ISBN 978-83-60810-22-4

The incomplete factorization preconditioners applied to the GMRES(m) method for solving Markov chains

Beata Bylina Jarosław Bylina
Institute of Mathematics
Marie Curie-Skłodowska University
Pl. M. Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: beata.bylina@umcs.pl, jaroslaw.bylina@umcs.pl

***Abstract*—This paper is a review and a comparison of some preconditioners based on incomplete factorizations of matrices — for matrices describing Markov chains. Three preconditioners are considered: ILU(0), ILU3, IWZ(0). Two of them (ILU(0), ILU3) are based on the LU factorization, the latter (IWZ(0))— on the WZ factorization. The preconditioners are investigated in respect of their usability for decreasing number of iterations in a projection method, namely GMRES(m). To chose the best preconditioner for such methods, authors introduce a measure called *iteration speed-up* (p) and some of its relatives, as well as they define a function giving an average number of restarts needed to achieve a given accuracy for matrices from a some set (Is). These measures are studied for two different cases of matrices describing Markov chains to compare influence of the examined incomplete preconditioners for GMRES(m).**

I. Introduction and motivation

While modelling probabilities stationary distributions (independent of time) with Markov chains, we obtain a following linear equation system:

$$\mathbf{Q}^T\mathbf{x} = \mathbf{0}, \qquad \mathbf{x} \geq \mathbf{0}, \qquad \mathbf{x}^T\mathbf{e} = 1, \tag{1}$$

where $\mathbf{Q}$ is a transition rate matrix, $\mathbf{x}$ is an unknown vector of states' probabilities and $\mathbf{e} = (1, 1, ..., 1)^T$. The matrix $\mathbf{Q}$ is a singular square one of size $n \times n$, of rank $n-1$, with a weakly dominant diagonal, usually a sparse, large and ill-conditioned one. These traits of $\mathbf{Q}$ cause the need to treat the system (1) specially.

One of the most popular methods to solve the system 1 is the GMRES method [13]. The full GMRES algorithm (that is: GMRES(n), n being the size of the system) is guaranteed to converge in at most n steps when we used full precision arithmetic (that means: no restarts are needed), but it is not very useful for large systems of equations, because a good approximate solution is often computed quite early, after very few iterations.

Moreover, the traditional GMRES (without restarts) requires quite a lot of space (additional $O(n)$ for every iteration, so $O(n^2)$ for n iterations), which is a big disadvantage for huge systems coming from Markov chains. On the other hand, GMRES(m) (that is, the GMRES method restarted after m iterations) requires at most $O(m \cdot n)$ additional space for computations, regardless of the number k of restarts (because after every restart the working space can be reused), but the achieved accuracy can be comparable (for an appropriate m, of course, which is often not easy to choose) to the accuracy achieved with GMRES after the same total number of iterations — that is $k \cdot m$ — but in the latter case the space needed is $O(k \cdot m \cdot n)$. So, we investigate a restarted version, GMRES(m), as a potentially more economical method for huge systems.

The very concept of the preconditioning is almost as old as iterative methods [8]. One of the most famous preconditioning techniques is the incomplete factorization of the original matrix $\mathbf{Q}$. The idea of the incomplete factorization was presented by Buleev [3], [4] and Varga [15]. The papers that popularized the incomplete factorizations were [9], [10].

There is a need for preconditioners that are fast, stable, scalable, easy to parallelize and that generate a small fill-in. In [1], [2] preconditioners for Krylov subspace methods for solving large singular linear systems arising from Markov modeling are considered.

The incomplete LU (ILU) factorization process computes a sparse lower triangular matrix $\mathbf{L}$ and a sparse upper triangular matrix $\mathbf{U}$. Here we discuss the ILU(0) factorization, the simplest form of the ILU preconditioners. ILU(0) consists in taking the zero pattern as the original matrix $\mathbf{Q}$. Using ILU(0) for solving Markov chains was shown in [14].

ILU3 is another kind of incomplete factorization based on the LU factorization. Here, the factors are nonzero only on their three central diagonals — for the matrix $\mathbf{L}$ it is the main diagonal and the one directly below, and for the matrix $\mathbf{U}$ it is the main diagonal and the one directly above. This factorization was not used to Markov chains so we wanted to test it.

The incomplete WZ factorization is originally described in some previous works [5]. In [6] we discussed its performance for GMRES. This work is a step forward in the investigation

This work was partially supported within the project N N516 479640 of the Ministry of Science and Higher Education of the Polish Republic (MNiSW) *Modele dynamiki transmisji, sterowania zatłoczeniem i jakością usług w Internecie*.

of such incomplete preconditioners and compares some kinds of IWZ and ILU.

We are concerned in the influence of the preconditioner's structure on the GMRES(m) method, so we are considering three incomplete factorization methods, namely ILU(0), ILU3 i IWZ(0). These methods differ with the structure of the factors and the number of nonzeros. In factorizations IWZ(0) and ILU(0) the number of nonzeros is exactly the same as in the original matrix $\mathbf{Q}$ but factors ($\mathbf{L}$ and $\mathbf{U}$ in ILU(0), $\mathbf{W}$ and $\mathbf{Z}$ in IWZ(0)) have different structures. The ILU3 factorization has usually less nonzero elements the the original matrix $\mathbf{Q}$ and the structure of the factors is similar to ILU(0). ILU3 is less accurate than ILU(0) and IWZ(0), so we can expect worse results.

We consider an impact of the incomplete factorization preconditioners on the GMRES(m) method for the numerical solution of Markov chains. We study the relationship between the number of iterations, the convergence rate of the GMRES(m) method and properties of the matrix $\mathbf{Q}$ and the structure of preconditioners. For better understanding of the behavior of the GMRES(m) convergence and its preconditioners we introduce some measures — the iteration speed-up among others (see Section IV-A).

The research was carried out for two case. The first case are matrices that have not got any particular structure and we assume that the matrix row and column ordering is given and cannot be changed. The second case are matrices of a Markov chain known from the literature as the epidemic model and these matrices has got a structure. For all those (sparse) matrices we introduced another measure — matrix density.

The rest of the paper is organized as follows. Section II recalls briefly the incomplete preconditioning. Section III presents two test cases. Section IV describes conducted numerical experiments. Section V contains some conclusions.

II. Incomplete preconditioners

The convergence rate of iterative methods depends on properties of the coefficient matrix of the linear system. If the matrix $\mathbf{Q}$ is ill-conditioned, this can make the convergence of iterative methods slow. One way to prevent such problems is to transform the system (1) into an equivalent system (that is, having the same solution), but with better numerical properties. Such a transformation can be done by preconditioning, that is by converting the system (1) into:

$$\mathbf{M}^{-1}\mathbf{Q}^T\mathbf{x} = \mathbf{0}, \qquad \sum_{i=1}^{n} x_i = 1, \qquad \mathbf{x} \geq \mathbf{0}, \tag{2}$$

where the nonsingular matrix $\mathbf{M}$ (known as a preconditioner) approximates the matrix $\mathbf{Q}^T$ in a manner. The system (2) has the same solutions as (1) but it is (hopefully) better conditioned.

The matrix $\mathbf{M}$ should have the following properties:

- its use should entail low memory requirements;
- its inverse should be cheaply applicable;
- the transformed problem (2) should converge faster (in shorter computational time) than the original problem.

Of course, there is a clear conflict among these three requirements, especially for the construction of general purpose preconditioners.

Generally, computing and using a good preconditioner is an expensive task consisting of finding the matrix $\mathbf{M}$ and its inverse. If the preconditioning is to be used, that cost should be refunded by a reduced number of iterations needed to acquire a required accuracy — or by using the same preconditioner for various linear systems.

The preconditioner matrix is usually built on the base of the original coefficients of the matrix $\mathbf{Q}$.

A. ILU(0) preconditioner

The incomplete LU factorization (denoted ILU) is based on the well known LU factorization, where a lower triangular matrix (with ones on the diagonal) $\widetilde{\mathbf{L}}$ and an upper triangular matrix $\widetilde{\mathbf{U}}$ are found and where the preconditioner matrix $\mathbf{M} = \widetilde{\mathbf{L}}\widetilde{\mathbf{U}}$ is a kind of approximation for the matrix $\mathbf{Q}^T$.

There are many variants of ILU, the most straightforward being ILU(0) [14]. In ILU(0) the computations are conducted as in the traditional (complete) LU factorization (that is, the Gaussian elimination), but any new nonzero element (l_{ij} and u_{ij}) arising in the process is dropped if it appears in the place of a zero element in the original matrix $\mathbf{Q}^T$. Hence, the factors together have the same number of nonzeros as the original matrix $\mathbf{Q}^T$. Thereby, the most important problem of the factorization of sparse matrices — the fill-in (which consists in appearing nonzero elements in new matrices on the places of zero elements in the original matrix, what makes dense the output factors and renders impossible their packed storage) — is eliminated. At the expense of accuracy, of course.

After ILU(0) we have:

$$\mathbf{Q}^T = \widetilde{\mathbf{L}}\widetilde{\mathbf{U}} + \mathbf{R}_{LU}, \tag{3}$$

where $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{U}}$ are (respectively) the lower triangular matrix and the upper triangular matrix and the remainder matrix $\mathbf{R}_{LU}$ is hoped to be small in a sense.

Let $\mathbf{M} = \widetilde{\mathbf{L}}\widetilde{\mathbf{U}}$, then $\mathbf{M}^{-1} = \widetilde{\mathbf{U}}^{-1}\widetilde{\mathbf{L}}^{-1}$ and the equation (2) takes the shape:

$$\widetilde{\mathbf{U}}^{-1}\widetilde{\mathbf{L}}^{-1}\mathbf{Q}^T\mathbf{x} = \mathbf{0}, \qquad \sum_{i=1}^{n} x_i = 1, \qquad \mathbf{x} \geq \mathbf{0}. \tag{4}$$

Let $\mathbf{S}_{LU} = \widetilde{\mathbf{U}}^{-1}\widetilde{\mathbf{L}}^{-1}\mathbf{Q}^T$. Now, the equation (4) takes the shape:

$$\mathbf{S}_{LU}\mathbf{x} = \mathbf{0}, \qquad \sum_{i=1}^{n} x_i = 1, \qquad \mathbf{x} \geq \mathbf{0}. \tag{5}$$

The following Octave code was used to generate the ILU(0) factors used in this article. For a given matrix $\mathbf{Q}$ two triangular matrices are constructed. The matrix (L) is lower triangular (the main diagonal of $\mathbf{L}$ is filled with ones only) and the matrix (U) — upper triangular. The time complexity is $O(n^3)$.

```
function [L,U]=ilu0(Q)
  [n,n]=size(Q); U=Q; L=zeros(n);
  for i=1:n, L(i,i)=1.0; end;
  for k=2:n,
    for i=1:k-1,
      if Q(k,i) != 0,
        L(k,i)=U(k,i)/U(i,i);
        for j=i+1:n,
          U(k,j)=U(k,j)-L(k,i)*U(i,j);
        end;
      end;
    end;
    for i=1:n,
      if Q(k,i)==0, U(k,i)=0; end;
    end;
  end;
  for i=2:n,
    for j=1:n,
      if i>j, U(i,j)=0; end;
    end;
  end;
end;
```

B. ILU3 preconditioner

ILU3 is an incomplete LU factorization conducted quite similarly to ILU(0), but the structure of output matrices $\mathbf{L}$ and $\mathbf{U}$ have nothing to do with the structure of the input matrix $\mathbf{Q}^T$. The output matrices simply consist of diagonals: the main one and its lower neighbor diagonal ($\mathbf{L}$) or the main one and its upper neighbor diagonal ($\mathbf{U}$).

As for now, this factorization was not used in solving Markov chains. However, it is quite fast ($O(n)$) and easy to use in parallel. The code is shown below.

```
function [L,U]=ilu3(Q)
  [n,n]=size(Q); U=Q; L=zeros(n);
  for i=1:n, L(i,i)=1.0; end;
  for k=2:n,
    i=k-1;
    L(k,i)=U(k,i)/U(i,i);
    U(k,k)=U(k,k)-L(k,i)*U(i,k);
    if (k<n),
      U(k,k+1)=U(k,k+1)-L(k,i)*U(i,k+1);
    end;
  end;
  for j=1:n,
    for i=1:n,
      if (i>j),U(i,j)=0;end;
      if (i+2<=j), U(i,j)=0;end;
    end;
  end;
end;
```

C. IWZ(0) preconditioner

The incomplete WZ (denoted IWZ) factorization is originally described in a previous works [5]; here we only recall it. The WZ factorization (on which IWZ is based) consists in decomposition of the given matrix ($\mathbf{Q}^T$ in the paper) into a product of two matrices: $\mathbf{W}$ and $\mathbf{Z}$ (Fig. 1).

The incomplete WZ factorization (IWZ) is based on the WZ factorization described above, where we find matrices $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{Z}}$ (of the form of the matrices $\mathbf{W}$ and $\mathbf{Z}$ shown in Fig. 1) and the product $\widetilde{\mathbf{W}}\widetilde{\mathbf{Z}}$ is a kind of approximation for the matrix $\mathbf{Q}^T$.

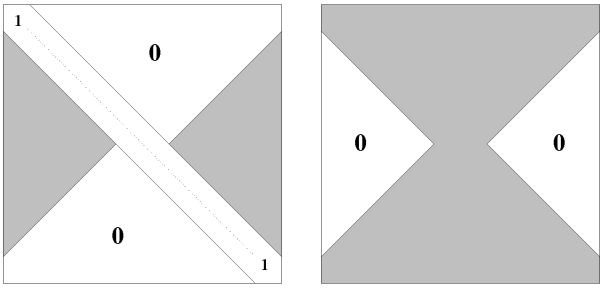

Fig. 1. The form of the output matrices in the WZ factorization (left: $\mathbf{W}$; right: $\mathbf{Z}$)

In IWZ(0) the computations are conducted as in the complete WZ factorization, but any new nonzero elements (w_{ij} and z_{ij}) arising in the process are dropped if they appear in the place of a zero element in the original matrix $\mathbf{Q}^T$. Hence, the factors together have the same number of nonzeros as the original matrix $\mathbf{Q}^T$. It is worth noting that we got the inverse of $\widetilde{\mathbf{W}}$ very easily, because [16]:

$$\widetilde{\mathbf{W}}^{-1} = (-1) \cdot (\widetilde{\mathbf{W}} - \mathbf{I}) + \mathbf{I} \qquad (6)$$

$$\text{(just like} \quad \mathbf{W}^{-1} = (-1) \cdot (\mathbf{W} - \mathbf{I}) + \mathbf{I}). \qquad (7)$$

After IWZ(0) we have:

$$\mathbf{Q}^T = \widetilde{\mathbf{W}}\widetilde{\mathbf{Z}} + \mathbf{R}_{WZ}, \qquad (8)$$

where $\widetilde{\mathbf{W}}$ and $\widetilde{\mathbf{Z}}$ are (respectively) matrices of the form of $\mathbf{W}$ and $\mathbf{Z}$ from Fig. 1 and the remainder matrix $\mathbf{R}_{WZ}$ is supposed to be small in a sense.

The time complexity is $O(n^3)$ — just like for ILU(0).

Here is an Octave code that shows the sequence of operations that must be followed and that provides some feel for the applicability of the algorithm. This (and previous, that is for ILU(0) and ILU3) code is not meant to represent production versions of the algorithms.

```
function [W,Z]=iwz0(Q)
  [n,n]=size(Q); Z=Q; W=zeros(n);
  for i=1:n, W(i,i)=1.0; end;
  for k=1:n/2-1,
    k2=n-k+1;
    det=Z(k,k)*Z(k2,k2)-Z(k2,k)*Z(k,k2);
    for i=k+1:k2-1,
      if Q(i,k)!=0,
        W(i,k)=(Z(k2,k)*Z(i,k2)
               -Z(k2,k2)*Z(i,k))/det;
      end;
      if Q(i,k2)!=0,
        W(i,k2)=(Z(k,k2)*Z(i,k)
                -Z(k,k)*Z(i,k2))/det;
      end;
      for j=k+1:k2-1,
        if Q(i,j)!=0,
          Z(i,j)=Z(i,j)+W(i,k)*Z(k,j)
                +W(i,k2)*Z(k2,j);
        end;
```

TABLE I
THE ESSENTIAL CHARACTERISTICS OF THE MATRICES USED IN THE TESTS

Group	matrix ID	n	nz	d
A	1	100	1190	11.9
B	2	100	388	3.9
A	3	1500	37955	25.3
B	4	1500	5873	3.9
A	5	3000	120590	40.2
B	6	3000	11636	3.9

```
      end;
    end;
  end;
  for i=1:n,
    for j=1:n,
      if ((i>j)&(i<n-j+1))
       | ((i<j)&(i>n-j+1)),
                 Z(i,j)=0; end;
    end;
  end;
end;
```

III. Test cases

Here we shortly present two cases which were used to investigate the influence of our preconditioners (ILU(0), ILU3 and IWZ(0)) on the convergence of GMRES(m). We assume that the investigated matrix cannot be a subject for any reordering.

A. Case I

The matrices of the Case I were created randomly, with given some parameters as n (the number of rows) and nz (the number of nonzeros) as well as the range of the elements outside the diagonal. For every nonzero element, its indices (that is the number of its row and the number of its column) were randomly (uniformly) chosen as well as its value. Then, the diagonal elements were computed (from: $q_{ii} = -\sum_{j\neq i} q_{ij}$) to get a correct transition rate matrix.

In Table I the essential characteristics of the matrices are presented (n is the number of rows and columns of the matrix, nz is the number of nonzeros in the matrix, $d = nz/n$).

For these matrices we can observe, that the matrices might have the same size and a different value of d. The matrices were divided into two groups, the first group include matrices with $d > 8$ (Group A), the second group include matrices with $d \leq 8$ (Group B).

The structure of the matrix with ID=3 is shown in Fig. 2. We can see that the matrices from Case I have no particular structure. It is worth noting that IWZ(0) and ILU(0) for them have not such a structure either, because their structures are based on structures of $\mathbf{Q}$. And ILU3 preconditioner is just a tri-diagonal — as usual.

B. Case II

The matrix of Case II was generated from a standard two-dimensional Markovian model [7], [11]. This particular example has been taken from [11], [12]. The states of the chain are described with two numbers (u, v), $u = 0, \ldots, N_x$, $v = 0, \ldots, N_y$ (here $N_x = 64$, $N_y = 16$). The matrix describing the two-dimensional Markov chain has a structure shown in Fig. 3.

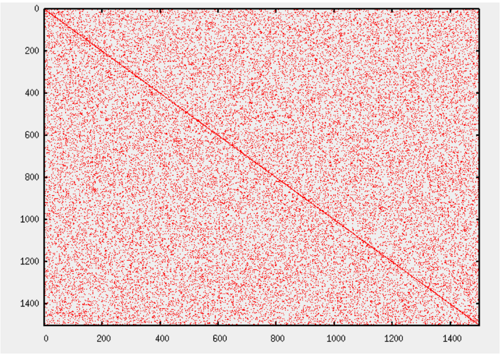

Fig. 2. The structure of the matrix with ID=3 of Case I

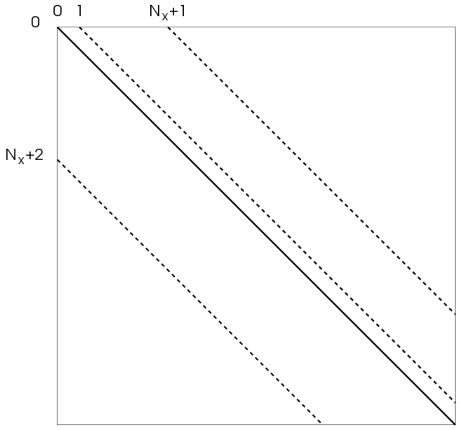

Fig. 3. The structure of the matrix of Case II

It is worth noting that for the matrix of Case II, the investigated preconditioners are also structured. It is so because ILU(0) and IWZ(0) inherit their structures after the original matrix (Fig. 3). But ILU3 is here somewhat a special case, because here it reduces to the Jacobi preconditioner (that is $\mathbf{L} = \mathbf{I}$ and $\mathbf{U} = \mathrm{diag}(q_{ii})$).

IV. Experimental results

The main goal of the numerical experiments was to test the incomplete factorization preconditioners in respect of their usability for the GMRES(m) method for matrices arising from Markov chains. The experiment was performed on a Pentium IV 2.8GHz computer with 1GB RAM under the Debian GNU/Linux operating system. We used a high-level programming language, namely Octave (a GNU equivalent of MATLAB).

A vector $\mathbf{x}^{(0)} = (x_i^{(0)})$ with $x_i^{(0)} = \frac{1}{i}$ was chosen as an initial vector. As a measure of accuracy of the solution we chose the 2-norm of the residual:

$$\varepsilon^{(k)}(\mathbf{Q}) = ||\mathbf{0} - \mathbf{Q}^T\mathbf{x}^{(k)}||_2 = ||\mathbf{Q}^T\mathbf{x}^{(k)}||_2. \qquad (9)$$

TABLE II
NUMBER OF ITERATIONS NEED TO ACHIEVE A GIVEN ACCURACY $\varepsilon^{(k)} < 10^{-16}$ FOR THE SELECTED VALUES OF THE PARAMETER m FOR CASE I (MATRICES FORM GROUP A ARE PRINTED NORMALLY, THOSE FROM GROUP B ARE DISTINGUISHED WITH A BOLD FONT)

m	method	matrix ID					
		1	**2**	3	**4**	5	**6**
$m = 1$	GMRES(m)	45	**87**	44	**185**	42	**271**
	IWZ(0)G–(m)	14	**29**	11	**33**	10	**34**
	ILU(0)G–(m)	14	**28**	12	**33**	11	**33**
	ILU3G–(m)	31	**66**	24	**80**	21	**84**
$m = 5$	GMRES(m)	8	**17**	7	**30**	6	**32**
	IWZ(0)G–(m)	3	**6**	3	**7**	2	**7**
	ILU(0)G–(m)	4	**6**	3	**7**	3	**7**
	ILU3G–(m)	6	**14**	5	**16**	5	**17**
$m = 10$	GMRES(m)	4	**8**	4	**14**	3	**15**
	IWZ(0)G–(m)	2	**3**	2	**4**	1	**4**
	ILU(0)G–(m)	3	**4**	3	**5**	3	**4**
	ILU3G–(m)	3	**7**	3	**8**	3	**9**

Above (and throughout the whole paper), as k, we consider the number of external iterations of GMRES(m), that is the number of restarts.

The accuracy has been studied experimentally for the matrices of Case I and Case II. We studied both the number of iterations needed to achieve a given accuracy, and the convergence rate. The stop condition used here is that the 2-norm of the residual (that is $\varepsilon^{(k)}(\mathbf{Q}) = ||\mathbf{Q}^T\mathbf{x}^{(k)}||_2$) is less than 10^{-16} (such a chosen value is quite real if we are to find probabilities of unlikely but important events — as a packet loss or a channel jamming — precisely). To improve the readability we did not use $\varepsilon^{(k)}(\mathbf{Q})$ in the results presentation, but rather:

$$acc(\mathbf{Q}, i) = -\log_{10} \varepsilon^{(i)}(\mathbf{Q}). \qquad (10)$$

A. Number of iterations

Table II shows numbers of iterations (external iterations, that is restarts) used to achieve a given accuracy for selected parameters m for four methods: GMRES(m) alone (denoted GMRES(m)) and GMRES(m) preconditioned with IWZ(0) (denoted IWZ(0)GMRES and (m)IWZ(0)G–(m)), ILU(0) (denoted ILU(0)GMRES(m) and ILU(0)G–(m)) and ILU3 (denoted ILU3GMRES(m) and ILU3G–(m)).

For a deeper analysis of the influence of the preconditioners on the method, we consider some more measures.

Let $I(M, \mathbf{A}, \varepsilon)$ denote a number of restarts (that is external iterations) needed to achieve the 2-norm of the residual less than ε for a given matrix $\mathbf{A}$ with a given method M. In other words, $I(M, \mathbf{A}, \varepsilon)$ is a minimal k for which defined in (9) $\varepsilon^{(k)}(\mathbf{A}) < \varepsilon$.

Let us define $p(PM, \mathbf{A}, \varepsilon)$, which shows the relationship between the number of iterations needed to achieve a given accuracy ε with a method M with no preconditioner and the same method M with a preconditioner P — both for the same matrix $\mathbf{A}$. We call it *iteration speed-up* and define as follows:

$$p(PM, \mathbf{A}, \varepsilon) = \frac{I(M, \mathbf{A}, \varepsilon)}{I(PM, \mathbf{A}, \varepsilon)}. \qquad (11)$$

Now, let $Is(M, \mathrm{Z}, \varepsilon)$ be an average number of restarts needed to achieve a given accuracy for matrices from a set Z:

$$Is(M, \mathrm{Z}, \varepsilon) = \operatorname*{avg}_{\mathbf{A}\in\mathrm{Z}} I(M, \mathbf{A}, \varepsilon). \qquad (12)$$

Next, we define $ps(PM, \mathrm{Z}, \varepsilon)$ which shows the relationship between the average number of iterations needed to achieve a given accuracy ε with a method M with no preconditioner and the same method M with a preconditioner P — for a set Z of matrices.

$$ps(PM, \mathrm{Z}, \varepsilon) = \frac{Is(M, \mathrm{Z}, \varepsilon)}{Is(PM, \mathrm{Z}, \varepsilon)}. \qquad (13)$$

At last, we are going to define some more characteristics with p defined above (11). They will be the maximal and minimal p for a given matrix $\mathbf{A}$ solved with a preconditioned GMRES(m) (for $m \in \{1, \ldots, m_0\}$; with a preconditioner P) as well as m giving that p:

$$\begin{aligned} p_{max}(P\mathrm{GMRES}(m), m_0, \mathbf{A}, \varepsilon) &= \\ = \max_{1\leq i\leq m_0} p(P\mathrm{GMRES}(i), \mathbf{A}, \varepsilon), \end{aligned} \qquad (14)$$

$$\begin{aligned} p_{min}(P\mathrm{GMRES}(m), m_0, \mathbf{A}, \varepsilon) &= \\ = \min_{1\leq i\leq m_0} p(P\mathrm{GMRES}(i), \mathbf{A}, \varepsilon), \end{aligned} \qquad (15)$$

$$\begin{aligned} m_{max}(P\mathrm{GMRES}(m), m_0, \mathbf{A}, \varepsilon) &= \\ = \operatorname*{arg\,max}_{1\leq i\leq m_0} p(P\mathrm{GMRES}(i), \mathbf{A}, \varepsilon), \end{aligned} \qquad (16)$$

$$\begin{aligned} m_{min}(P\mathrm{GMRES}(m), m_0, \mathbf{A}, \varepsilon) &= \\ = \operatorname*{arg\,min}_{1\leq i\leq m_0} p(P\mathrm{GMRES}(i), \mathbf{A}, \varepsilon). \end{aligned} \qquad (17)$$

Whenever we omit the parameter ε (as in $p(\mathrm{ILU(0)GMRES}(m), \mathbf{Q})$ where we mean iteration speed-up of the GMRES method restarted after m inner iterations preconditioned with the ILU(0) factorization — and similar), we assume $\varepsilon = 10^{-16}$.

Fig. 4 shows $ps(P\mathrm{GMRES}(m), \mathrm{A})$ and $ps(P\mathrm{GMRES}(m), \mathrm{B})$ — that is the relationship between the parameter m and the average number of restarts (external iterations) needed to achieve the given accuracy for P being preconditioners: ILU(0), ILU3, IWZ(0). The average number of iterations is counted for two groups of matrices: Group A with $d > 8$ (the matrices 1, 3, 5) and Group B with $d \leq 8$ (the matrices 2, 4, 6).

Some conclusions that can be drawn from Table II and Fig. 4 are following.

- With the increase of the parameter m the average number of iterations needed to achieve the assumed accuracy of the method GMRES(m), IWZ(0)GMRES(m), ILU3GMRES(m) and ILU(0)GMRES(m) decreases.
- Regardless of the size of the matrix, the number of iterations needed to achieve a given accuracy is practically the same and depends on the value of $d = nz/n$ (see Group A versus Group B).
- In the methods IWZ(0)GMRES(m), ILU(0)GMRES(m) and ILU3GMRES(m) the number of outer iterations (that is, restarts) needed to achieve the given convergence is

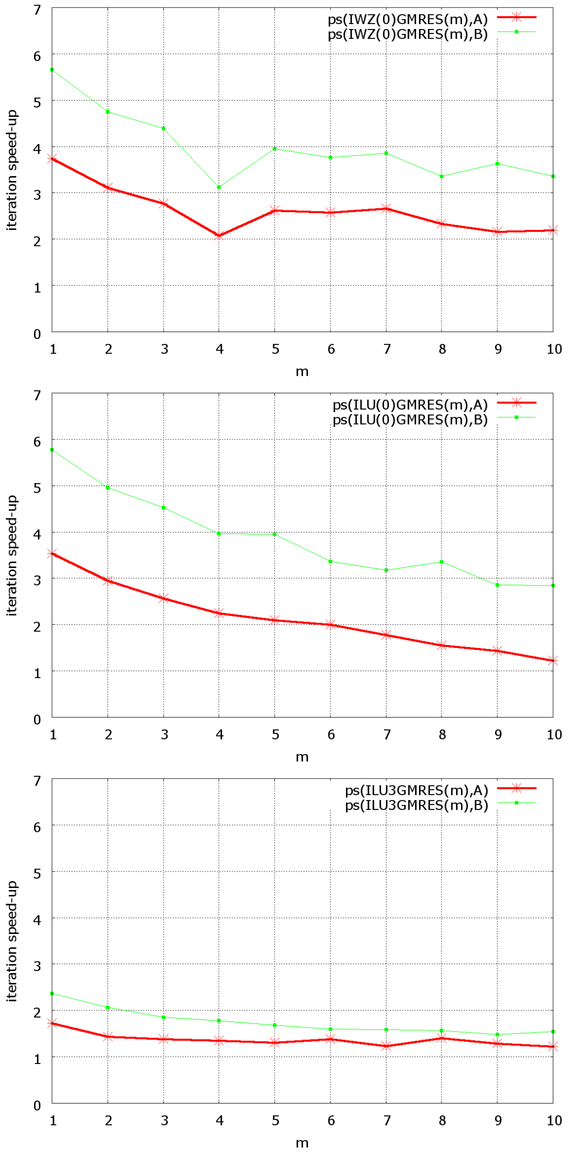

Fig. 4. The relationship between the parameter m and the average number of restarts (external iterations) needed to achieve the accuracy 10^{-16} for the matrices of Group A ($d > 8$) and Group B ($d <= 8$) for Case I with preconditioners IWZ(0) (top); ILU(0) (middle); ILU3 (bottom)

less than in the GMRES(m) method, regardless of the parameter m and the parameter d.

- The value of the iteration speed-up (p) for the matrices of Group A is not less than for those of Group B.
- For the matrices of Group B, the growth of $p(\text{ILU}(0)\text{GMRES}(m), \mathbf{Q})$ and $p(\text{IWZ}(0)\text{GMRES}(m), \mathbf{Q})$ is more uniform than for the matrix of Group A.

Table III shows the minimal and maximal values of p (that is p_{min} and p_{max} from (14) and (15)) and the values of the parameter m for which those values of p are reached (m_{min} and m_{max} from (16) and (17)) — in Case I.

Table IV provides values $p(\text{ILU}(0)\text{GMRES}(m), \mathbf{Q})$, $p(\text{IWZ}(0)\text{GMRES}(m), \mathbf{Q})$, $p(\text{ILU3GMRES}(m), \mathbf{Q})$ for the matrix from Case II. In that table we omit $\mathbf{Q}$ because there is only one such a matrix.

TABLE III
MAXIMAL AND MINIMAL VALUES OF THE CHARACTERISTICS p AND THE RESPECTIVE VALUES OF m FOR MATRICES OF CASE I

precond.	value	matrix ID					
		1	2	3	4	5	6
IWZ(0)	p_{max}	3.21	3.00	4.00	5.61	4.20	**7.97**
	m_{max}	9	8	1	2	1	1
	p_{min}	2.00	2.50	2.00	3.50	2.00	3.75
	m_{min}	8	2	7	10	6	9
ILU(0)	p_{max}	3.21	3.11	3.67	5.69	3.82	**8.21**
	m_{max}	1	1	1	1	1	1
	p_{min}	1.33	2.00	1.33	2.80	1.00	3.20
	m_{min}	10	10	10	10	10	9
ILU3	p_{max}	1.45	1.32	1.83	2.33	2.00	**3.23**
	m_{max}	1	1	1	2	1	1
	p_{min}	1.18	1.11	1.25	1.67	1.00	1.60
	m_{min}	3	8	7	9	10	9

TABLE IV
VALUES OF $p(\text{ILU}(0)\text{GMRES}(m), \mathbf{Q})$, $p(\text{IWZ}(0)\text{GMRES}(m), \mathbf{Q})$, $p(\text{ILU3GMRES}(m), \mathbf{Q})$ FOR THE MATRIX $\mathbf{Q}$ OF CASE II FOR DIFFERENT VALUES m

m	p(IWZ(0)G–(m))	p(ILU(0)G–(m))	p(ILU3G–(m))
5	**5.07**	**7.89**	**0.45**
14	5.00	5.00	0.25
25	4.50	3.00	0.16
29	3.50	2.33	0.15
33	3.00	2.00	0.19
41	2.00	1.33	0.17
49	1.50	1.00	0.14
61	1.50	1.50	0.16

From Tables III and IV it can be concluded the following.

- With the growth of parameter m (where m changes from 1 to 10) the value of p for IWZ(0) and for ILU(0) decreases in both cases.
- IWZ(0) and ILU(0) improve (that is: decreases) the number of restarts for both cases — even 8 times.
- The ILU3 preconditioner in Case I always reduces the number of restarts. However, in Case II it always spoils the performance.
- IWZ(0) and ILU(0) behave similarly for both cases.

B. Convergence rate of GMRES(m)

1) Case I: Fig. 5 presents the relationship between the number of iterations and the achieved accuracy $acc(\mathbf{Q}, i)$ (see (10)) for the matrices with ID 3 and 4 (from Group A and B, respectively) for the methods GMRES(m), IWZ(0)GMRES(m), ILU(0)GMRES(m) and ILU3GMRES(m) for two selected values of parameter m. The values of m were chosen after the analysis of Table III and Fig. 4. The maximal iteration speed-up is for $m = 1$ (regardless of the preconditioner) and the minimal iteration speed-up is usually for $m = 10$.

Fig. 5 shows that the higher value of the parameter m, the more rapidly convergent is the method GMRES(m). Analogously, the higher value of the parameter m means that IWZ(0)GMRES(m), ILU(0)GMRES(m) and

ILU3GMRES(m) methods are faster convergent regardless of the matrix.

The convergence curve $acc(\mathbf{Q}, i)$ as a function of i is almost of the same shape for any particular parameter m for the GMRES(m) method and the IWZ(0)GMRES(m), ILU(0)GMRES(m) and ILU3GMRES(m) methods — only for the preconditioned GMRES(m) methods the curve is shifted upwards. It means that these methods are faster convergent than the GMRES(m) method.

The best preconditioner for GMRES(m), regardless of the parameter m and of the properties of the matrix, is IWZ(0).

For the the matrix of ID 3 from Group A and $m = 10$ (Fig. 5, higher middle) all the investigated preconditioners give similar results.

2) Case II: Fig. 6 shows the relationship between the number of iterations and the accuracy $acc(\mathbf{Q}, i)$ (10) for the GMRES(m) method and the IWZ(0)GMRES(m), ILU(0)GMRES(m), ILU3GMRES(m) methods for $m = 5$ and $m = 49$. The values of m was based on Table IV — the maximal iteration speed-up (regardless of the preconditioner) is usually for $m = 5$ and minimal — for $m = 49$.

Fig. 6 and Table IV show the following conclusions.

- The preconditioner ILU3 is completely not fitted for Case II, worsening the accuracy of the method GMRES(m). As we noticed in the end of Section III-B, ILU3 reduces to the Jacobi preconditioner for such banded matrices as in Case II and it causes such a poor accuracy, because the Jacobi preconditioner is rather little effective.
- The best preconditioner is ILU(0). However, as m grows, the iteration speed-up declines.

For the matrix of Case II it is not always the case, that IWZ(0) improves the convergence rate, because that matrix has a banded structure. ILU(0), being diagonally arranged, seems to be more consistent with the matrix's structure than IWZ(0) — the structure of the latter "crosses" the structure of the original matrix and thus spoils it.

V. Conclusion

Those numerical experiments helped us understand the effect of incomplete preconditioners on the convergence of preconditioned Krylov subspace methods — like GMRES(m).

Using IWZ(0) and ILU(0) improves the convergence of the GMRES(m) method (that is, decreases the number of the iterations needed to achieve the required accuracy — even 8 times) used to solve sparse linear equation systems connected to Markov chains.

Both preconditioners — IWZ(0) and ILU(0) — work similarly in terms of the iteration speed-up for the described method and matrices. However, ILU3 appears to be completely useless in the presented cases.

The rate of the convergence of the projection method GMRES(m) as well as the preconditioned one does not depend on the size of the matrix $\mathbf{Q}$.

The speed of convergence in terms of numbers of iterations (restarts) of GMRES(m) depends on the structure of the matrix. There were tested matrices for two different cases and

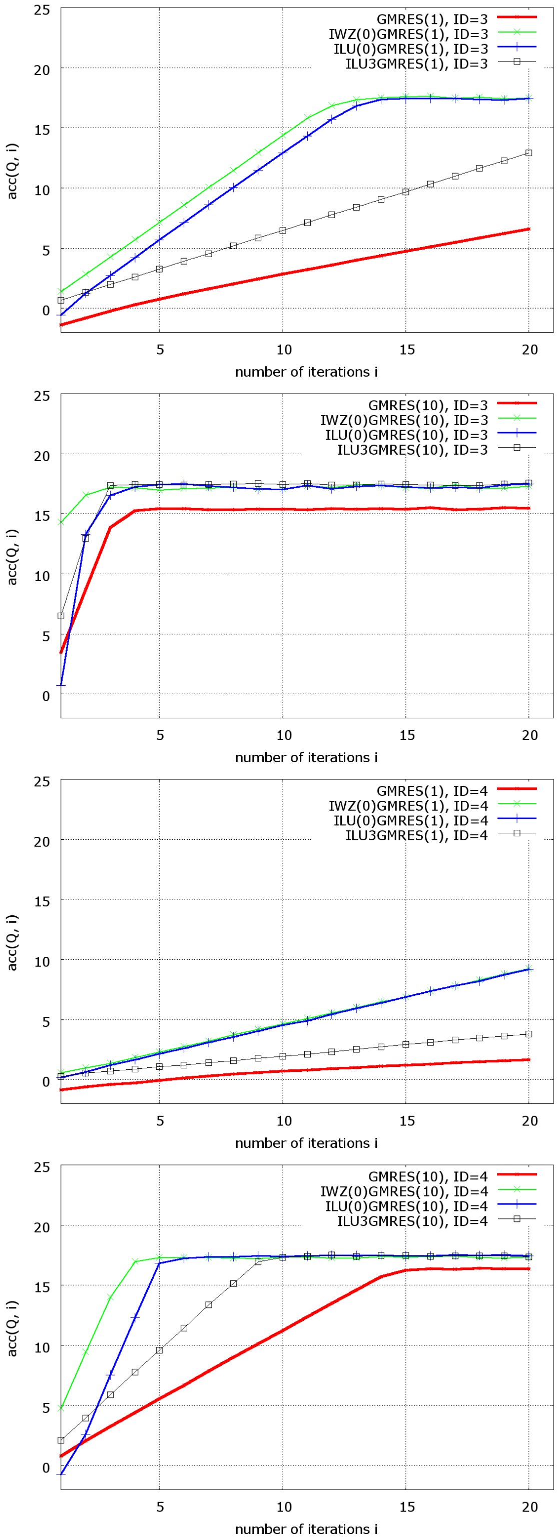

Fig. 5. The plot of the achieved accuracy $acc(\mathbf{Q}, i)$ as a function of i for various matrices and values of m: matrix $ID = 3$ of Group A, $m = 1$ (top); matrix $ID = 3$ of Group A, $m = 10$ (higher middle); matrix $ID = 4$ of Group B, $m = 1$ (lower middle); matrix $ID = 4$ of Group B, $m = 10$ (bottom)

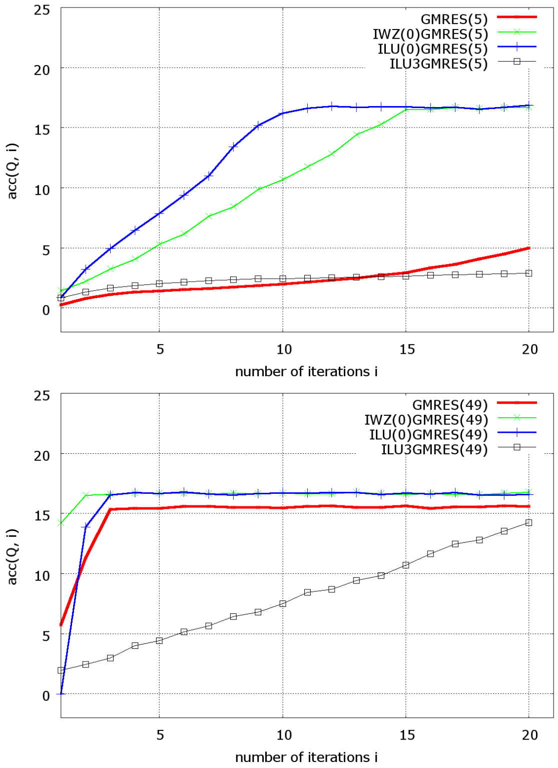

Fig. 6. The plot of the achieved accuracy $acc(\mathbf{Q}, i)$ as a function of i for the matrix of Case II for various values of the parameter m: $m = 5$ (top); $m = 49$ (bottom)

they were characterized by the fact that the matrices from case I had no structure, the matrix from case II had a really clear structure.

The matrices of Case I were further distinguished by their density, given by $d = nz/n$. And the set of the matrices which had a low value of the parameter d ($d < 8$) gave slower convergence and they require additional techniques to improve the rate of the convergence. This technique could be some more complicated preconditioning.

References

[1] M. Benzi, B. Ucar: Block Triangular preconditioners for M-matrices and Markov chains. ETNA (Electronic Transactions on Numerical Analysis), Vol. 26, pp.209–227, 2007.

[2] M. Benzi, B. Ucar: Product preconditioning for Markov chain problems, Proceedings of the 2006 Markov Anniversary Meeting (Charleston, SC, 12-14 June 2006), Boson Books, Raleigh, NC, 2006, pp. 239–256.

[3] N. I. Buleev: A numerical method for solving two-dimensional diffusion equations. At. Energ. 6 (1959), p. 338. [In Russian.]

[4] N. I. Buleev: A numerical method for solving two- and three-dimensional diffusion equations. Mat. Sb. 51 (1960), p. 227. [In Russian.]

[5] B. Bylina, J. Bylina: Incomplete WZ decomposition algorithm for solving Markov chains, *Journal of Applied Mathematics*, vol. 1 (2008), n. 2, p. 147–156.

[6] B. Bylina, J. Bylina: The experimental analysis of GMRES convergence for solution of Markov chains, *Proceedings of the International Multiconference on Computer Science and Information Technology*; October 18–20, 2010, Wisï£¡a, Poland, 5 (2010), ISSN 1896-7094, ISBN 978-83-60810-27-9, IEEE Catalog Number CFP1064E-CDR, pp. 281–288.

[7] T. Dayar, W. J. Stewart.: Comparison of partitioning techniques for two-level iterative solvers on Large, Sparse Markov chins, *SIAM Journal on Scientific Computing* 21, p. 1691 (2000)

[8] C. G. J. Jacobi: Über eine neue Aufiösungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen. Aston. Nachrichten 22 (1845), p. 297.

[9] D. S. Kershaw: The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations. J. Comput. Phys. 26 (1978), p. 43.

[10] J. A. Meijerink, H. A. van der Vorst: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. Comput. 31 (1977), p. 148.

[11] P. K. Pollett, D. E. Stewart: An Efficient Procedure for Computing Quasi-Stationary Distributions of Markov Chains with Sparse Transition Structure, *Advances in Applied Probability* 26 (1994), p. 68.

[12] C. J. Ridler-Rowe: On a Stochastic Model of an Epidemic, *Advances in Applied Probability*, vol. 4, 1967, p. 19–33.

[13] Y. Saad, M. H. Schultz: GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM Journal of Scientific and Statistical Computing*, 7, 1986, p. 856–869.

[14] W. Stewart: *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Chichester, West Sussex 1994.

[15] R. S.Varga: Factorizations and normalized iterative methods. In: Boundary Problems in Differential Equations (ed. by R. E. Langer), Univ. Wisconsin Press, Madison (1960).

[16] P. Yalamov, D. J. Evans: The WZ matrix factorization method, *Parallel Computing* 21 (1995), p. 1111.