MATHEMATICS OF COMPUTATION
VOLUME 40, NUMBER 161
JANUARY 1983, PAGES 219-242

Preconditioning and Two-Level Multigrid Methods of Arbitrary Degree of Approximation

By O. Axelsson and I. Gustafsson

Abstract. Let h be a mesh parameter corresponding to a finite element mesh for an elliptic problem. We describe preconditioning methods for two-level meshes which, for most problems solved in practice, behave as methods of optimal order in both storage and computational complexity. Namely, per mesh point, these numbers are bounded above by relatively small constants for all $h \geqslant h_0$, where h_0 is small enough to cover all but excessively fine meshes.

We note that, in practice, multigrid methods are actually solved on a finite, often even a fixed number of grid levels, in which case also these methods are not asymptotically optimal as $h \to 0$. Numerical tests indicate that the new methods are about as fast as the best implementations of multigrid methods applied on general problems (variable coefficients, general domains and boundary conditions) for all but excessively fine meshes. Furthermore, most of the latter methods have been implemented only for difference schemes of second order of accuracy, whereas our methods are applicable to higher order approximations. We claim that our scheme could be added fairly easily to many existing finite element codes.

1. Introduction. Consider the numerical solution of elliptic boundary value problems discretized by finite element methods. We assume that the boundary is polygonal or consists of planes. We note that in practical problems one often has a fine enough grid already after the definition of the boundary and the minimal number of vertices needed for a first (coarse) triangulation. Anyhow, if not so, in most cases one makes only a few steps of mesh refinement. Hence the power of multigrid methods—their optimal order of computational complexity—is most often not achieved fully, because optimality requires a large number of recursively defined meshes (for details see, e.g., [4] and for further references see [7]). Hence one might as well consider other methods, perhaps simpler and more effective on a fixed mesh, but which are not asymptotically optimal.

Here we shall describe a method which uses only a fixed mesh, but for which one nevertheless achieves a low order of computational complexity and of seemingly optimal order except for, from a practical viewpoint, excessively small meshes. To be more precise, the computational cost per mesh point is bounded by $c \log N$ for $N \leqslant N_0$, where N is the number of mesh points, N_0 is large enough to cover most applications and c is small enough that the method is competitive with multigrid methods. As is well known, the latter need recursion and the usual smoothing followed by corrections of the solutions on the different mesh levels. We claim that the new method is more suitable for implementation in existing finite element packages. In fact most packages for the multigrid methods are only for second order difference methods.

Received July 22, 1981; revised June 21, 1982.

1980 *Mathematics Subject Classification*. Primary 65N30, 65F10.

©1983 American Mathematical Society
0025-5718/82/0000-0711/$05.75

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

The success of the new method is based on the following facts. Let p be the degree of the piecewise polynomial functions used in the approximation of the solution. Then,

(i) the number of vertex nodes in a "triangulation" of a domain in d dimensions ($d = 2, 3$) is only $O(N/p^d)$,

(ii) with a particular choice of basis functions one gets finite element stiffness matrices with a 2×2 block structure, where the diagonal block of largest order, namely that *not* associated with the vertex nodes, has a spectral condition number which is bounded above by a number independent of h.

These observations were already made in [3] but there they were used only for a diagonal block scaling (preconditioning). The diagonal block of smallest order was supposed to be solved exactly by a direct or a multigrid method, and the diagonal block of largest order was supposed to be solved by a simple iterative method.

We shall later see that a large gain in speed and in simplification of the method is achieved if we use incomplete factorizations either of the diagonal blocks or as a full block matrix preconditioning.

Although the method is applicable to a wide variety of partial differential equation problems, in this study we consider only second order elliptic problems.

2. Preliminaries. We prove at first some general statements needed later. Let v, t, e denote the number of vertex nodes (including those on a Dirichlet boundary), triangles and edges, respectively, in a triangulation of a plane, bounded and polygonal domain. Then $e = v + t - 1$. We assume that the triangulation is regular, i.e., all angles exceed $\theta_0 > 0$ where θ_0 is independent of N, the number of nodes.

Let $p \geqslant 2$ be an integer. In addition to the vertex points, place $p - 1$ (disjoint) nodes on each edge and (if $p \geqslant 3$), $(p - 1)(p - 2)/2$ interior nodes on each triangle. Note that the total number of nodes on each triangle is $(p + 1)(p + 2)/2$ which equals the number of coefficients in a complete polynomial (in the Euclidean coordinates x, y) of degree p. The nodes may be placed in regular positions on the edges and in the interior as is illustrated in Figure 2.1, but in Section 6 we shall present a more efficient choice of nodes (and basis functions) for $p \geqslant 3$. We will also then see that the method is applicable to the case $p = 1$.

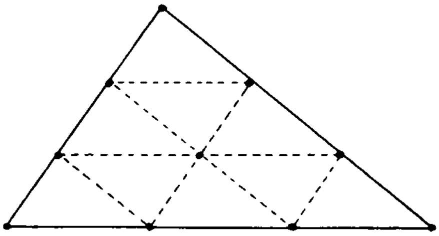


FIGURE 2.1

An example of regularly placed nodes for $p = 3$

The condition that all angles are bounded from below by $\theta_0 > 0$ is easily achieved in the following way. Let Ω_1 be a coarse mesh constructed by a triangulation of the

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

polygonal domain. This mesh is in general not uniform and may, for instance, be finer in some parts of the domain where we expect that the solution is less regular. With it we associate a mesh parameter $h = 1$. Let θ_0 be the smallest angle of all triangles in the mesh. The coarse mesh is now uniformly refined by dividing each edge by h^{-1} (an integer). Then the angles are preserved in the resulting mesh Ω_h so θ_0 is still a lower bound; see Figure 2.2. For the following we assume also that the original mesh has no angle $> \pi/2$. Hence this is so also for Ω_h. Actually, the mesh refinement can be made locally, and still the angles are preserved as Figure 2.3 shows.

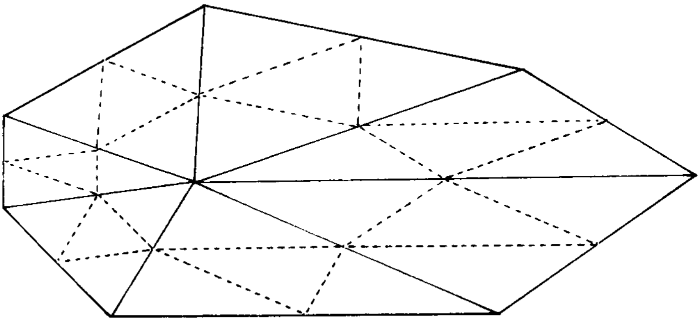


FIGURE 2.2

Uniform mesh refinement; solid lines correspond to mesh Ω_1 and solid lines $\cup$ dotted lines correspond to mesh $\Omega_{1/2}$

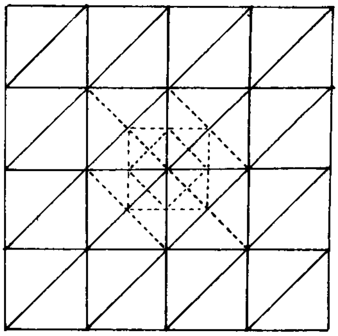


FIGURE 2.3

Local, angle-preserving mesh refinement

With every node (except those on a Dirichlet boundary) we associate a basis function with the usual compact support and whose restriction to the triangle is a polynomial of degree at most p and such that the set of the basis functions is linearly independent. Then each polynomial of degree at most p is uniquely represented as a linear combination of these basis functions. Note in particular that three, but not more, of the basis functions defined on any one triangle may be linear.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

The total number of nodes in the triangulation is

$$N = v + (p-1)e + \frac{(p-1)(p-2)}{2}t, \tag{2.1}$$

and as $N \to \infty$, $e/v \to 3$ (and therefore $t/v \to 2$). Hence by (2.1),

$$v/N \to p^{-2}, \qquad N \to \infty. \tag{2.2}$$

If the interior nodes are eliminated by static condensation (see Section 6), then the number of remaining nodes is

$$N_0 = v + (p-1)e$$

and

$$v/N_0 \to (3p-2)^{-1}, \qquad N \to \infty.$$

In a similar way, one finds that for a corresponding three-dimensional problem the ratio of vertex and total number of nodes

$$v/N \to p^{-3}, \qquad N \to \infty.$$

Another fundamental result we shall need later is the following.

Consider the boundary value problem

$$\begin{cases} -\displaystyle\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + bu = f, \qquad \mathbf{x} \in \Omega \subset \mathbf{R}^d, \\ u = 0, \mathbf{x} \in \Gamma_1, \ \displaystyle\sum_{i,j} a_{ij}\frac{\partial u}{\partial x_j}n_i = g, \mathbf{x} \in \Gamma - \Gamma_1 = \Gamma_2, \end{cases} \tag{2.3}$$

where n_i are the components of the unit normal. We assume that the (symmetric) matrix $[a_{ij}(\mathbf{x})]_{i,j=1}^{d}$ is uniformly positive definite, $b \geqslant 0$, $|a_{ij}|$ and b are uniformly bounded from above on Ω and that meas$(\Gamma_1) \neq 0$. It is only for ease of presentation that we have not considered more general boundary conditions. With this boundary value problem we associate the bilinear form

$$a(u,v) = \int_\Omega \left[\sum_{i,j} a_{ij}\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j} + buv\right] d\Omega, \qquad u, v \in V,$$

$$V = \{v \in H^1(\Omega), v = 0 \text{ on } \Gamma_1\}$$

and the variational formulation

$$a(u,v) = \int_\Omega fv \, d\Omega + \oint_{\Gamma_2} gv \, d\Gamma_2, \quad \forall v \in V.$$

a $(\cdot,\cdot)$ is symmetric, coercive and bounded so that, as is well known, for every $f \in L^2(\Omega)$, $g \in L^2(\Gamma_2)$ there exists a unique solution $\hat{u} \in V$.

The generalized C-B-S inequality,

$$|a(u,v)| \leqslant \{a(u,u)a(v,v)\}^{1/2} \quad \forall u, v \in V,$$

is easily proven.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

The upper bound is not taken if ∇u, ∇v are linearly independent, but it may (for instance if $b \equiv 0$) become arbitrarily close as ∇u, ∇v become closer to being linearly dependent. If V_1, V_2 are finite-dimensional subspaces of V with only the trivial element in common and such that V_1 contains the constant function we get an even stronger result:

$$(2.4) \qquad |a(u,v)| \leq \gamma\{a(u,u)a(v,v)\}^{1/2}, \qquad 0<\gamma<1, \forall u \in V_1, \forall v \in V_2,$$
$$V_1 \cap V_2 = \{0\}.$$

Here γ depends on the type of basis functions chosen for V_1, V_2 but it is independent on h. For the previously defined triangulation we let V_1 be the subspace spanned by the set of basis functions $\{\lambda_i\}$ associated with the vertex nodes. These basis functions should be such that the constant function is contained in the subspace. In particular, we may let the basis functions be a complete set of linear functions. V_2 is spanned by the remaining set $\{\phi_j\}$ of basis functions and hence the constant function is *not* contained in V_2. For a particular mesh the corresponding number $\gamma_0(h)$ may be calculated in the following way. Consider the bilinear form

$$a_l(u,v) = \int_{e_l} \left[\sum_{i,j} a_{ij} \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_j} + buv \right] d\Omega,$$

where e_l is the lth triangle (in an arbitrary ordering of the triangles). The corresponding number

$$(2.5) \qquad \gamma_l(h) = \max_{u \in V_1, v \in V_2} \left[|a_l(u,v)| \{a_l(u,u)a_l(v,v)\}^{-1/2} \right]$$

is calculated, and by summation we get

$$|a(u,v)| \leq \sum_l |a_l(u,v)| \leq \sum_l \gamma_l a_l(u,u)^{1/2} a_l(v,v)^{1/2}$$
$$\leq \gamma_0(h) \left\{ \sum_l a_l(u,u) \sum_l a_l(v,v) \right\}^{1/2} = \gamma_0(h)\{a(u,u)a(v,v)\}^{1/2},$$

where $\gamma_0(h) = \max_l \gamma_l(h)$. Note that because of the uniform mesh refining, there exists a γ, $\gamma_0(h) \leq \gamma < 1$, which is independent of the mesh parameter h. (In fact, $\gamma_0(h) \to \gamma$ as $h \to 0$.) For some particular examples, see Section 3.

Consider now the element matrix $\mathcal{A}_l$ associated with the triangle e_l and associate the local orders $1,2,3$ to the vertex nodes and $4,5,\ldots,q$ with the remaining nodes. For e_l, being an element at a Dirichlet boundary, we consider subsets of these nodes, and we then proceed in a similar way as follows: The matrix $\mathcal{A}_l$ has a 2×2 block form

$$\mathcal{A}_l = \begin{bmatrix} A_l & C_l \\ C_l^t & B_l \end{bmatrix},$$

where

$$A_l = \left[a_l\left(\lambda_j^{(l)}, \lambda_i^{(l)}\right)\right]_{i,j=1}^{3}, \qquad B_l = \left[a_l\left(\phi_j^{(l)}, \phi_i^{(l)}\right)\right]_{i,j=4}^{q},$$
$$C_l = \left[a_l\left(\phi_j^{(l)}, \lambda_i^{(l)}\right)\right]_{i=1,2,3, j=4,5,\ldots,q}.$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Let

$$\mathbf{A}_l = \begin{bmatrix} A_l & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{B}_l = \begin{bmatrix} 0 & 0 \\ 0 & B_l \end{bmatrix},$$

and consider the following quotients between the quadratic forms associated with $\mathcal{Q}_l$ and with the diagonal block part of $\mathcal{Q}_l$.

Note at first that by (2.4) we have (since $|a_1a_2| \leqslant \frac{1}{2}(a_1^2 + a_2^2)$)

$$\begin{aligned} (2.6) \qquad & (1-\gamma)[a(u,u) + a(v,v)] \\ & \leqslant a(u,u) + a(v,v) + 2a(u,v) = a(u+v, u+v) \\ & \leqslant (1+\gamma)[a(u,u) + a(v,v)] \quad \forall u \in V_1, \forall v \in V_2. \end{aligned}$$

With

$$u = \sum_{i=1}^{3} \alpha_i^{(l)} \lambda_i^{(l)}, \quad v = \sum_{i=4}^{q} \alpha_i^{(l)} \phi_i^{(l)}, \quad w = u + v,$$

we get

$$\boldsymbol{\alpha}^{(l)'} \mathcal{Q}_l \boldsymbol{\alpha}^{(l)} = a_l(w,w), \qquad \boldsymbol{\alpha}^{(l)'} \mathbf{A}_l \boldsymbol{\alpha}^{(l)} = a_l(u,u) \quad \text{and} \quad \boldsymbol{\alpha}^{(l)'} \mathbf{B}_l \boldsymbol{\alpha}^{(l)} = a_l(v,v).$$

Hence by (2.6)

$$(1-\gamma_l)\boldsymbol{\alpha}^{(l)'}(\mathbf{A}_l + \mathbf{B}_l)\boldsymbol{\alpha}^{(l)} \leqslant \boldsymbol{\alpha}^{(l)'} \mathcal{Q}_l \boldsymbol{\alpha}^{(l)} \leqslant (1+\gamma_l)\boldsymbol{\alpha}^{(l)'}(\mathbf{A}_l + \mathbf{B}_l)\boldsymbol{\alpha}^{(l)},$$

where γ_l is the constant defined in (2.5).

Finally, by summation over all triangles e_l, we get the global correspondence

$$(2.7) \qquad (1-\gamma)\boldsymbol{\alpha}'(\mathbf{A}+\mathbf{B})\boldsymbol{\alpha} \leqslant \boldsymbol{\alpha}' \mathcal{Q} \boldsymbol{\alpha} \leqslant (1+\gamma)\boldsymbol{\alpha}'(\mathbf{A}+\mathbf{B})\boldsymbol{\alpha} \quad \forall \boldsymbol{\alpha} \in \mathbf{R}^N.$$

We then make a global ordering of nodal points such that vertex nodes are numbered first. Then the resulting assembled matrices have the following properties, which was already observed in [3].

LEMMA 2.1. *Let the set of basis functions* $\{\lambda_i\}$ *and* $\{\phi_j\}$ *be defined as above. Then the matrices* $A = [a(\lambda_j, \lambda_i)]$ *and* $B = [a(\phi_j, \phi_i)]$ *are positive definite and have spectral condition numbers* $\kappa_1 = O(h^{-2})$, $h \to 0$ *and* $\kappa_2 = O(1)$, $h \to 0$, *respectively.*

Proof. The result for the matrix A is well known (see e.g. [4]). In particular, positive definiteness follows because $\text{meas}(\Gamma_1) \neq 0$. For the matrix $B_l = [a_l(\phi_j, \phi_i)]_{i,j=4}^q$ we get

$$0 < \mu_l^{(1)} \boldsymbol{\alpha}_l' \boldsymbol{\alpha}_l \leqslant a_l(v,v) \leqslant \mu_l^{(0)} \boldsymbol{\alpha}_l' \boldsymbol{\alpha}_l \quad \forall v = \sum_{i=4}^{q} \alpha_{l_i} \phi_i^{(l)},$$

where $\mu_l^{(1)}$, $\mu_l^{(0)}$ are the extreme eigenvalues of B_l. Note that $\mu_l^{(1)}$ is positive because constant functions are excluded from the space V_2. By summation over the triangles e_l we get

$$0 < \min_l \mu_l^{(1)} \boldsymbol{\alpha}' \boldsymbol{\alpha} \leqslant \boldsymbol{\alpha}' B \boldsymbol{\alpha} \leqslant p_0 \max_l \mu_l^{(0)} \boldsymbol{\alpha}' \boldsymbol{\alpha},$$

where p_0 $(= 2)$ is the largest number of triangles meeting at any same nonvertex node. Hence

$$\kappa_2 = \kappa(B) \leqslant p_0 \max_l \mu_l^{(0)} / \min_l \mu_l^{(1)},$$

and this number is bounded above by a number independent on h, because of the uniform mesh refining. □

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

LEMMA 2.2. *The spectral condition number of*

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}^{-1} \mathcal{A}, \qquad \mathcal{A} = \begin{bmatrix} A & C \\ C^t & B \end{bmatrix},$$

is bounded from above by $(1+\gamma)/(1-\gamma)$ *and* $\|F\|_2 \leqslant \gamma$, *where* $F = A^{-1/2}CB^{-1/2}$.

Proof. The first part follows at once from (2.7) because

$$(2.8) \qquad 1-\gamma \leqslant \frac{\boldsymbol{\alpha}^t \mathcal{A} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^t(\mathbf{A}+\mathbf{B})\boldsymbol{\alpha}} \leqslant 1+\gamma \quad \forall \boldsymbol{\alpha} \in \mathbf{R}^N, \boldsymbol{\alpha} \neq \mathbf{0}.$$

For the second part we note that

$$\begin{bmatrix} A^{-1/2} & 0 \\ 0 & B^{-1/2} \end{bmatrix} \mathcal{A} \begin{bmatrix} A^{-1/2} & 0 \\ 0 & B^{-1/2} \end{bmatrix} = \begin{bmatrix} I & F \\ F^t & I \end{bmatrix},$$

so by (2.8), the eigenvalues of $[\begin{smallmatrix} 0 & F \\ F^t & 0 \end{smallmatrix}]$ are in the interval $[-\gamma, \gamma]$. Hence

$$\|F\|_2 = \{\rho(FF^t)\}^{1/2} \leqslant \gamma,$$

where $\rho(\cdot)$ is the spectral radius. □

Note that the above upper bounds are independent of h, that is of the number of nodes N. The first part of Lemma 2.2 was proved in [3] in a slightly different way.

3. Preconditioned Iterative Methods. The discretized version of (2.3) is

$$(3.1) \qquad a(u,v) = \int_\Omega fv\, d\mathbf{x} + \oint_{\Gamma_2} gv\, ds \quad \forall v \in V_h^{(p)} = V_1 \oplus V_2,$$

where $u \in V_h^{(p)}$ and $V_h^{(p)}$ is the finite-dimensional subspace of V consisting of continuous, piecewise polynomials of degree at most p. With the previously defined ordering of the nodes we get a linear system of algebraic equations on the form

$$(3.2) \quad \begin{bmatrix} A & C \\ C^t & B \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \quad \text{where } \mathbf{f}_k = \int_\Omega fv\, d\mathbf{x} + \int_{\Gamma_2} gv\, ds,\ v = \begin{cases} \lambda_i, & k=1, \\ \phi_i, & k=2. \end{cases}$$

Let

$$\mathcal{A} = \begin{bmatrix} A & C \\ C^t & B \end{bmatrix}, \quad \mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)^t, \quad \mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2)^t.$$

Then (3.2) is equivalent to $\mathcal{A}\mathbf{u} = \mathbf{f}$, and we shall study iterative methods to solve this system of equations. A basic iterative method can be stated

$$(3.3) \qquad \mathcal{C}(\mathbf{u}^{(m+1)} - \mathbf{u}^{(m)}) = -\beta_m(\mathcal{A}\mathbf{u}^{(m)} - \mathbf{f}), \qquad m = 0, 1, \ldots,$$

where $\mathcal{C}$ is a so-called preconditioning matrix and β_m are iteration parameters. $\mathcal{C}$ will be chosen as a product of two sparse triangular matrices and is symmetric, positive definite. It has the same block partitioning as $\mathcal{A}$. The rate of convergence of (3.3) depends on the spectral condition number κ_1 of $\mathcal{C}^{-1/2}\mathcal{A}\mathcal{C}^{-1/2}$. For instance, if the conjugate gradient method is used to accelerate (3.3), then the number of iterations to reach a fixed relative accuracy is bounded above by a number proportional to $\sqrt{\kappa_1}$, see, e.g., [1].

In the sections to follow we shall consider various choices of $\mathcal{C}$. The most efficient among these involve (modified) incomplete factorizations of the block diagonal

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

matrices A and B. We briefly recall the basic results regarding modified incomplete Cholesky (MIC) factorizations; for details see [6]. These methods are modifications of the methods presented in [9].

The MIC methods will be applied to the matrix A and can be stated

$$LL^t = \tilde{A} = A + \tilde{D} + R,$$

where $\tilde{D}$ is a positive diagonal matrix and R is the defect matrix being positive semidefinite and having row sums equal to zero. The degree of accuracy of the factorization can be controlled by letting L contain more or less nonzero entries. For well-structured (model) problems, where A has nonzero entries only in certain (sub-) diagonals, we use the notation MIC(d) to indicate that L contains d more nonzero diagonals than (the lower part of) A.

Applied to a weakly diagonally dominant matrix, the MIC factorizations are stable. If further A is an L-matrix, i.e., if $a_{ij} \leqslant 0$, $i \neq j$ (which is the case in our applications), then $\kappa(\tilde{A}^{-1/2}A\tilde{A}^{-1/2}) = O(h^{-1})$, $h \to 0$ (while $\kappa(A) = O(h^{-2})$). More precisely, there exist constants C_1, C_2, C_3 independent of h such that

$$C_1 \leqslant \mathbf{z}^t A\mathbf{z}/\mathbf{z}^t\tilde{A}\mathbf{z} \leqslant C_2 + C_3 h^{-1} \quad \forall \mathbf{z} \in R^{N'}, \tag{3.4}$$

where $N' = O(h^{-2})$ is the order of A. For instance, for a model Laplace problem over the unit square with uniform right-angled triangulation and linear f.e., (3.4) holds with $C_1 = \frac{1}{2}$, $C_2 = 1$, $C_3 = 1/\pi$ for the MIC(0) method and $C_3 = 0.68/\pi$ for the MIC(1) method.

4. Diagonal Block Preconditioning. In this section we let $\mathcal{C} = \mathfrak{D} = [\begin{smallmatrix} \tilde{A} & 0 \\ 0 & \tilde{B} \end{smallmatrix}]$ where $\tilde{A}$, $\tilde{B}$ may be regarded as approximations of A, B, respectively. We assume that they are symmetric and positive definite.

Let $\tilde{a}_l(u, v)$, $\tilde{b}_l(u, v)$ be the bilinear forms corresponding to $\tilde{A}$, $\tilde{B}$, i.e., the restrictions to the lth element of $\boldsymbol{\alpha}^t\tilde{A}\boldsymbol{\beta}$, $\boldsymbol{\alpha}^t\tilde{B}\boldsymbol{\beta}$, with

$$u = \sum_{i=1}^{3} \alpha_{l_i}\lambda_i^{(l)}, \qquad v = \sum_{i=4}^{q} \beta_{l_i}\phi_i^{(l)}.$$

Let $a^{(i)}$, $b^{(i)}$, $i = 0, 1$, be positive numbers such that

$$\begin{cases} a_l^{(1)}\tilde{a}_l(u, u) \leqslant a_l(u, u) \leqslant a_l^{(0)}\tilde{a}_l(u, u) & \forall u = \sum\limits_{i=1}^{3} \alpha_{l_i}\lambda_i^{(l)}, \\ b_l^{(1)}\tilde{b}_l(v, v) \leqslant a_l(v, v) \leqslant b_l^{(0)}\tilde{b}_l(v, v) & \forall v = \sum\limits_{i=4}^{q} \beta_{l_i}\phi_i^{(l)}. \end{cases} \tag{4.1}$$

We have

$$|a_l(u, v)| \leqslant \gamma_l a_l(u, u)^{1/2} a_l(v, v)^{1/2} \leqslant \tfrac{1}{2}\gamma\zeta_l a_l(u, u) + \tfrac{1}{2}\gamma_l\zeta_l^{-1} a_l(v, v), \tag{4.2}$$

where $\zeta_l > 0$.

In the following theorem we give an upper bound for the condition number of $\mathfrak{D}^{-1/2}\mathfrak{A}\mathfrak{D}^{-1/2}$.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

THEOREM 4.1. *Let $\mathcal{A}, \mathcal{D}$ be defined as above. Assume that* (4.1) *and* (4.2) *are satisfied. Then*

$$\text{(4.3)}\quad \kappa_1 = \kappa(\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2})$$

$$\leqslant \max_l \left\{ \frac{1}{1-\gamma_l^2}\frac{b_l^{(0)}}{b_l^{(1)}} \left[\frac{1}{2}\left(1+\frac{a_l^{(0)}}{b_l^{(0)}}\right) + \left\{ \left[\frac{1}{2}\left(1-\frac{a_l^{(0)}}{b_l^{(0)}}\right)\right]^2 + \frac{a_l^{(0)}}{b_l^{(0)}}\gamma_l^2 \right\}^{1/2} \right] \right.$$

$$\left. \times \left[\frac{1}{2}\left(1+\frac{b_l^{(1)}}{a_l^{(1)}}\right) + \left\{ \left[\frac{1}{2}\left(1-\frac{b_l^{(1)}}{a_l^{(1)}}\right)\right]^2 + \frac{b_l^{(1)}}{a_l^{(1)}}\gamma_l^2 \right\}^{1/2} \right] \right\}.$$

Proof. An upper bound for the eigenvalues of $\mathcal{D}^{-1}\mathcal{A}$ is found in the following way. By (2.6) and (4.2),

$$\begin{aligned} a_l(u+v, u+v) &\leqslant (1+\gamma_l\zeta_l)a_l(u,u) + \left(1+\gamma_l\zeta_l^{-1}\right)a_l(v,v) \\ &\leqslant (1+\gamma_l\zeta_l)a_l^{(0)}\tilde{a}_l(u,u) + \left(1+\gamma_l\zeta_l^{-1}\right)b_l^{(0)}\tilde{b}_l(v,v). \end{aligned}$$

Hence

$$\text{(4.4)}\qquad a_l(u+v, u+v) \leqslant (1+\gamma_l\beta_l)a_l^{(0)}\left[\tilde{a}_l(u,u) + \tilde{b}_l(v,v)\right],$$

if ζ_l is such that $(1+\gamma_l\zeta_l)a_l^{(0)} = (1+\gamma_l\zeta_l^{-1})b_l^{(0)}$, that is,

$$\zeta_l = \frac{b_l^{(0)}/a_l^{(0)} - 1}{2\gamma_l} \underset{(-)}{+} \left[\left(\frac{b_l^{(0)}/a_l^{(0)} - 1}{2\gamma_l} \right)^2 + \frac{b_l^{(0)}}{a_l^{(0)}} \right]^{1/2}.$$

Hence

$$(1+\gamma_l\zeta_l)a_l^{(0)} = \left(b_l^{(0)} + a_l^{(0)}\right)/2 + \left\{ \left[(b^{(0)} - a^{(0)})/2\right]^2 + b_l^{(0)}a_l^{(0)}\gamma_l^2 \right\}^{1/2}.$$

(Note that the upper bound (4.3) is sharper than the trivial upper bound we get with $\zeta_l = 1$, namely

$$a_l(u+v, u+v) \leqslant (1+\gamma_l)\max\left(a_l^{(0)}, b_l^{(0)}\right)\left[\tilde{a}_l(u,u) + \tilde{b}_l(v,v)\right].$$

Assume that $b_l^{(0)} \geqslant a_l^{(0)}$. (This can always be achieved by scaling.) Then in fact

$$(1+\gamma_l)b_l^{(0)} - (1+\gamma_l\beta_l)a_l^{(0)}$$

$$= \gamma_l b_l^{(0)} \left\{ 1 - a_l^{(0)}\gamma_l \left[\frac{b_l^{(0)} - a_l^{(0)}}{2} + \left(\left(\frac{b_l^{(0)} - a_l^{(0)}}{2} \right)^2 + b_l^{(0)}a_l^{(0)}\gamma_l^2 \right)^{1/2} \right]^{-1} \right\} \geqslant 0.)$$

Similarly we get a lower bound. Let $\gamma_l < \xi_l < \gamma_l^{-1}$. Then

$$\begin{aligned} \text{(4.5)}\qquad a_l(u+v, u+v) &\geqslant \left(1-\gamma_l\xi_l^{-1}\right)a_l(u,u) + (1-\gamma_l\xi_l)a_l(v,v) \\ &\geqslant (1-\gamma_l\xi_l)b_l^{(1)}\left[\tilde{a}_l(u,u) + \tilde{b}_l(v,v)\right], \end{aligned}$$

where ξ_l is such that

$$\left(1-\gamma_l\xi_l^{-1}\right)a_l^{(1)} = (1-\gamma_l\xi_l)b_l^{(1)},$$

i.e.,

$$\xi_l = \frac{1 - a_l^{(1)}/b_l^{(1)}}{2\gamma_l} + \left[\left(\frac{1 - a_l^{(1)}/b_l^{(1)}}{2\gamma_l} \right)^2 + \frac{a_l^{(1)}}{b_l^{(1)}} \right]^{1/2}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

and

$$(1 - \gamma_l\xi_l)b_l^{(1)}$$

$$= a_l^{(1)}b_l^{(1)}(1-\gamma_l^2)\left\{\frac{b_l^{(1)}+a_l^{(1)}}{2} + \left[\left(\frac{b_l^{(1)}+a_l^{(1)}}{2}\right)^2 + a_l^{(1)}b_l^{(1)}\gamma_l^2\right]^{1/2}\right\}^{-1} > 0.$$

Finally from (4.4) and (4.5) the result follows by summation over all elements. □

We note that in the case $b_l^{(0)} = a_l^{(0)}$, $b_l^{(1)} = a_l^{(1)}$ (4.3) reduces to the trivial bound

$$\kappa_1 \leqslant \max_l \left\{\frac{1+\gamma_l}{1-\gamma_l}\frac{b_l^{(0)}}{b_l^{(1)}}\right\}. \tag{4.6}$$

In particular, for $\tilde{A} = A$, $\tilde{B} = B$ we have

$$\kappa_1 \leqslant \max_l \left\{\frac{1+\gamma_l}{1-\gamma_l}\right\} = \frac{1+\gamma}{1-\gamma}, \tag{4.7}$$

which we already derived in Lemma 2.2.

We also remark that we can often obtain a sharper bound than that given in (4.3) by calculating (on each element) upper and lower bounds $\lambda_l^{(1)}, \lambda_l^{(0)}$ of the quotient

$$\frac{a_l(u+v, u+v)}{\tilde{a}_l(u,u) + \tilde{b}_l(v,v)}, \quad u = \sum_{i=1}^{3} \alpha_{l_i}\lambda_i^{(l)}, \quad v = \sum_{i=4}^{q} \beta_{l_i}\phi_i^{(l)}. \tag{4.8}$$

Then,

$$\kappa_1 \leqslant \max_l \lambda_l^{(0)} / \min_l \lambda_l^{(1)}.$$

(A similar approach was used in [2].)

In Figure 4.1 we have plotted the bound on the condition number κ_1 as a function of h^{-1} when $\tilde{B} = \operatorname{diag}(B)$ and $\tilde{A}$ is a MIC(1) factorization of A. Here $p = 2$, $[a_{ij}]_{i,j=1}^2 = I$, $b = 0\ \forall \mathbf{x} \in \Omega = \{(x_1, x_2) \in [0,1] \times [0,1]\}$ and we assume a uniform right-angled triangulation of Ω. This problem shall be referred to as the model problem. The bounds $a_l^{(0)}$ and $a_l^{(1)}$ for the eigenvalues of $\tilde{A}^{-1}A$ were taken from (3.4).

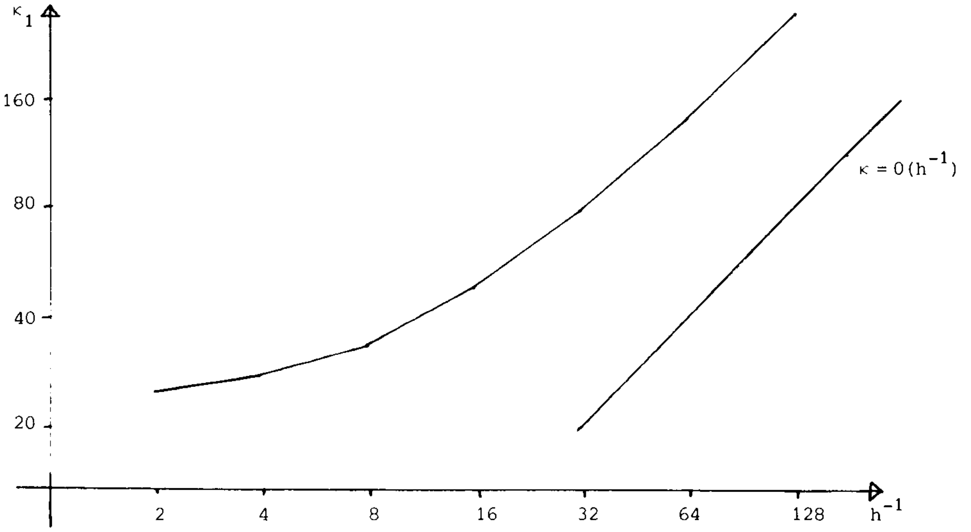


FIGURE 4.1

The upper bound for the condition number κ_1 as a function of h^{-1} for $p = 2$, $\tilde{B} = \operatorname{diag}(B)$ and $\tilde{A}$ a MIC(1) *factorization of A. Model Poisson problem. The scales are logarithmic and the slope of the "line" $\kappa = Ch^{-1}$ is indicated.*

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

We note that the condition number is only slightly increasing for $h \geqslant h_0$, say $h_0 = 1/8$, but for $h \ll h_0$ it grows like $O(h^{-1})$, $h \to 0$. By use of a more accurate incomplete factorization one can make h_0 smaller; see the numerical tests in Section 6.

By calculating upper and lower bounds of the quotient (4.8) we have also found that κ_1 is (often) rather insensitive to $\kappa_0 = \kappa(\tilde{B}^{-1}B)$. For our model problem and $p = 2$, κ_1 increases only by a factor $3/2$ when $\tilde{B} = B$ ($\kappa_0 = 1$) is replaced by $\tilde{B} = \operatorname{diag}(B)$ ($\kappa_0 \simeq 6$).

5. Full Block Incomplete Factorization. Let, as before, $L_1 L_1^t = \tilde{A}$ and $L_2 L_2^t = \tilde{B}$ be two incomplete factorizations of A and B, respectively. We consider preconditionings of $\mathcal{A}$ by a full block incomplete factorization on the form

$$\mathcal{C} = \mathcal{F} = \begin{bmatrix} L_2 & 0 \\ CL_2^{-t} & L_1 \end{bmatrix} \begin{bmatrix} L_2^t & L_2^{-1}C^t \\ 0 & L_1^t \end{bmatrix} = \begin{bmatrix} \tilde{B} & C^t \\ C & \tilde{A} + C\tilde{B}^{-1}C^t \end{bmatrix}.$$

Note that we have reordered the system so that

$$\mathcal{A} = \begin{bmatrix} B & C^t \\ C & A \end{bmatrix}.$$

Since $\tilde{A}$ and $\tilde{B}$ are symmetric and positive definite, so is $\mathcal{F}$. Let a_i, b_i, $i = 0, 1$, be lower and upper bounds of the following quotients of the bilinear forms corresponding to B, $\tilde{B}$ and A, $\tilde{\tilde{A}} = \tilde{A} + C\tilde{B}^{-1}C^t$, respectively:

$$\begin{aligned} 0 < b_1 \leqslant a(v,v)/\tilde{b}(v,v) \leqslant b_0 \quad & \forall v \in V_2, \\ 0 < a_1 \leqslant a(u,u)/\tilde{a}(u,u) \leqslant a_0 \quad & \forall u \in V_1. \end{aligned}$$

We assume that $\tilde{A}$ and $\tilde{B}$ are scaled such that

$$b_1 \leqslant 1 \leqslant b_0 < \gamma^{-2}, \quad a_1 \leqslant 1 \leqslant a_0 \leqslant \gamma^{-2}.$$

THEOREM 5.1. *Let $\mathcal{A}$, $\mathcal{F}$, a_i, b_i, $i = 0, 1$ be defined as above. Then*

$$\kappa(\mathcal{F}^{-1}\mathcal{A}) = \max_i \mu_i / \min_i \mu_i,$$

where

$$(5.1) \qquad \max_i \mu_i \leqslant 1 + \frac{1}{1-\gamma^2}\left\{\frac{a_1^{-1} + b_1^{-1}}{2} - 1 + \left[\left(\frac{a_1^{-1} - b_1^{-1}}{2}\right)^2 + \left(b_1^{-1} - 1\right)\left(a_1^{-1} - 1\right)\gamma^2\right]^{1/2}\right\},$$

$$(5.2) \qquad \min_i \mu_i \geqslant 1 - \frac{1}{1-\gamma^2}\left\{1 - \frac{a_0^{-1} + b_0^{-1}}{2} - \left[\left(\frac{a_0^{-1} - b_0^{-1}}{2}\right)^2 + \left(1 - a_0^{-1}\right)\left(1 - b_0^{-1}\right)\gamma^2\right]^{1/2}\right\}.$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Proof. We assume at first that $a_1 < 1$. Let $v = \sum \alpha_i \phi_i$, $u = \sum \beta_i \lambda_i$, and let μ_i be the eigenvalues of $\mathcal{A}^{-1}\mathcal{F}$. We have

$$\begin{aligned}\mu &:= [\alpha^t, \beta^t]\mathcal{F}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \Big/ [\alpha^t, \beta^t]\mathcal{A}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\\ &= \frac{\tilde{b}(v,v) + 2a(v,u) + \tilde{\tilde{a}}(u,u)}{a(u+v, u+v)}\\ &= 1 + \frac{\tilde{b}(v,v) - a(v,v) + \tilde{\tilde{a}}(u,u) - a(u,u)}{a(v,v) + 2a(v,u) + a(u,u)}\\ &\leqslant 1 + \frac{(b_1^{-1} - 1)a(v,v) + (a_1^{-1} - 1)a(u,u)}{(1 - \gamma\zeta)a(v,v) + (1 - \gamma\zeta^{-1})a(u,u)}, \qquad \gamma < \zeta < \gamma^{-1}, \forall u \in V_1, v \in V_2.\end{aligned}$$

We choose ζ such that

$$r_0 = \frac{b_1^{-1} - 1}{a_1^{-1} - 1} = \frac{1 - \gamma\zeta}{1 + \gamma\zeta^{-1}} \quad \text{i.e. } \zeta = \frac{1 - r_0}{2\gamma} + \left[\left(\frac{1 - r_0}{2\gamma}\right)^2 + r_0\right]^{1/2}.$$

Then (5.1) follows, and because of symmetry ($a_1 \leftrightarrow b_1$) we realize that this bound is valid also for $a_1 = 1$. In the same way we get

$$\mu \geqslant 1 - \frac{(1 - b_0^{-1})a(v,v) + (1 - a_0^{-1})a(u,u)}{(1 - \gamma\xi)a(v,v) + (1 - \gamma\xi^{-1})a(u,u)},$$

where

$$\xi = \frac{1 - s_0}{2\gamma} + \left[\frac{(1 - s_0)^2}{2\gamma} + s_0\right]^{1/2}, \qquad s_0 = \frac{1 - b_0^{-1}}{1 - a_0^{-1}}.$$

Hence (5.2) follows and the theorem is proved. □

We consider now some special cases:

Case (i): $\tilde{B} = B$. Then $b_1 = b_0 = 1$ so

$$\kappa(\mathcal{F}^{-1}\mathcal{A}) \leqslant \frac{a_1^{-1} - \gamma^2}{a_0^{-1} - \gamma^2} = \kappa(\tilde{\tilde{A}}^{-1}A)\frac{1 - a_1\gamma^2}{1 - a_0\gamma^2}. \tag{5.3}$$

Case (ii): $\tilde{B} = B$, $a_0 = 1$. Then from (5.3)

$$\kappa(\mathcal{F}^{-1}\mathcal{A}) \leqslant \frac{\kappa(\tilde{\tilde{A}}^{-1}A) - \gamma^2}{1 - \gamma^2}.$$

In particular, if $\tilde{B} = B$, $\tilde{A} = A$, then from Lemma 2.2, $A \leqslant \tilde{\tilde{A}} \leqslant A(1 + \gamma^2)$. (Here, inequality stands for inequality between the corresponding quadratic forms.) Hence

$$\kappa(\mathcal{F}^{-1}\mathcal{A}) \leqslant 1/(1 - \gamma^2). \tag{5.4}$$

Case (iii): $\tilde{A} = A$, $\tilde{B} \geqslant B$. Then $a_1 = (1 + \gamma^2)^{-1}$, $a_0 = b_0 = 1$, $b_1 = \kappa_0^{-1}$, where $\kappa_0 = \kappa(\tilde{B}^{-1}B)$, and we get

$$\begin{aligned}\kappa(\mathcal{F}^{-1}\mathcal{A}) &\leqslant 1 + \frac{1}{1 - \gamma^2}\left\{\frac{\gamma^2 + \kappa_0 - 1}{2} + \left[\left(\frac{\kappa_0 - 1 - \gamma^2}{2}\right)^2 + (\kappa_0 - 1)\gamma^4\right]^{1/2}\right\}\\ &\nearrow \kappa_0/(1 - \gamma^2) \quad \text{as } \kappa_0 \to \infty.\end{aligned} \tag{5.5}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Case (iv): $\tilde{A} = A - C\tilde{B}^{-1}C^t$ (that is $\tilde{\tilde{A}} = A$), $\tilde{B} \geqslant B$. Then $a_1 = a_0 = b_1 = 1$, $b_1 = \kappa_0^{-1}$ and

$$\kappa(\mathcal{F}^{-1}\mathcal{Q}) \leqslant (\kappa_0 - \gamma^2)/(1-\gamma^2). \tag{5.6}$$

Case (v): $\tilde{A} = A - C\tilde{\tilde{B}}^{-1}C^t$, $\tilde{B} = B$ where $\tilde{\tilde{B}}$ is any matrix, e.g. a diagonal matrix, such that $\tilde{\tilde{B}} \geqslant B$. Then $b_0 = b_1 = a_0 = 1$, $a_1 = [1 + (1 - \kappa_0^{-1})\gamma^2]^{-1}$, where $\kappa_0 = \kappa(\tilde{\tilde{B}}^{-1}B)$ and

$$\kappa(\mathcal{F}^{-1}\mathcal{Q}) \leqslant (\kappa_0 - \gamma^2)/\{\kappa_0(1-\gamma^2)\}. \tag{5.7}$$

We note that Cases (iv) and (v) involve making an (incomplete) factorization of $A - C\tilde{B}^{-1}C^t$ (or $A - C\tilde{\tilde{B}}^{-1}C^t$), which makes it necessary to assemble this matrix. However, in practice κ_0 is often much larger than γ (see Section 6), which leads to minor differences between the (estimates of the) condition numbers in Case (ii) and Case (v) and in Case (iii) and Case (iv), respectively. Hence it does not pay off to make the more complicated factorization (of $A - C\tilde{B}^{-1}C^t$).

We assume that $\tilde{B}$ is obtained by a (stable) IC factorization of B so $\kappa_0 = \kappa(\tilde{B}^{-1}B) = O(1)$, $h \to 0$. (In particular $\tilde{B}$ might even be equal to diag(B).) Since B is in general not an M-matrix, it might be necessary to use shifted incomplete factorizations (SIC) [8] or in some other way ensure a stable factorization. For $\tilde{A} = A$ we then easily see that $\kappa(\mathcal{F}^{-1}\mathcal{Q}) = O(1)$, $h \to 0$ as well. As in the diagonal block reconditioning, we get $\kappa(\mathcal{F}^{-1}\mathcal{Q}) = O(h^{-1})$, $h \to 0$ if $\tilde{A}$ is a MIC factorization of A, where values of a_0, a_1 can be derived from (3.4). However, the condition number behaves in the same way as in the diagonal block case, i.e., it is almost independent of h for $h \geqslant h_0$, where h_0 is dependent on the degree of accuracy of the MIC factorization.

Let us now compare the upper bounds for the condition numbers $\kappa(\mathcal{D}^{-1}\mathcal{Q})$, $\kappa(\mathcal{F}^{-1}\mathcal{Q})$ of the diagonal and full block preconditionings. At first note that in the case $\tilde{B} = B$, $\tilde{A} = A$, by (4.8) and (5.4),

$$\kappa(\mathcal{D}^{-1}\mathcal{Q})/\kappa(\mathcal{F}^{-1}\mathcal{Q}) = \frac{(1+\gamma)(1-\gamma^2)}{1-\gamma} = (1+\gamma)^2 < 4.$$

In practice γ is close to one, and hence we can expect about twice as many iterations for the diagonal block as for the full block preconditioning. On the other hand, the full block method needs more computational work per iteration (e.g. the solution of 6 triangular systems) so it will be preferable only if we use *incomplete* factorizations of B (and A).

In the full block preconditioning the bounds of $\kappa(\mathcal{F}^{-1}\mathcal{Q})$ are proportional to κ_0, while (as already pointed out in Section 4) in the diagonal block method $\kappa(\mathcal{D}^{-1}\mathcal{Q})$ is fairly insensitive to κ_0. This indicates that the full block factorization is more effective relative to the diagonal block factorization for more accurate (but not too accurate) incomplete factorizations of B than for less accurate factorizations.

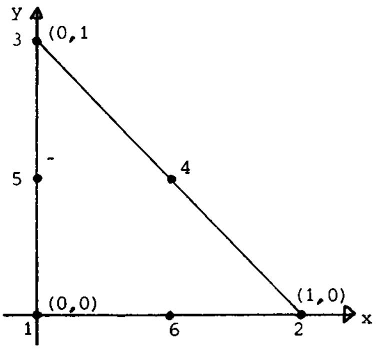
6. Examples and Numerical Tests. As our model problem we take

$$\begin{cases} -\nabla(a\nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{6.1}$$

where Ω is the unit square, $a \equiv 1$ and f is a constant function. We make a uniform right-angled triangulation (with triangle sides of length h, h, $\sqrt{2}h$) to obtain Ω_h.

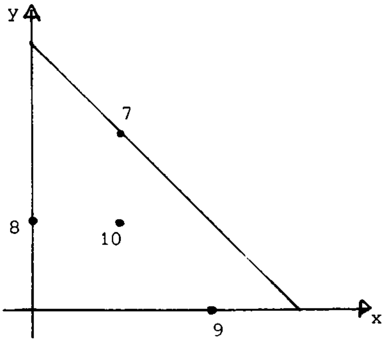
License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

In case $p = 2$ we choose linear basis functions associated to vertex points and quadratic functions associated to the midpoints of the edges, see Figure 6.1. For $p = 3$ we add four cubic basis functions, the standard Lagrangian cubic basis functions associated with one point on each side and the midpoint; see Figure 6.2.



$$\phi_1 = 1-x-y \qquad \phi_4 = 4xy$$
$$\phi_2 = x \qquad \phi_5 = 4(1-x-y)y$$
$$\phi_3 = y \qquad \phi_6 = 4(1-x-y)x$$

FIGURE 6.1
Basis functions for $p = 2$ (on basic element)



$$\phi_7 = \frac{27}{2}(y-1/3)xy$$
$$\phi_8 = \frac{27}{2}(2/3-x-y)y(1-x-y)$$
$$\phi_9 = \frac{27}{2}(1-x-y)/(x-1/3)x$$
$$\phi_{10} = 27(1-x-y)xy$$

FIGURE 6.2
Additional basis functions for $p = 3$

The corresponding element matrix is

$$\mathcal{Q}_l = \frac{1}{6}\left[\begin{array}{rrr|rrr} 6 & -3 & -3 & -8 & 4 & 4 \\ -3 & 3 & 0 & 4 & -4 & 0 \\ -3 & 0 & 3 & 4 & 0 & -4 \\ \hline -8 & 4 & 4 & 16 & -8 & -8 \\ 4 & -4 & 0 & -8 & 16 & 0 \\ 4 & 0 & -4 & -8 & 0 & 16 \end{array}\right]$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

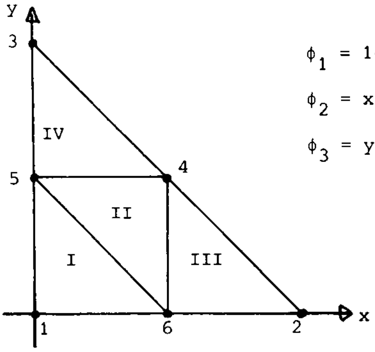
for $p = 2$ and

$$\mathcal{A}_l = \frac{1}{240}\left[\begin{array}{rrr|rrr|rrr|r}
240 & -120 & -120 & -320 & 160 & 160 & -180 & 90 & 90 & 0 \\
-120 & 120 & 0 & 160 & -160 & 0 & 90 & -90 & 0 & 0 \\
-120 & 0 & 120 & 160 & 0 & -160 & 90 & 0 & -90 & 0 \\
\hline
-320 & 160 & 160 & 640 & -320 & -320 & 360 & -72 & -288 & 0 \\
160 & -160 & 0 & -320 & 640 & 0 & -504 & 144 & 0 & 432 \\
160 & 0 & -160 & -320 & 0 & 640 & -72 & -216 & 360 & 432 \\
\hline
-180 & 90 & 90 & 360 & -504 & -72 & 810 & 81 & 81 & -486 \\
90 & -90 & 0 & -72 & 144 & -216 & 81 & 810 & 0 & -486 \\
90 & 0 & -90 & -288 & 0 & 360 & 81 & 0 & 810 & 0 \\
\hline
0 & 0 & 0 & 0 & 432 & 432 & -486 & -486 & 0 & 1944
\end{array}\right]$$

for $p = 3$.

Construction of Local Finite Element Matrices. We note that the 3×3 and 6×6 principal submatrices of the element matrix for $p = 3$ are the element matrices corresponding to $p = 1$ and $p = 2$, respectively. In general, for $p \geqslant 2$ we add to the set of basis functions for $V_h^{(p-1)}$, $p + 1$ complete basis functions for $V_h^{(p)} \setminus V_h^{(p-1)}$. In this way we can build up the element matrix for $p = q$ successively from those for $p = 1, p = 2, \ldots, p = q - 1$. For $p \geqslant 3$ we now eliminate all interior nodes (by static condensation), i.e. in the case $p = 3$ node nr. 10, see Figure 6.2. In general the number of interior nodes is $(p - 2)(p - 1)/2$. Besides the reduction of the number of unknowns this has the desired effect of reducing the condition number $\kappa_0 = \kappa(D^{-1}B)$, $D = \operatorname{diag}(B)$. (In our model problem with $p = 3$ κ_0 was reduced by a factor about $2/3$.)

The Case of Linear-Linear Basis Functions. The method presented in this paper can also be applied to $p = 1$ in the following way. Let the basic triangle consist of four uniform triangles of size $h/2$, see Figure 6.3. To the vertex nodes we associate the same linear basis functions as in the case $p = 2, 3$. To the remaining nodes we associate piecewise linear basis functions which are linear on each subtriangle, i.e., the standard linear basis functions corresponding to $h/2$.



$$\phi_1 = 1-x-y$$

$$\phi_2 = x$$

$$\phi_3 = y$$

$$\phi_4 = \begin{cases} 0 & (x,y) \in \mathrm{I} \\ 2x-2y-1 & (x,y) \in \mathrm{II} \\ 2y & (x,y) \in \mathrm{III} \\ 2x & (x,y) \in \mathrm{IV} \end{cases}$$

$$\phi_5 = \begin{cases} 2y & (x,y) \in \mathrm{I} \\ 1-2x & (x,y) \in \mathrm{II} \\ 0 & (x,y) \in \mathrm{III} \\ 2-2x-2y & (x,y) \in \mathrm{IV} \end{cases}$$

$$\phi_6 = \begin{cases} 2x & (x,y) \in \mathrm{I} \\ 1-2y & (x,y) \in \mathrm{II} \\ 2-2x-2y & (x,y) \in \mathrm{III} \\ 0 & (x,y) \in \mathrm{IV} \end{cases}$$

FIGURE 6.3
Basis functions for $p = 1$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

We get the following element stiffness matrix, where the entries are obtained by assembling over the four subelements;

$$\mathcal{A}_l=\frac{1}{6}\begin{bmatrix} 6 & -3 & -3 & -6 & 3 & 3 \\ -3 & 3 & 0 & 3 & -3 & 0 \\ -3 & 0 & 3 & 3 & 0 & -3 \\ -6 & 3 & 3 & 12 & -6 & -6 \\ 3 & -3 & 0 & -6 & 12 & 0 \\ 3 & 0 & -3 & -6 & 0 & 12 \end{bmatrix}.$$

We note that here $V_1 = V_h^{(1)}$, $V_2 \oplus V_1 = V_{h/2}^{(1)}$, $V_1 \cap V_2 = \{0\}$, and hence the method can be regarded as a two-level method for linear finite element approximations.

Comparisons of Various Diagonal and Full Block Preconditioning Methods. Since we have a uniform mesh and constant material coefficients, we can calculate γ and κ_0 from one single element matrix not meeting a Dirichlet boundary. These numbers are then also valid as upper bounds for elements at a Dirichlet boundary since then we consider subsets of nodes (basis functions).

Let λ_0 be the largest eigenvalue of B, and let $\kappa_0 = \kappa(D^{-1}B)$, $D = \text{diag}(B)$. We will consider some different preconditionings already discussed in Sections 4 and 5. These give rise to the methods $\mathcal{M}_i$, $i = 1,\dots,6$, with condition numbers κ_i, $i = 1,\dots,6$, as described in Table 6.1. In Table 6.2 we give values of γ and the corresponding bounds of κ_i, $i = 0,\dots,6$, calculated from the general expressions given in Sections 4 and 5, for our model problem and $p = 1,2,3$. κ_2 is calculated by direct use of (4.8).

TABLE 6.1
Various diagonal block (DB) *and full block* (FB) *preconditioning methods*

Method	DB/FB	$\tilde{B}$	$\tilde{A}$	Case in Section 5
1	DB	B	A	
2	DB	D	A	
3	FB	B	A	(ii)
4	FB	$\lambda_0 D$	$A-\lambda_0^{-1}CD^{-1}C^t$	(iv)
5	FB	$\lambda_0 D$	A	(iii)
6	FB	B	$A-\lambda_0^{-1}CD^{-1}C^t$	(v)

TABLE 6.2
The values of γ, κ_i, $i = 0,\dots,6$, *for our model problem with* $p = 1,2,3$

p	γ	κ_0	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6
1	.707	5.9	5.9	10.4	2	10.7	11.4	1.84
2	.816	5.9	9.9	15.2	3	15.5	16.9	2.7
3	.846	17.6	12.0	31.9	3.6	59.4	61.1	3.4

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

As was pointed out already above, the gain in the condition number by making a factorization of $A - \lambda_0^{-1}CD^{-1}C^t$ (methods $\mathcal{M}_4$, $\mathcal{M}_6$) instead of A (methods $\mathcal{M}_5$, $\mathcal{M}_3$) is minor. In all tests we got the same number of iterations for $\mathcal{M}_4$ and $\mathcal{M}_5$ as well as for $\mathcal{M}_6$ and $\mathcal{M}_3$. We also recall that we can come arbitrarily close to the results for exact factorizations of A by using accurate enough incomplete factorizations. Also note that the factorization work for A is relatively small compared to that for B and other arithmetic operations in the method, because the order of A is relatively small.

In Tables 6.3 and 6.4 we give the number of iterations needed in the conjugate gradient (CG) method to reduce the relative residual error by a factor $\varepsilon = 10^{-4}$ for various preconditioning methods and for different values of p and h. The iterations were stopped when $(\mathbf{r}^k, \mathbf{r}^k) \leqslant \varepsilon^2(\mathbf{r}^0, \mathbf{r}^0)$, where $\mathbf{r}^k = \mathcal{A}\mathbf{u}^k - \mathbf{f}$, $k = 0, 1, \ldots,$ and $\mathbf{u}^0 \equiv 0$. Later on we will also consider more accurate starting approximations. In the tables we indicate the methods $\mathcal{M}_i$, $i = 1, 2, 3, 5$ for which the analysis is made in Sections 4 and 5 and for which the bounds of the condition numbers are given in Table 6.2.

TABLE 6.3

The number of iterations for the diagonal block factorization methods for various (incomplete) factorizations of A and B and for p = 1, 2, 3, ε = 10^{-4} and different sizes of the mesh

p	h^{-1}	Exact fact. of B				B appr. by D				IC(0) of B
		Factorizations of A				Factorizations of A				MIC(4) of A
		MIC(0)	MIC(2)	MIC(4)	exact	MIC(0)	MIC(2)	MIC(4)	exact	
1	4	9	8	8	8	12	10	10	10	9
	8	10	9	8	8	13	11	10	10	9
	16	13	9	8	8	16	12	10	10	9
	32	18	10	9	8	21	14	11	10	9
2	4	10	9	9	9	13	11	11	12	10
	8	12	10	10	10	14	12	12	12	11
	16	15	10	10	10	17	13	12	12	11
	32	19	11	10	10	22	15	12	12	11
3	3	4	3	3	3	18	17	17	17	9
	6	11	10	10	10	19	18	18	18	13
	12	14	12	12	12	21	18	18	18	14
	24					24	18	18		

M_1 M_2

In the diagonal block method we also tried $\tilde{B} = \omega\,\text{diag}(B)$, $\omega \neq 1$, but it turned out that $\omega = 1$ is optimal (or close to optimal).

We note that when systems with the matrix B were solved by iteration in $\mathcal{M}_3$ (the method proposed by Bank and Dupont [3]), then 3 and 6 iterations were needed for $p = 2$ and 3, respectively, to yield the same number of outer iterations as in our method.

We see that if we use a sufficiently accurate incomplete factorization of A, the number of iterations stays the same as for the exact factorization of A for $h \geqslant h_0$. For instance, if $p = 2$, $h_0 = 1/32$, then MIC(4) is sufficient and if $p = 3$, $h_0 = 1/12$,

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

then MIC(2) is sufficient. By considering the total work i.e. the number of operations (multiply-adds) we find that (see also Table 6.5) among the diagonal block and full-block factorizations, respectively, those indicated by double lines in the tables are preferable. In the following these methods will be denoted diagonal block (DB) and full block (FB) factorization, respectively.

TABLE 6.4

The number of iterations for the full block factorization methods for various (incomplete) factorizations of A and B and for $p = 1, 2, 3$, $\varepsilon = 10^{-4}$ and different sizes of the mesh

p	h^{-1}	Exact fact. of B				IC(0) factorization of B				IC(-1) of B	B appr. by $\lambda_0 D$	
		Factorizations of A				Factorizations of A				MIC(4)	Fact. of A	
		MIC(0)	MIC(2)	MIC(4)	exact	MIC(0)	MIC(2)	MIC(4)	exact	of A	MIC(4)	exact
1	4	4	3	3	3	5	4	4	4	7	8	8
	8	4	3	3	3	6	4	4	4	7	8	8
	16	5	4	3	3	8	4	4	4	7	9	8
	32					10	5	4	4	7	9	8
2	4	4	3	3	3	6	5	5	5	8	10	10
	8	5	4	4	4	7	5	5	5	8	11	10
	16	6	4	4	4	8	5	5	5	8	11	10
	32					10	6	5	5	8	11	10
3	3	3	2	2	2	7	6	6	6	11	15	15
	6	6	5	5	5	8	7	7	7	12	18	18
	12	7	5	5	5	8	7	7	7	12	20	19

M_3 M_5

In the following table we give the computational complexity and storage requirement for these methods for $p = 1, 2, 3$, $h \geqslant h_0$ and $\varepsilon = 10^{-4}$. Note that in the case $p = 3$ the work estimate for the DB method is valid for $h_0 = 1/24$ as well. These values of h_0 are in most cases small enough to get a small enough discretization error because, as is well known, for smooth enough problem data the L_2-error of the solution is of order $O(h^{p+1})$, $h \to 0$. The work estimates include factorization work, and no consideration has been given to the fact that we have $\mathbf{u}^0 \equiv 0$ and that some elements in the matrices are zero because of the actual triangulation and problem data. Hence, these estimates are in principle also valid for more general (variable coefficient) problems (if the number of iterations stays the same). Within parentheses we also give the figures obtained if we *do* consider the number of zeros in the matrix.

TABLE 6.5

The work (W) *and storage* (S) *per unknown required for the* DB *and* FB *methods for* $p = 1, 2, 3$, $h \geqslant h_0$ *and* $\varepsilon = 10^{-4}$

Method	DB			FB		
P	1	2	3	1	2	3
h_0	1/32	1/32	1/24	1/32	1/32	1/12
W	240(220)	260(240)	420(390)	140(130)	180(170)	340(310)
S	7	7	6	9	9	10

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

When comparing these numbers one should bear in mind that the higher order methods ($p \geqslant 2$) give in general the same discretization errors as a lower order method ($p = 1$) for a much coarser grid. We comment further on this later in this section.

We note that the FB method needs less computational effort than the DB method, the difference, however, being relatively smaller for larger values of p. The DB method might sometimes be preferable because of its less need of storage and since it is simpler to implement. In this method we only need to assemble A (in order to make the incomplete factorization) and the diagonal of B. As is well known,one may calculate the product $\mathscr{A}\cdot\mathbf{x}$, needed in the CG method, from the element matrices without having to assemble and store the global matrices.

For the storage requirements in Table 6.5 we have assumed that $\mathscr{A}\cdot\mathbf{x}$ is calculated in this way. In our model problem we have only one element matrix and in more general problems one may have only a small number of different element matrices in which case the storage requirement will be only slightly greater than that given in Table 6.5.

In Figure 6.4 we have drawn the number of iterations as a function of h^{-1} for the full block method, $p = 2$ with IC(0) factorization of B and different (approximate) factorizations of A. The discrete behavior of the number of iterations has been smoothed out by calculating (the real number) $\tilde{k} = k \ln \varepsilon^2 / \ln\{(\mathbf{r}^k, \mathbf{r}^k)/(\mathbf{r}^0, \mathbf{r}^0)\}$ when the iterations have been terminated for $(\mathbf{r}^k, \mathbf{r}^k) \leqslant \varepsilon^2 (\mathbf{r}^0, \mathbf{r}^0)$. The figure illustrates how the point, where the $O(h^{-1/2})$ behavior of the number of iterations comes into effect, depends on the degree of accuracy of the incomplete factorization. The scale is logarithmic and the slope of the line $\tilde{k} = h^{-1/2}$ is indicated.

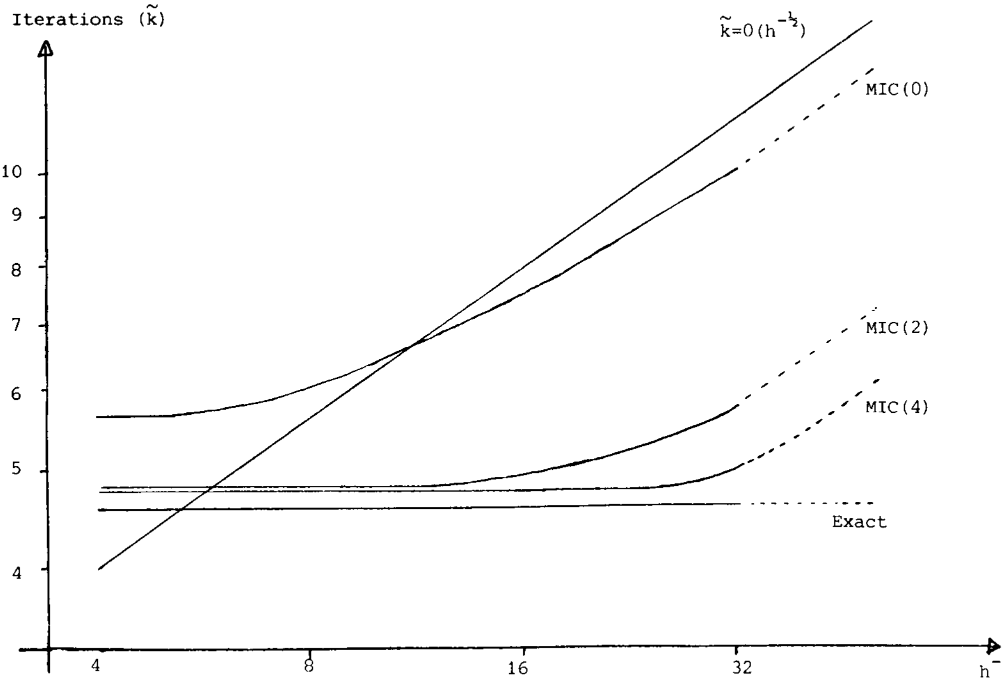


FIGURE 6.4

The number of iterations for the FB *method*, $p = 2$, $\varepsilon = 10^{-4}$ *as a function of* h^{-1} *for different (approximate) factorizations of* A

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

More General Test Problems. We have also tested problem (6.1) with discontinuous material coefficient; $a \equiv 1$ for $x \leq \frac{1}{2}$ and $a \equiv d$, $x > \frac{1}{2}$. The number of iterations depends only slightly on d, see Table 6.6. We note that the estimated values of the condition numbers in Table 6.2 are valid also in this case.

TABLE 6.6

The number of iterations for the FB *and* DB *methods for* $p = 2$, $\varepsilon = 10^{-4}$, $h \in [1/32, 1/4]$ *for different values of the discontinuity parameter d*

method \ d	1	10	100	1000
FB	5	6	6	6
DB	12	13	14	14

For smoothly varying coefficient a we expect the same or almost the same rate of convergence as for $a \equiv 1$. For e.g. $a = 1 + x^2 + y^2$, $p = 2$, $h = 1/8$ we namely find (by actually computing these values) that the change in κ_0 and γ from the case $a \equiv 1$ is only 1.2 and 0.07 percent, respectively. Obviously, for smaller values of h the change is even smaller.

Furthermore, we have obtained the same or almost the same rate of convergence for unisotropic problems. Even in this case the derived estimates of the condition numbers (in particular also of γ) hold.

Work Estimates. Let us now compare the work needed in a model problem to obtain a desired accuracy for $p = 1$ and $p = 2$. If the solution u is smooth (i.e. if the problem data f, g, Ω is smooth) then the errors in the L_2-norm of the solution is of order $O(h^{p+1})$, $h \to 0$. Hence the number of unknowns $N^{(p)}$, $p = 1, 2$ (needed to get a discretization error less than ε) are related by

$$N^{(2)} \simeq C(N^{(1)})^{2/3}. \tag{6.2}$$

We consider the problem (6.1) with f chosen such that $u = (1 - x)^2x^2(1 - y^2)y^2$. Then for $\varepsilon = .3 \cdot 10^{-4}$, $N^{(1)} = 225$ $(h = 1/16)$ and $N^{(2)} = 49$ $(h = 1/4)$ nodes were required, respectively.

To solve this problem with $p = 2$ by the FB method we also consider the task of choosing a good starting approximation. To this end we solve the problem with $p = 1$ on a coarser grid $(h = \frac{1}{2})$ by the preconditioned CG method (or by recursive use of the method described in this paper similarly to the multigrid method). The obtained solution is linearly interpolated to the finer mesh to yield a starting approximation for the iterations on this finer mesh.

This latter idea is used in [2] and [6]. We note that we obtain the solution on the coarser mesh by solving iteratively a system with matrix *A for which we already have made an incomplete factorization.* This system does not have to be solved to excessively high accuracy, often only a couple of iterations suffice. In our test problem one iteration (in fact the incomplete factorization is exact for this small system) was needed on the coarser mesh to obtain the starting approximation and then only two iterations were needed on the finer mesh to get a total error of the same size as the discretization error (say two times the discretization error). This corresponds to an operation count of about $78N^{(2)} \approx 17N^{(1)}$ operations. This work estimate should be used in comparisons with methods using finite differences or

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

linear finite elements. Because of the relation (6.2) we expect even better work estimates (related to $N^{(1)}$) for smaller discretization errors i.e. for larger values of $N^{(1)}$. In our test example $\varepsilon = 2 \cdot 10^{-5}$ required $N^{(1)} = 3969$ ($h = 1/64$) and $N^{(2)} = 361$ ($h = 1/10$), respectively. In this case three iterations were needed on the coarser and finer meshes and we got a computational cost of about $126N^{(2)} \approx 11.5N^{(1)}$ operations.

The Factor $\log N$. In connection with working on two level grids we remark that the factor $\log N$ in the work estimate in Section 1 can be dispensed of in the following well-known way.

Assume that we first solve the problem on the coarser mesh Ω_h to an accuracy of $O(h^{p+1})$, i.e. to an accuracy of the same order as the discretization error on this level. We then use interpolation of order p to get an initial approximation on the finer grid $\Omega_{(h/k)}$, $k \geqslant 2$ an integer. (This is achieved by simply evaluating the finite element solution onto the new mesh points.) Now we only have to achieve a relative accuracy of order $(1/k)^{p+1}$ (i.e. independent on h) to get a final error of the size of the discretization error on the finer mesh. Hence in the estimate for this number of iterations, no factor $\log N$ appears.

Initially, on the coarser mesh we may get such an accurate solution by recursively working on coarser and coarser grids, but as we already have pointed out (this being one of the main points of this paper), because of the small size of the problem on the coarser meshes we eventually may solve the system exactly or by iteration to almost full accuracy. In fact in the work estimates for the multigrid methods one often assumes that the solution on the coarser grid is already given.

7. Extensions, Conclusions and Comparisons With Other Methods. The methods presented in this paper are readily generalized to other types of meshes e.g. quadrilateral meshes. For bilinear f.e. based on a rectangular mesh division of the model problem we expect an even faster rate of convergence since calculated values of γ, κ_i, $i = 0,\ldots,6$, are smaller than if we have a triangular mesh. Actually, the derived estimates are even better than in the *linear* ($p = 1$) triangular case, so the number of iterations would be smaller than what we get in that case; see Section 6.

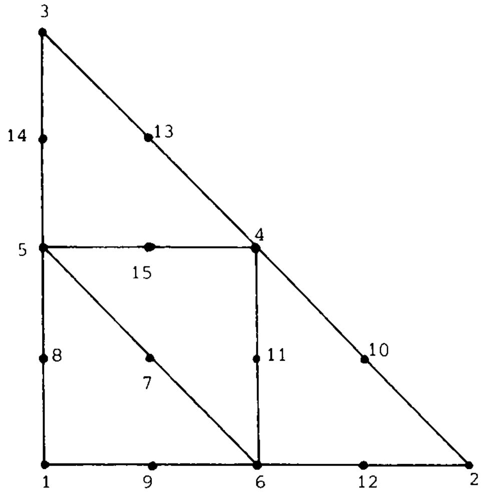


FIGURE 7.1
Element in a three-level method, $p = 2$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

As already pointed out our method can be regarded as a two-level method. The idea can be generalized to a multi-level method; see also [3]. For simplicity, we consider a three-level approach to the case $p = 2$ and a right-angled triangulation, where each element consists of four small elements, see Figure 7.1.

With the vertex nodes $(1, 2, 3)$ we associate linear functions (with support on the whole element) and with the midpoints of the edges $(4, 5, 6)$ we associate piecewise linear functions being linear over each subtriangle. (This corresponds to the case $p = 1$ in Section 6.) With the remaining nodes we associate piecewise quadratic basis functions, the standard Lagrangian basis functions of degree $p = 2$ corresponding to $h/2$. We get a stiffness matrix with the structure

$$\mathcal{A} = \left[\begin{array}{c|cc} A_1 & C_1 & C_2 \\ \hline C_1^t & A_2 & C_3 \\ C_2^t & C_3^t & A_3 \end{array}\right] \quad \text{and let} \quad \mathcal{C}_1 = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & C_3 \\ 0 & C_3^t & A_3 \end{bmatrix},$$

$$\mathcal{C}_2 = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \quad \text{be diagonal block preconditionings to } \mathcal{A}.$$

In a similar way as in the two-level case we get that $\kappa(A_1) = O(h^{-2})$, $h \to 0$, $\kappa(A_2) = O(1)$, $h \to 0$, $\kappa(A_3) = O(1)$, $h \to 0$ and order $(A_1) = O(N/16)$ (in general $O\{N/(p2^{l-2})^d\}$ in an l-level method in d dimensions). Apparently, this will lead to a smaller h_0 (compared to the same degree of accuracy of the incomplete factorization of A_1) than in the two-level method. However, γ (corresponding to the indicated block-partitioning of $\mathcal{A}$) and $\kappa(\mathcal{C}_1^{-1}\mathcal{A})$ are larger than in the two-level method, about .850 compared to 0.816 and 13.7 compared to 5.9, respectively. If we consider the three-block diagonal preconditioning $\mathcal{C}_2$, we get even larger values. Furthermore, the matrix is more dense due to the fact that basis functions associated to vertex nodes have larger support. We conclude that in the approach we use here it is *not* preferable to use more than two levels. At this point we remark that the work involving A, i.e. the work on the coarser mesh, is minor compared to the entire work. For instance, for the DB method, $p = 2$, this work amounts to only about $1/6$ of the total work to solve the model problem.

The computational complexity for the DB method is comparable with that for the method based on spectral equivalence presented in [2] for $p = 2$ and $N = 1000$. For $N = 4000$ the DB method is about 25% faster. For $p = 1$, the methods presented in this paper are slower than MICCG methods based on standard f.e., unless h is excessively small. This is so because the matrix is more dense,due to the fact that the basis functions have larger support.

If in the diagonal block method the systems of equations with matrices A and B are solved by a direct method (Gaussian-elimination) and by iteration, respectively, the work estimate is more than 600 operations per unknown for $p = 2$, $N = 4000$ and more than 1000 operations per unknown for $p = 3$, $N = 1200$. Hence we have reduced the work by a factor of about 0.4 by using *incomplete* factorizations; see Table 6.5. The storage requirements are more than halved. An alternative to our method might be to solve the system with matrix A *approximately* by some other method e.g. a multigrid method.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

For three-dimensional problems we expect the new method to be even more competitive than other methods because in a d-dimensional problem the order of the matrix A is only $O(N/p^d)$, where N is the number of unknowns.

We conclude that we have derived a class of methods having complexity in arithmetic operations and storage effectively independent of h for $h \geqslant h_0$, where h_0 is sufficiently small to cover most applications. Compared with other iterative and direct methods, the methods are highly competitive with respect to computational cost as well as to storage requirement.

The efficiency of the method is comparable to the best implementations of multigrid methods for solving model problems [7]. Our method is however applicable to more general problems with no or a small increase only in work estimates and avoids the problem of working with several levels. The derived upper bounds for the computational cost are valid also for discontinuous, unisotropic and smoothly varying material coefficients.

The rate of convergence of the usual multigrid methods seems to be much more sensitive to variable and/or unisotropic coefficients and to general domains. In [10] it is reported that 25 to 80 operations per mesh point are needed in various implementations of the multigrid method but the actual computing time for general domains was increased by a factor of 4 to 5 compared to the model problem on the unit square. In general, overhead operations seem to contribute to a large portion of the computing time for the multigrid method on general domains, whereas this matters little in our method.

We also remark that, if one examines multigrid methods applied to a fixed number of grid levels, one finds that the method can be formulated in terms of a preconditioned iterative method.

To summarize our arguments of this slightly lengthy paper we claim that, in practice, in the multigrid method one works with few levels of grids. Then one might as well consider simpler iterative methods which are also more suitable for general (high order) finite element methods and which on actually mostly used meshes and domains gives about the same computer times or, at least if $p > 1$, much smaller computer times in order to calculate a solution to a given order of accuracy. Such a method, a two-level preconditioned conjugate gradient method, has been presented in this paper. The method is also highly competitive to earlier similar methods of preconditioned conjugate gradient type. Finally, it is easy to program and is well studied for implementations in existing software for the finite element method.

Mathematisch Instituut
Katholieke Universiteit
Nijmegen, The Netherlands

Department of Computer Science
Chalmers University of Technology
Göteborg, Sweden

1. O. Axelsson, "A class of iterative methods for finite element equations," *Comput. Methods Appl. Mech. Engrg.*, v. 9, 1976, pp. 123–137.

2. O. Axelsson & I. Gustafsson, *A Preconditioned Conjugate Gradient Method for Finite Element Equations, Which is Stable for Rounding Errors*, Information Processing 80 (S. H. Lavington, ed.), North-Holland, Amsterdam, 1980, pp. 723–728.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

3. R. BANK & T. DUPONT, *Analysis of a Two-Level Scheme for Solving Finite Element Equations*, Report CNA-159, Center for Numerical Analysis, The University of Texas at Austin, 1980.

4. A. BRANDT, "Multi-level adaptive solutions to boundary value problems," *Math. Comp.*, v. 31, 1977, pp. 333–390.

5. I. FRIED, "Bounds on the extremal eigenvalues of the finite element stiffness and mass matrices and their spectral condition numbers," *J. Sound Vibration*, v. 22, 1972, pp. 407–418.

6. I. GUSTAFSSON, *Stability and Rate of Convergence of Modified Incomplete Cholesky Factorization Methods*, Thesis, Report 79.02R, Department of Computer Sciences, Chalmers University of Technology, Göteborg, Sweden, 1979.

7. P. W. HEMKER, "Introduction to multigrid methods," *Colloquium Numerical Solution of Partial Differential Equations* (J. G. Verwer, ed.), MC SYLLABUS 44, Mathematisch Centrum, Amsterdam, 1980, pp. 59–67.

8. D. KERSHAW, "The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations," *J. Comput. Phys.*, v. 26, 1978, pp. 43–65.

9. 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. Comp.*, v. 31, 1977, pp. 148–162.

10. U. TROTTENBERG, Private communication, 1981.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use