Bull. Nov. Comp. Center, Num. Anal., 2 (1993), 19–35

On iterative solving of linear algebraic equations in p-h-versions of finite element methods

V.P. Il'in and S.S. Radionov

The goal of this paper is investigation of the efficiency of iterative preconditioned cojugate gradient methods for solving linear systems of equations which arise in p-h-version of finite element methods. The results of numerical experiments are presented for the model boundary value problem with different values of h, p and iterative parameters. Some conclusions on the comparative costs of algorithms are made on the base of analysis of matrix structure of linear systems.

1. Introduction

We consider an efficiency of modern iterative algorithms in application for sparse linear systems of equations in p-h-versions of finite element methods [1]. The aim is to compare the numerical costs for different values of p, h and some variants of iterative methods for different input data. For simplicity the Poisson equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x,y) \tag{1.1}$$

is solved in a square computational domain with the Dirichlet or Neumann conditions on the different parts of boundary.

In Section 2 the structures of local and global stiffness matrices are described for the square elements. Section 3 includes the formulas of symmetric successive over relaxation method in the efficient Eisenstat implementation with conjugate gradient acceleration, see [2,3,4]. The discussion on the estimates of computer costs is presented in comparison with the direct Cholesky factorization algorithm. In the last section we present and discuss the results of numerical experiments to analyse separately the influence of preconditioning, iterative parameters, neccessary exactness and characteristics of boundary value problems.

2. The matrices of p-h-finite element methods

For standard square element in a local coordinates

$$R = \{-1 \le x \le 1, \quad -1 \le y \le 1\}$$

there are used three types of basic functions with the total number $N_t = (p+1)^2$:

4 nodal functions

$$N_{k,l} = \frac{1+kx}{2} \cdot \frac{1+ly}{2}, \quad k,l = \pm 1; \tag{2.1}$$

$4(p-1)$ side functions

$$\begin{aligned} S_{k,l}^{(m)} &= Q_m(x\cdot \mid l \mid + y \cdot \mid k \mid) \frac{1+kx}{2} \cdot \frac{1+ly}{2}, \quad \mid k \mid + \mid l \mid = 1, \\ Q_m(z) &= \sqrt{\frac{2m+1}{2}} \int\limits_{-1}^{z} P_m(z')dz', \\ P_m(z) &= \frac{1}{2^m \cdot m!} \frac{d^m}{dz^m}(z^2-1)^m, \\ & m = 1, 2, \ldots, p-1, \end{aligned} \tag{2.2}$$

$(p-1)^2$ internal functions

$$I_{k,l} = Q_k(x) \cdot Q_l(y), \quad k,l = 1, \ldots, p-1. \tag{2.3}$$

We unify the basic functions into set $\varphi = \{\varphi_q(x,y)\}$ with the ordering

$$\varphi = \{N, S, I\},$$

where N, S and I mean the subsets of nodal, side and internal functions. In more detail

$$\begin{aligned} N &= \{N_{1,1}, N_{-1,1}, N_{1,-1}, N_{-1,-1}\}, \\ S &= \{S_{1,0}^{(1)}, \ldots, S_{1,0}^{(p-1)}, S_{-1,0}^{(1)}, \ldots, S_{-1,0}^{(p-1)}, \\ & \quad S_{0,1}^{(1)}, \ldots, S_{0,1}^{(p-1)}, S_{0,-1}^{(1)}, \ldots, S_{0,-1}^{(p-1)}\}, \\ I &= \{I_{1,1}, \ldots, I_{1,p-1}, I_{2,1}, \ldots, I_{2,p-1}, \ldots, I_{p-1,1}, \ldots, I_{p-1,p-1}\}. \end{aligned}$$

It is natural to connect the different basic functions with geometric object of element. We will consider the nodal functions corresponded to the mesh points, the groups $S_{1,0}^{(k)}$, $S_{-1,0}^{(k)}$, $S_{0,1}^{(k)}$, $S_{0,-1}^{(k)}$, $k = 1, \dots, p-1$, of side function corresponded to the right, left, upper and low element sides, and the groups $I_{k,q}$, $q = 1, \dots, p-1$; $I_{q,l}, q = 1, \dots, p-1$ corresponded to k-th vertical or l-th horizontal line inside the element.

Local stiffnes matrix is defined as

$$a = \{a_{q,q'}\}, \quad q, q' = 1, 2, \dots, (p+1)^2.$$

Here $a_{q,q'}$ is scalar product

$$a_{q,q'} = (\varphi_q, \varphi_{q'}) = \int\limits_R \left(\frac{\partial \varphi_q}{\partial x} \cdot \frac{\partial \varphi_{q'}}{\partial x} + \frac{\partial \varphi_q}{\partial y} \cdot \frac{\partial \varphi_{q'}}{\partial y} \right) dxdy. \qquad (2.4)$$

The block structure of matrix A is of the following type:

$$A = \begin{bmatrix} NN & NS & NI \\ SN & SS & SI \\ IN & IS & II \end{bmatrix}.$$

Here NN, SS and II denote the square submatrices of the orders 4, $4(p-1)$ and $(p-1)^2$ which correspond to scalar products of nodal, side and internal basic functions. The blocks $NS = (SN)^t$, $SI = (IS)^t$ and $NI = (IN)^t = 0$ are rectangular submatrices.

In particular, the matrix

$$NN = \begin{bmatrix} \frac{2}{3} & -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{6} & \frac{2}{3} & -\frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{6} \\ -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} & \frac{2}{3} \end{bmatrix}$$

has 16 nonzero entries. The matrix SS is block diagonal one with the block of order 2:

$$SS = \begin{bmatrix} S_1 & S_2 & 0 & 0 \\ S_2 & S_1 & 0 & 0 \\ 0 & 0 & S_1 & S_2 \\ 0 & 0 & S_2 & S_1 \end{bmatrix}.$$

One diagonal block corresponds to the vertical mesh intervals and the other one to horisontal. The subblocks S_1 correspond to connections between the basic functions from the common sides, and S_2 – from the opposite.

These subblocks have the same tridiagonal structures (between each pair of nonzero diagonals there is one zero diagonal):

$$S_k = \begin{bmatrix} + & 0 & - & \ddots & \ddots \\ 0 & + & \ddots & - & \ddots \\ - & \ddots & + & \ddots & - \\ \ddots & - & \ddots & + & 0 \\ \ddots & \ddots & - & 0 & + \end{bmatrix}.$$

So, the matrix SS has $8(3p-7)$ nonzero elements. Of course, the above consideration is valid for $p > 3$. If $p = 2$, the matries S_k are scalar values in fact, and for $p = 3$, S_k are diagonal matries.

The matrix (of block order $p-1$)

$$II = \begin{bmatrix} B & 0 & -C & 0 & . & . & . \\ 0 & B & 0 & \ddots & \ddots & 0 & . \\ -C & 0 & \ddots & \ddots & \ddots & . & . \\ 0 & \ddots & \ddots & \ddots & \ddots & . & 0 \\ . & \ddots & \ddots & \ddots & \ddots & 0 & -C \\ . & 0 & \ddots & \ddots & 0 & B & 0 \\ . & . & . & 0 & -C & 0 & B \end{bmatrix}$$

includes diagonal positive definite submatrix C, diagonal square blocks of order $p-1$ with three nonzero diagonals (as S_k)

$$B = \begin{bmatrix} + & 0 & - & & & \\ 0 & \ddots & \ddots & \ddots & 0 & \\ - & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & - \\ & 0 & \ddots & \ddots & \ddots & 0 \\ & & & - & 0 & + \end{bmatrix}$$

and the total nonzero entries 1 for $p = 2$ and $(p-1)(5p-13)$ for $p \geq 3$. It is useful to remark that if side and internal functions at each element side or "inner line" will be numbered in "red-black" ordering (firstly – all odd functions and secondly – even), then S_1, S_2 and B become block diagonal matrices of block order 2, and each subblocks will be "usual" tridiagonal matrix. Nondiagonal block NS has the form

$$NS = [NS_1,\ NS_2,\ NS_3,\ NS_4]$$

$$NS_1 = \begin{bmatrix} - & - & 0 & \cdots & 0 \\ + & + & 0 & \cdots & 0 \\ - & + & 0 & \cdots & 0 \\ + & - & 0 & \cdots & 0 \end{bmatrix}, \quad NS_2 = \begin{bmatrix} + & + & 0 & \cdots & 0 \\ - & - & 0 & \cdots & 0 \\ + & - & 0 & \cdots & 0 \\ - & + & 0 & \cdots & 0 \end{bmatrix},$$

$$NS_3 = \begin{bmatrix} - & - & 0 & \cdots & 0 \\ - & + & 0 & \cdots & 0 \\ + & + & 0 & \cdots & 0 \\ + & - & 0 & \cdots & 0 \end{bmatrix}, \quad NS_4 = \begin{bmatrix} + & + & 0 & \cdots & 0 \\ + & - & 0 & \cdots & 0 \\ - & - & 0 & \cdots & 0 \\ - & + & 0 & \cdots & 0 \end{bmatrix}.$$

Here the numbers of total and nonzero entries are $16(p-1)$ and (32).The blocks NS_1, NS_2, NS_3, NS_4 have $p-1$ columns and are corresponded to the sets of basic functions $S^m_{1,0}$, $S^m_{-1,0}$, $S^m_{0,1}$, $S^m_{0,-1}$. Above "-" and "+" denote the signs of matrix nonzero entries.

The matrix SI is rectangular one and has $4 \times (p-1)$ block structure

$$SI = \begin{bmatrix} S^{(1)}_{1,0} & S^{(2)}_{1,0} & \cdots & S^{(p-1)}_{1,0} \\ S^{(1)}_{-1,0} & S^{(2)}_{-1,0} & \cdots & S^{(p-1)}_{-1,0} \\ S^{(1)}_{0,1} & S^{(2)}_{0,1} & \cdots & S^{(p-1)}_{0,1} \\ S^{(1)}_{0,-1} & S^{(2)}_{0,-1} & \cdots & S^{(p-1)}_{0,-1} \end{bmatrix}$$

with the square blocks of the equal same order $p-1$. The blocks $S^{(m)}_{1,0}$ and $S^{(m)}_{-1,0}$ are diagonal matrices for $m = 1, 2$ and zero – for others m. The blocks $S^{(m)}_{0,1}$ and $S^{(m)}_{0,1}$ have nonzero entries in the two first columns at the row with number m only. So, the matrix SI has the total $4(p-1)^3$ entries and $8(p-1)$ nonzero elements. In more detail, the entries of matrix block SI are defined by the scalar products of basic functions:

$$(S^{(m)}_{1,0}, I_{k,l}) = \begin{cases} -1/\sqrt{6}, & \text{for } l = 1,\ m = k, \\ -1/3\sqrt{10}, & \text{for } l = 2,\ m = k, \\ 0, & \text{otherwise}, \end{cases}$$

$$(S^{(m)}_{-1,0}, I_{k,l}) = \begin{cases} -1/\sqrt{6}, & \text{for } l = 1,\ m = k, \\ -1/3\sqrt{10}, & \text{for } l = 2,\ m = k. \\ 0, & \text{otherwise}, \end{cases}$$

$$(S_{0,1}^{(m)}, I_{k,l}) = \begin{cases} -1/\sqrt{6}, & \text{for } l = 1,\ m = k, \\ -1/3\sqrt{10}, & \text{for } l = 2,\ m = k, \\ 0, & \text{otherwise,} \end{cases}$$

$$(S_{0,-1}^{(m)}, I_{k,l}) = \begin{cases} -1/\sqrt{6}, & \text{for } l = 1,\ m = k, \\ -1/\sqrt{3}, & \text{for } l = 2,\ m = k, \\ 0, & \text{otherwise.} \end{cases}$$

In the block NS there are only the following nonzero entries:

$$\begin{aligned}(N_{1,-1}, S_{0,-1}^{(m)}) &= (N_{1,1}, S_{1,0}^{(m)}) \\ &= (N_{1,1}, S_{-1,0}^{(m)}) = (N_{1,1}, S_{0,1}^{(m)}) = \begin{cases} -1/2\sqrt{6}, & m = 1, \\ -1/6\sqrt{10}, & m = 2, \end{cases}\end{aligned}$$

$$\begin{aligned}(N_{1,1}, S_{1,0}^{(m)}) &= (N_{1,1}, S_{0,1}^{(m)}) \\ &= (N_{1,1}, S_{0,-1}^{(m)}) = (N_{-1,1}, S_{1,0}^{(m)}) = \begin{cases} 1/2\sqrt{6}, & m = 1, \\ 1/6\sqrt{10}, & m = 2, \end{cases}\end{aligned}$$

$$\begin{aligned}(N_{1,1}, S_{1,0}^{(m)}) &= (N_{-1,-1}, S_{-1,0}^{(m)}) \\ &= (N_{1,1}, S_{0,1}^{(m)}) = (N_{-1,-1}, S_{0,-1}^{(m)}) = \begin{cases} -1/2\sqrt{6}, & m = 1, \\ 1/6\sqrt{10}, & m = 2, \end{cases}\end{aligned}$$

$$\begin{aligned}(N_{-1,-1}, S_{0,1}^{(n)}) &= (N_{-1,1}, S_{0,-1}^{(m)}) \\ &= (N_{1,1}, S_{1,0}^{(m)}) = (N_{-1,-1}, S_{1,0}^{(m)}) = \begin{cases} 1/2\sqrt{6}, & m = 1, \\ -1/6\sqrt{10}, & m = 2. \end{cases}\end{aligned}$$

Nonzero entries of diagonal block SS are constructed by the scalar products of basic functions corresponded to the common sides:

$$(S_{k,l}^{(m)}, S_{k,l}^{(m')}) = \begin{cases} \dfrac{2}{3} + \dfrac{1}{2(2m-1)(2m+3)}, & m' = m, \\ \dfrac{-1}{2\sqrt{(2m+1)(2m+5)(2m+3)}}, & m' = m+2, \end{cases}$$

and corresponded to the opposite sides only:

$$\cdot\ (S_{k,l}^{(m)}, S_{-k,-l}^{(m')}) = \begin{cases} \dfrac{1}{3} - \dfrac{1}{2(2m-1)(2m+3)}, & m' = m, \\ \dfrac{1}{2\sqrt{(2m+1)(2m+5)}(2m+3)}, & m' = m+2. \end{cases}$$

And at last, nonzero entries of block II are:

$$(I_{k,l}, I_{k',l'}) = \frac{2}{(2m-1)(2m+3)}$$

$$+ \begin{cases} \dfrac{2}{(2n-1)(2n+3)}, & \text{for } k = k' = m,\ l = l' = n, \\ 0, & \text{for } k = k' = m,\ l' \neq l,\ l' \neq l+2, \\ & \text{or } \ l = l' = m,\ k' \neq k,\ k' \neq k+2, \\ -\dfrac{1}{\sqrt{(2n+1)(2n+5)(2n+3)}}, & \text{for } k = k' \pm 2 = m,\ l = l' = n \\ & \text{or } \ l = l' + 2 = m,\ k = k' = n. \end{cases}$$

For describing the global stiffness matrix A we have to define the ordering of all basic functions. Let the computational domain be a square divided into $n \times n$ elements by meshlines $x_i = x_0 + ih$, $y_j = y_0 + jk$; $i, j = 0, 1, \ldots, n$, h is some meshsize. So, there are n^2 elements, $(n+1)^2$ nodes, $n(n+1)$ vertical sides (segments between nodes) and $n(n+1)$ horizontal sides. The total number of unknowns and the order of the global system of equations

$$Au = f, \quad A = D - L - U, \tag{2.5}$$

with square matrix A equals to

$$N = (n+1)^2 + 2(p-1)n(n+1) + (p-1)^2 n^2 = (np+1)^2. \tag{2.6}$$

Here $D = \text{diag}(A)$ and L, U are strictly low and upper triangular parts of matrix A.

In (2.5) $u = \{u_k\}$ means the global vector which entries are numbered in the following way:

node entries in the row-by-row ordering –

$$\kappa = i + (j-1)(n+1); \quad i, j = 1, 2, \ldots, n+1,$$

vertical side entries in the similar ordering –

$$\kappa = (n+1)^2 + 1, \ldots, (n+1)^2 + (p-1)(n+1)n,$$

horizontal side entries –

$$\kappa = (n+1)[n+1+(p-1)n]+1,\dots,(n+1)^2+2(p-1)(n+1)n,$$

internal entries in the element-by-element ordering –

$$\kappa = (n+1)[n+1+2(p-1)n]+1,\dots,N.$$

Such matrix A is constructed without taking into account the boundary conditions and is called unconstrained matrix. For obtaining the resulting constrained matrix it is necessary to modify the rows of the matrix A and right side vector f in (2.5), corresponded to the boundary nodes and sides.

If some basic function is connected with the boundary side or node with the Neumann condition, then corresponding row of matrix A doesn't change. In the case of the Dirichlet condition the row becomes with the unit at the main diagonal and zero other entries.

The structure of the global stiffness matrix for the described ordering can be presented as a similar local one:

$$A = \begin{bmatrix} GNN & GNS & 0 \\ GSN & GSS & GSI \\ 0 & GIS & GII \end{bmatrix}.$$

Here blocks are

GNN – square submatrix of order $(n+1)^2$,

GSS – square block of order $2(p-1)n(n+1)$,

GII – square submatrix of order $(p-1)^2n^2$,

$GNS = (GSN)'$ – rectangular matrix with $(n+1)^2$ rows and $2(p-1)n(n+1)$ columns,

$GSI = (GIS)'$ – rectangular matrix of the form $[2(p-1)n(n+1)]\cdot[(p-1)^2n^2]$.

Let us consider the structure of each block now (in an unconstrained form, i.e., without considering of boundary condition). Matrix GNN is nine-diagonal with the semi-bandwidth $d = n+2$. In other words, this matrix is block tridiagonal of block order $n+1$ and each block is tridiagonal one (of order $n+1$). Nonzero entries of GNN equal to 16/3 at the main diagonal and 2/3 at the others. The total number of nonzero entries of GNN equals to

$$Q_{NN} = (3n+1)^2.$$

Submatrix GSS is block-diagonal of block order 2 (first block is corresponded to the vertical sides and the second – to horizontal ones. Each diagonal block GSS_k, $k = 1, 2$, is block-diagonal matrix of block order n with the equal blocks GSS_0 at the main diagonal ($GSS_k = \text{diag}\{GSS_0\}$), GSS_0 is block-tridiagonal matrix of block order $n+1$ with the subblocks $S_{11} = \text{diag}\{S_1\}$, $S_{22} = \text{diag}\{S_2\}$, which are block-diagonal matries. Here S_1, S_2 are the same as in SS, see above. In other words, GSS is nine-diagonal with semi-bandwidth $d = (p-1)n$:

$$GSS_0 = \begin{array}{|c|c|c|c|}\hline S_{11} & S_{22} & & \\ \hline S_{22} & S_{11} & S_{22} & \\ \hline & S_{22} & S_{11} & S_{22} \\ \hline & & S_{22} & S_{11} \\ \hline \end{array}, \qquad GSS = \begin{array}{|c|c|}\hline GSS_1 & 0 \\ \hline 0 & GSS_2 \\ \hline \end{array}.$$

So, the total number of nonzero entries of GSS equals to

$$Q_{SS} = 2n(3n+1)(3p-7).$$

This equality is valid for $q > 3$. For $p = 2$ and $p = 3$ the values $(3p-7)$ have to be replaced by 1 and 2 correspondingly. The biggest diagonal block GII is block-diagonal matrix of block order n^2 and the order of each block is $(p-1)^2$ (it is exactly the local stiffness matrix II, see above). So the total number of nonzero entries of GII equals to

$$Q_{II} = n^2(p-1)(5p-13)$$

for $p \geq 3$ and $Q_{II} = n^2$ for $p = 2$.

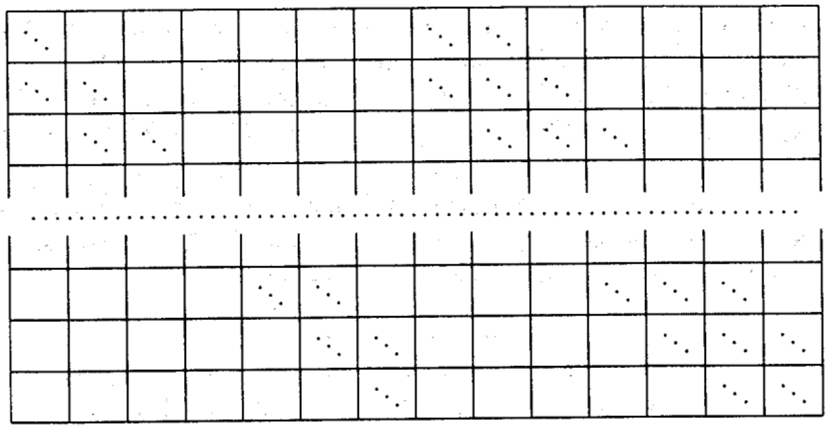
Offdiagonal block GNS has $(n+1)^2$ rows, $2(p-1)n(n+1)$ columns and in general 24 nonzero entries in the each row. Maximum distance from nonzero entry to main diagonal equals to $2n(n+1)(p-1)$ approximately.

In block form GNS can be divided by vertical line into two equal subblocks, the left one is corresponded to the "vertical" side functions and right – to the "horizontal" functions.

The left subblock is presented as two-block-diagonal matrix with $n+1$ block rows and n block columns. Corresponding sub-subblocks are identical rectangular matrices with $n+1$ rows, $(n+1)(p-1)$ columns and 6 nonzero diagonals for $p \geq 3$ (3 – for $p = 2$).

The right subblock is block-tridiagonal matrix with $n+1$ block rows and columns. Each sub-subblock is matrix with $n+1$ rows, $n(p-1)$ columns and 4 nonzero diagonals for $p \geq 3$ (2 – for $p = 2$).

$GNS =$



Block $GSI = (GIS)'$ has $2(p-1)n(n+1)$ rows, $(p-1)^2n^2$ columns and in general 4 nonzero entries in the each row. The biggest distance from nonzero entry to main diagonal equals to $(p-1)^2n^2$ approximately.

In block form GSI is divided by vertical line into two equal subblocks. The left one corresponds to vertical sides and presents the block-diagonal matrix (of block order n) with identical sub-subblocks, consisted of $((n+1)(p-1)$ rows, $n(p-1)^2$ columns and 4 nonzero diagonals. The right subblock indcludes $n+1$ block rows, n block columns and two block diagonals with identical sub-subblocks consisted of $n(p-1)$ columns and nonzero diagonals.

The total number of nonzero entries of GSI equals to

$$Q_{SI} = 8n^2,$$

and of matrix A –

$$Q_A = Q_{NN} + Q_{SS} + Q_{II} + 2Q_{NS} + 2Q_{SI} \approx n^2(12 + 5p^2).$$

The last approximate equality is valid for $p > 3$, and

$$Q_A \approx 57n^2 \,\text{for}\, p = 3, \quad Q_A \approx 48n^2 \,\text{for}\, p = 2.$$

3. An iterative algorithms

We apply the preconditioned conjugate gradient methods in the form

$$\begin{aligned} &r^\circ = f - Au^\circ, \quad p^\circ = B^{-1}r^\circ, \\ &u^{k+1} = u^k + \alpha_k p^k, \quad \alpha_k = (r^k, B^{-1}r^k)/(Ap^k, p^k), \\ &r^{k+1} = r^k - \alpha_k Ap^k, \quad p^{k+1} = B^{-1}r^{k+1} + \beta_k p^k, \\ &\beta_k = (r^{k+1}, B^{-1}r^{k+1})/(r^k, B^{-1}r^k). \end{aligned} \tag{3.1}$$

Here B is symmetric preconditioning matrix

$$B = (G - L)G^{-1}(G - U) \tag{3.2}$$

with some diagonal matrix $G = 1/\omega D$, $0 < \omega < 2$, corresponded to the well-known Symmetric Successive Over Relaxation (SSOR) method.

Now we describe an efficient implementation of iterative process (3.1), based on some modification of Eisenstat's idea [4]. Let us introduce the vectors

$$\begin{aligned} &\bar{u}^k = U_b u^k, \quad \bar{p}^k = U_b p^k, \quad \bar{r}^k = L_b^{-1} r^k, \quad \bar{f} = L_b^{-1} f, \\ &B = L_b U_b, \quad U_b = L_b' = G^{1/2} - G^{-1/2} U \end{aligned} \tag{3.3}$$

and matrix

$$\begin{aligned} \bar{A} &= L_b^{-1} A U_b = (E - I)^{-1} + (E - \bar{U})^{-1} \\ &\quad + (\bar{E} - \bar{L})^{-1}(2E - \bar{D})(E - \bar{U})^{-1}, \\ \bar{L} &= G^{-1/2} L G^{-1/2}, \quad \bar{U} = G^{-1/2} U G^{-1/2}, \\ \bar{D} &= G^{-1/2} D G^{-1/2}. \end{aligned} \tag{3.4}$$

Then from 3.1 for new vectors there is equivalent iterative process

$$\begin{aligned} &\bar{r}^\circ = \bar{f} - \bar{A}\bar{u}^\circ, \quad \bar{p}^\circ = \bar{r}^\circ, \\ &\bar{u}^{k+1} = \bar{u}^k + \alpha_k \bar{p}^k, \quad \bar{r}^{k+1} = \bar{r}^k - \alpha_k \bar{A}\bar{p}^k, \\ &\bar{p}^{k+1} = \bar{r}^{k+1} + \beta_k \bar{p}^k \end{aligned} \tag{3.5}$$

with identity preconditioning matrix $\bar{K} = L_b^{-1} K U_b^{-1} = E$ and the same parameters α_k, β_k.

It is essentual, that multiplication the vector by the matrix $\bar{A}$ is cheap enough operation:

$$\begin{aligned} &\bar{A}\bar{p}^k = (E - \bar{L})^{-1}\bar{t}^k + \bar{q}^k, \\ &\bar{q}^k = (E - \bar{U})^{-1}\bar{p}^k, \quad \bar{t}^k = \bar{p}^k + (2E - \bar{D})\bar{q}^k. \end{aligned} \tag{3.6}$$

For not very sparse matrix A (more than 3 nonzero entries in each row in average) the implementation of formulas (3.9) is twice cheaper approximately, in compare with (3.1). Of course, it is supposed that the matrices $\bar{L}$, $\bar{U}$ and $\bar{D}$ are computed before iterations and resulting vector $u^n = U_b^{-1}\bar{u}^n$ is computed after the iterations.

It is evidently that matrix A is positive definite, but only for $p = 1$ it is of the Stiltjes type (non-positive offdiagonal elements).

An estimates of the convergence velocity of SSOR-iterations for such kind of equations is unknown and our goal is only to make experimental investigations.

4. The results of numerical experiments

The iterations in experiments were performed until the validity of inequality

$$\| r^k \|_2 \,/\, \| r^\circ \|_2 \leq \varepsilon \tag{4.1}$$

for given small enough ε.

Our main criteria of efficiency of algorithms is the number of iterations. The total number of arithmetic operations for solution of system is evaluated approximately by

$$Q = n_\epsilon \cdot (n_c + n_a) \tag{4.2}$$

where n_a is the cost of multiplication $\bar{A} \times \bar{p}$, $n_c = 10N$ is the rest cost of conjugate gradient algorithm per one iteration and n_ϵ is the number of iterations.

As followes from (3.6), the values n_a can be estimated by the approximate equality

$$n_a = 2(Q_A + N),$$

where N, Q_A are the order and the total number of nonzero entries of matrix A. By means of evaluation of Q_A, N for different p we can write

$$n_a = \begin{cases} 2[(3n+1)^2 + (n+1)^2] \approx 20n^2, & p = 1, \\ 104n^2, & p = 2, \\ 132n^2, & p = 3, \\ 2[n^2(12+5p^2) + (1+pn)^2], & p > 3. \end{cases}$$

Calculations were made in double precisions on the IBM PC AT with coprocessor and frequency 12 Mg, in the environment MS-DOS 4.0 with available core ~550 Kbyte which permitted to compute variants for $p \leq 17$ only.

In the following tables there are numerical results for the model problem in the unit square $\Omega = (0,1) \times (0,1)$ with $f(x,y) = 0$ in (1.1) and the Neumann boundary condition $\partial u/\partial n = g(x,y)$ except one angle point of the computational square with the Dirichlet condition. The boundary conditions are defined from the choosen exact solution

$$u_e = Re\Big(\frac{1}{a^2+z^2} + \frac{1}{a^2-z^2}\Big), \quad a = 1.05, \quad z = x + \sqrt{-1}y. \tag{4.3}$$

In each cell of Table 1 there are the numbers of iterations (above) for SSOR-preconditioner, $\omega = 1.5$, $\epsilon = 10^{-5}$, and numbers of unknowns below for different n and p.

Table 1. Numbers of unknowns and iterations for SSORCG, $\epsilon = 10^{-5}$, $\omega = 1.5$

n \ p	1	2	3	4	6	8	9	12	16
1	3 4	8 9	13 16	18 25	25 49	33 81	35 100	49 169	62 289
2	7 9	15 25	19 49	25 81	36 169	44 289	46 361	59 625	74 1089
4	9 25	20 81	21 169	30 289	38 625	46 1089	48 1369	62 2401	76 4225
8	14 81	22 289	23 625	33 1089	42 2401	51 4225	52 5329	– –	– –
16	20 289	26 1089	27 2401	36 4225	– –	– –	– –	– –	– –
31	33 1024	36 3269	– –	– –	– –	– –	– –	– –	– –

Table 2. Time costs of iterative solution (SSORCG, $\epsilon = 10^{-5}$, $\omega = 1.5$) and computing of matrices

n \ p	1	2	3	4	6	8	9	12	16
1	2.3 0.4	2.9 0.6	3.8 0.9	4.7 1.3	7.8 2.4	12.9 4.4	16.8 5.4	36.3 12.0	87.3 24.2
2	3.3 0.6	4.6 1.1	6.2 2.3	7.8 4.2	12.9 10.0	20.8 19.3	25.8 24.3	50.6 51.1	115.1 107.2
4	5.6 0.8	8.5 3.8	12.3 7.8	15.6 16.3	25.6 38.3	40.0 74.9	49.7 95.9	94.0 206.2	205.0 427.9
8	11.3 2.6	22.3 13.9	30.6 30.7	40.4 66.2	66.1 161.9	105.0 321.9	129.7 404.6	– –	– –
16	36.4 10.7	117.4 61.4	151.9 137.0	193.0 280.5	– –	– –	– –	– –	– –
31	255.5 59.2	1172.6 304.7	– –	– –	– –	– –	– –	– –	– –

Table 2 includes the time costs of iterative solution and computing of the global stiffness matrices (in seconds) for the same experiments.

For analysing of the influence of values ε in (4.1) on the numbers of iterations and the accuracy of iterative solution u^k we show in Table 3 the values of k_ϵ and the errors

$$\delta = \frac{\| u_{ex} \| - \| u_{fe} \|}{\| u_{ex} \|} \tag{4.4}$$

for the similar experiments (SSORCG, $\omega = 1.5$) with $\epsilon = 10^{-3}$, 10^{-4}, 10^{-5} and $p = 2$ only (u_{ex} and u_{fe} are exact and finite element solution, $\| u \| = (Au, u))^{1/2}$).

Table 3. The numbers of iterations and errors of iterative solutions ($p = 2$, $\omega = 1.5$)

$\epsilon \setminus n$	1	2	4	8	16	31
10^{-3}	7 $3.9 \cdot 10^{-1}$	11 $2.4 \cdot 10^{-1}$	13 $1.1 \cdot 10^{-1}$	15 $4.4 \cdot 10^{-2}$	19 $1.1 \cdot 10^{-2}$	29 $8.9 \cdot 10^{-4}$
10^{-4}	8 $3.9 \cdot 10^{-1}$	13 $2.4 \cdot 10^{-1}$	17 $1.1 \cdot 10^{-1}$	18 $4.4 \cdot 10^{-2}$	22 $1.1 \cdot 10^{-2}$	33 $8.9 \cdot 10^{-4}$
10^{-5}	8 $3.9 \cdot 10^{-1}$	15 $2.4 \cdot 10^{-1}$	20 $1.1 \cdot 10^{-1}$	22 $4.4 \cdot 10^{-2}$	26 $1.1 \cdot 10^{-2}$	36 $8.9 \cdot 10^{-4}$

Let us make one more observation – what values of p and n are optimal for necessary accuracy Δ. In Table 4 there are for different p the values of the time t_o of SSORCG iterations (in seconds, $\omega = 1.5$), numbers of elements n_o (in one direction) and numbers of corresponding iterations under condition

$$n_o = \min\{n : \delta \leq \Delta\}.$$

In every cell the upper value corresponds to $\Delta = 10^{-3}$ and low to $\Delta = 10^{-4}$.

Table 4. The values of iteration time costs and errors for different p and $n_o(p, \Delta)$

$p \setminus n_0$	3	4	6	8	12	14	15	16
t_0	117 259	83 181	60 125	43 120	51 113	17 77	20 89	24 107
n_0	15 21	9 13	5 7	3 5	2 3	1 2	1 2	1 2
δ	$8 \cdot 10^{-4}$ $8 \cdot 10^{-5}$	$8 \cdot 10^{-4}$ $6 \cdot 10^{-5}$	$4 \cdot 10^{-4}$ $7 \cdot 10^{-5}$	$2 \cdot 10^{-4}$ $5 \cdot 10^{-5}$	$6 \cdot 10^{-5}$ $1 \cdot 10^{-5}$	$3 \cdot 10^{-4}$ $3 \cdot 10^{-5}$	$2 \cdot 10^{-4}$ $1 \cdot 10^{-5}$	 $4 \cdot 10^{-6}$

For understanding the sources of efficiency of the preconditioned conjugate gradient method it is usefull to obtain separately the influence of preconditioning and conjugate gradient acceleration. In Table 5 we give the numbers of iterations ($\epsilon = 10^{-5}$ as in Table 1) for SSOR-steepest descent (SSORSD) method ($\omega = 1.5$, $\beta_k \equiv 0$ in (3.1) or (3.8)) and for "explicit" conjugate gradient iterative process (in (3.1) $B \equiv E$ is identity matrix).

It is important to remark that for large enough n, p the numbers of arithmetical operations per one iteration are approximately the same for each method – SSORCG, SSORSD and explicit CG.

Table 5. The numbers of iterations for SSORSD (above) and explicit CG (below)

n \ p	2	3	4	6	8	9	12	16
1	41 7	39 13	99 20	177 37	275 53	275 60	489 88	727 120
2	49 20	63 28	107 41	183 58	263 78	265 79	445 111	655 133
4	117 33	139 37	165 57	219 74	291 91	313 88	479 125	699 152
8	349 45	393 49	427 64	491 84	555 102	583 101	– –	– –
16	1151 72	1259 75	1351 90	– -	– -	– -	– -	– -

We didn't make the search of the optimal iterative parameter. But some experiments are presented in Table 6 for SSORCG, $\epsilon = 10^{-5}$. In each cell the upper number corresponds to $\omega = 1.1$, the middle – to $\omega = 1.2$ and the lower – to $\omega = 1.3$.

Table 6. Numbers of iterations for different ω (1.1, 1.2, 1.3), SSORCG, $\epsilon = 10^{-5}$

n \ p	2	3	4	8	12	16
1	– 7 7	10 10 –	15 13 –	26 27 –	40 40 –	53 53 –
2	– 13 13	15 16 –	21 21 –	35 36 –	49 49 –	61 63 –
4	– 17 17	17 17 –	25 25 –	39 39 –	53 53 –	64 65 –
8	– 19 19	19 19 –	28 27 –	44 45 –	– – –	– – –
16	– 26 25	27 27 –	32 33 --	– – –	– – –	– – –

In some sense, the considered discrete boundary value problem (the Dirichlet condition in one point only) is the worst. Intuitively, for the other boundary value conditions the number of iterations has to be decrease. An additional experiments confirm such conjecture. For example, we give in Table 7 the numbers of iterations for the Dirichlet boundary value problem with the conditions, corresponded to exact solution of the Laplace equation $u = \sin \pi x \cdot \mathrm{sh}\pi y / \mathrm{sh}\pi$. In each square of Table 7 the numbers are appropriated to $\epsilon = 10^{-5}$, 10^{-6}, 10^{-7}, zero initial data u° and $\omega = 1,5$.

Table 7. Numbers of iteration for the Dirichlet boundary value problem ($\epsilon = 10^{-5}, 10^{-6}, 10^{-7}$)

n \ p	1	2	3	4
2	1, 1 ,1	6, 6 ,6	9,11,12	11,16,19
4	5, 6 ,7	11,15,19	13,19,25	17,25,32
8	6, 8 ,10	11,18,25	12,19,26	16,27,38
16	8,11,14	13,20,27		

There are similar pictures for another Dirichlet boundary value problems with different solutions. The analysis of preceding experimental data permits to make some preliminary conclusions.

a. The application of conjugate gradient acceleration has very high efficiency (in compare with steepest descent) for every values of n and p. The speed up by SSOR-preconditioner for large n, p is approximately two, for small n and p the preconditioning effect decreases.

b. The sensivity of SSORCG method to values of iterative parameter ω is small enough and differences in time costs are approximately 10–20 percent (an optimal values $\omega_{opt} \approx 1.1 \div 1.3$).

c. The increasing of "stop-iteration" criteria ϵ in (4.1) from 10^{-5} to 10^{-3} doesn't change the resulting accuracy practically, see Table 3.

d. For given accuracy the optimal pairs (n, p) correspond to minimal $n = 1$ or 2 (in the sense of minimal iteration time cost).

Of course, there are a lot of open questions on the application and optimization of iterative methods for solving high order finite element equations:

- The choosing of iterative parameters and criteria ϵ for necessary accuracy Δ of numerical solution or given values n and p;
- The role of "implicitness" of incomplete factorization methods for efficiency of iterative process;

- Adapting of iterative algorithms for different properties of original continuous problem (coefficients of equations, boundary conditions, domain shape and so on);

- The influence of uknowns ordering on the constructing of optimal iterative processes.

These and other aspects present the wide field of activity in the topic of consideration to find the main goal: the search of optimal method for solving a class of problems with given accuracy.

References

[1] B. Szabo, I. Babuska, Finite Element Analysis, John Wiley & Sons, 1991.

[2] L.A. Hageman, D.M. Young, Applied Iterative Methods, Academic Press, 1981.

[3] V.P. Il'in, Iterative Incomplete Factorization Methods, World Scientific Publishing Co., 1992.

[4] S.C. Eisenstat, Efficient implementation of a class of preconditioned conjugate gradient methods, SIAM J. Sci. Stat. Comput., Vol. 2, 1–4, 1981.