applications in mathematics and engineering sciences. He is a referee of many mathematical journals.

Dariusz Kurzyk is a PhD student at Faculty of Applied Mathematics at Silesian University of Technology. He is also research assistant in Institute of Theoretical and Applied Informatics, Polish Academy of Sciences. His research interests are in the areas of applied mathematics and mathematical physics especially linear algebra, algorithms and data structures, parallel computations, artificial intelligence, quantum information theory and quantum algorithms.

Tomasz Trawiński is an assistant professor in the Department of Mechatronics, Faculty of Electrical Engineering of the Silesian University of Technology since 2000. His professional interests revolve around: mechatronics, automation and robotics, electromechanical transducers. Over the past few years his research focuses on methods of modeling and testing mass storage devices components, so-called branched manipulators and block matrix algebra utilization in technical computations.