BIT 1997

A CHARACTERIZATION OF MAPPING UNSTRUCTURED GRIDS ONTO STRUCTURED GRIDS AND USING MULTIGRID AS A PRECONDITIONER

CRAIG C. DOUGLAS[1], SACHIT MALHOTRA[2], and MARTIN H. SCHULTZ[1*]

February 12, 1997

[1]*Department of Mathematics, University of Kentucky*
715 Patterson Office Tower, Lexington, KY 40506–0027, USA
and Department of Computer Science, Yale University
P.O. Box 208285, New Haven, CT 06520-8285, USA.
email: douglas@ccs.uky.edu

[2]*Scientific Computing Associates*
One Century Tower, 265 Church Street, New Haven, CT 06510-7010, USA.
email: malhotra@sca.com

[3]*Department of Computer Science, Yale University*
P.O. Box 208285, New Haven, CT 06520-8285, USA.
email: schultz-martin@cs.yale.edu

Abstract.

Many problems based on unstructured grids provide a natural multigrid framework due to using an adaptive gridding procedure. When the grids are saved, even starting from just a fine grid problem poses no serious theoretical difficulties in applying multigrid.

A more difficult case occurs when a highly unstructured grid problem is to be solved with no hints how the grid was produced. Here, there may be no natural multigrid structure and applying such a solver may be quite difficult to do.

Since unstructured grids play a vital role in scientific computing, many modifications have been proposed in order to apply a fast, robust multigrid solver. One suggested solution is to map the unstructured grid onto a structured grid and then apply multigrid to a sequence of structured grids as a preconditioner.

In this paper, we derive both general upper and lower bounds on the condition number of this procedure in terms of computable grid parameters.

We provide examples to illuminate when this preconditioner is a useful (e.g., p or h-p formulated finite element problems on semi-structured grids) or should be avoided (e.g., typical computational fluid dynamics (CFD) or boundary layer problems). We show that unless great care is taken, this mapping can lead to a system with a high condition number which eliminates the advantage of the multigrid method.

AMS subject classification: 65F20.

Key words: Multigrid, unstructured grids, iterative methods.

---

*This work was partially supported by ONR Grant # N0014-91-J-1576.

1 Introduction.

Multigrid methods are one of the most efficient approaches to the solution of partial differential equations. These methods are critically dependent on the presence of natural multilevel structures in the domain. A well known folk theorem is the following: In the case of unstructured grids multigrid methods are not, in general, conveniently applicable since this natural structure is not available.

In [7], multigrid for CFD problems based on unstructured grids is described for a particular class of problems, namely, the simulation of airflow over a foil. Multigrid on unstructured grids is reported to be only a factor of 3 slower than on a highly structured grid on a vector supercomputer. This is based on years of work at NASA to make the multigrids code run fast on this class of computer.

An approach suggested in [10] is the creation of a single *auxiliary* grid to which the error can be transferred from the unstructured grid. This is the first approach anyone implementing domain decomposition methods would think of, possibly resulting in locally uniform meshes with small mesh spacings.

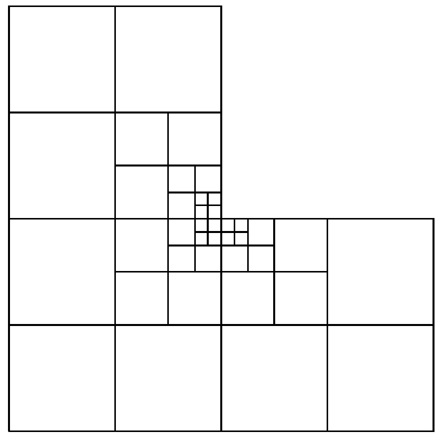
The auxiliary grid is chosen so that it has a natural multigrid structure. Although the multigrid preconditioning approach does seem reasonable, we show that this construction requires great care if the expected fast convergence rate of the multigrid method is to be preserved.

Figure 1.1 contains two grids on a L-shaped domain. Figure 1.1(a) shows a graded grid by the re-entrant corner (multigrid archeologists [8, p. ix] will recognize this as an Achi Brandt example [3]). If the mesh spacings are h, $h/2$, $\cdot$, $2^{1-k}h$, for some $k > 1$, then there are only $9k + 3$ elements associated with the graded grid. However, there are 3×4^k elements associated with the uniform grid in Figure 1.1(b). Even if k is fairly small, the uniform grid has an intolerable number of extra elements with respect to the graded grid.

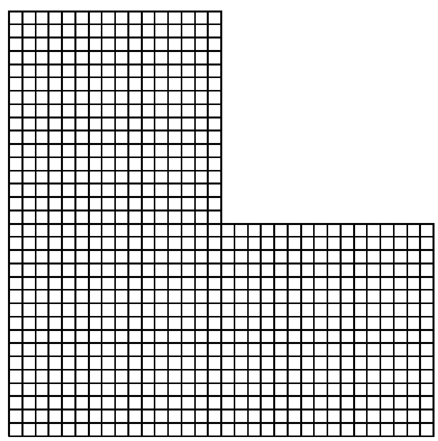
In fact, in the case of badly matched grids, the condition number of the transformed problem can be extremely large. This is solely due to interpolation errors when mapping data between grids. The error in transferring the residual onto the auxiliary grid can eliminate the advantages of fast convergence on the uniform grid.

In [10], a general method is developed and analyzed. All of the examples contained there assume a nearly structured grid which is mapped onto a similar, but structured grid. The primary theorem in [10] requires that certain expressions be constants. For general problems (even many quite simple ones), these expressions are not all constants. In this paper, we relax certain assumptions slightly to allow much more general problems to be analyzed.

Some of the examples in [10] are usually solved on the multigrid grids using lower order basis functions than the original problem. This reduces the condition number of the preconditioned system and would in practice reduce the complexity for the running time of the preconditioner. This suggests that this preconditioner is particularly useful for problems with a p or h-p finite element formulation on semi-structured grids. On vector computers, the preconditioner would have advantages; on RISC based machines multigrid already works this



(a) Graded grid with $9k+3$ elements (mesh spacings h, $h/2$, $\cdot$, $2^{1-k}h$)



(b) Uniform grid with $3{\times}4^k$ elements

Figure 1.1: L shaped domain

class of problems well.

An alternate approach is taken by Brenner in [4]. Optimal preconditioners are produced using the same underlying grid. For complicated finite elements, lower order, less complicated finite elements that are related to the original ones are used. The problems raised in this paper do not appear simply as a result of not changing the grid.

In this paper, we see that for problems with low order basis functions and unstructured grids, the preconditioner really is not practical. This suggests that for many CFD problems (e.g., air flow across a foil, reservoir simulation, pollution tracking, or combustion simulation) which already use as low order basis functions as makes sense and highly unstructured grids, that this preconditioner is highly unsuitable.

In §2, we develop an algorithm and provide both upper and lower bounds on the condition number of the preconditioned problem. In §3, we provide three examples demonstrating when to use or avoid this preconditioner. In §4, we provide numerical experiments based on the examples of §3. In §5, we draw some concrete conclusions about the usefulness of this preconditioner.

2 Auxiliary space method: an abstract framework.

In this section, we follow the exposition and notation of [10] closely. The method suggested is a general scheme based on a relaxation scheme and an auxiliary space. The method is normally a two level nonnested multigrid preconditioner.

Assume that A is a symmetric, positive definite operator on a linear inner product space $\mathcal{V}$ with respect to an inner product $(\cdot,\cdot)$. Consider the following set of linear equations:

$$Ax = y,$$

which we assume was derived by discretizing a boundary value problem.

The suggested method uses an auxiliary linear product space $\mathcal{V}_0$ together with the operator A_0 that is symmetric positive definite with respect to an inner product $[\cdot,\cdot]$. The space $\mathcal{V}_0$ is an approximation of $\mathcal{V}$ though not usually a subspace. The operator A_0 may be viewed as the approximation of A in $\mathcal{V}_0$. This, in turn, is preconditioned by another symmetric positive definite operator $B_0 : \mathcal{V}_0 \rightarrow \mathcal{V}_0$.

We also need a projection operator $\Pi^t : \mathcal{V} \rightarrow \mathcal{V}_0$. The operator Π plays the role of the interpolation operator in the multigrid method. The smoother on $\mathcal{V}$ is denoted by R. The proposed preconditioner in this method is given by

$$B = R + \Pi B_0 \Pi^t. \tag{2.1}$$

With each grid $\mathcal{V}$ we associate a constant h which captures the spacing of the points in the grid. We also note that for any $u, v \in \mathcal{V}_h$,

$$(BAu, v)_A = (RAu, v)_A + [B_0 A_0 \Pi^* u, \Pi^* v]_{A_0}, \tag{2.2}$$

where $(\cdot,\cdot)_A = (A\cdot,\cdot)$, $[\cdot,\cdot]_{A_0} = [A_0\cdot,\cdot]$, $\Pi^* = A_0^{-1}\Pi^t A$ satisfies

$$[\Pi^* v, w]_{A_0} = (v, \Pi w)_A, \qquad v \in \mathcal{V}, w \in \mathcal{V}_0.$$

Let A have eigenvalues $\lambda_A^1 \leq \lambda_A^2 \leq \ldots \leq \lambda_A^N = \rho_A$ with corresponding eigenvectors $v_A^1, \ldots, v_A^N$.

We will now show that the transformation of the grid as suggested entails a degradation in the conditioning of the problem. It is the magnitude of the condition number of this transformation that actually governs the efficacy of the procedure.

In Theorem 2.1 we make a number of technical assumptions which relate to the properties of the grids $\mathcal{V}$ and $\mathcal{V}_0$. Assumption (2.3) deals with the relaxation operator R and implies that the operator actually reduces the norm of the error on the original grid. Assumption (2.4) implies that B_0 is an effective preconditioner on the auxiliary grid, i.e., it has a bounded condition number. Assumptions (2.5) and (2.6) imply that the norm of a vector do not change by more than a constant factor in going from one grid to the other. Assumption (2.7) is related to the error in a particular vector when transferring data from one grid to the other and back.

THEOREM 2.1. *Assume there are nonnegative constants $\alpha_0(h)$, $\alpha_1(h)$, λ_0, λ_1 and $\beta_1(h)$ such that for all $v \in \mathcal{V}$ and $w \in \mathcal{V}_0$,*

$$\alpha_0(h)\rho_A^{-1}(v,v) \le (Rv,v) \le \alpha_1(h)\rho_A^{-1}(v,v), \tag{2.3}$$

$$\lambda_0[w,w]_{A_0} \le [B_0A_0w,w]_{A_0} \le \lambda_1[w,w]_{A_0}, \tag{2.4}$$

and

$$\|\Pi w\|_A^2 \le \beta_1(h)\|w\|_{A_0}^2. \tag{2.5}$$

Furthermore assume there exists a linear operator $P : \mathcal{V} \to \mathcal{V}_0$, and positive constants $\beta_0(h)$ and $\gamma_0(h)$ such that

$$\|Pv\|_{A_0}^2 \le \beta_0(h)^{-1}\,\|v\|_A^2 \tag{2.6}$$

and

$$\|v - \Pi Pv\|^2 \le \gamma_0(h)^{-1}\rho_A^{-1}\|v\|_A^2. \tag{2.7}$$

Then the preconditioner given by (2.1) satisfies

$$\kappa(BA) \le (\alpha_1(h) + \beta_1(h)\lambda_1)((\alpha_0(h)\gamma_0(h))^{-1} + (\beta_0(h)\lambda_0)^{-1}) \tag{2.8}$$

and

$$\kappa(BA) \ge (\alpha_0(h) - \lambda_1\beta_1(h))(\alpha_1(h)\rho_A^{-1}\lambda_A^1 + \lambda_1\beta_1(h))^{-1}. \tag{2.9}$$

PROOF. See [10, Thm 2.1] for a proof of (2.8). Now we prove (2.9), which is new. From (2.2),

$$\begin{aligned} (BAv,v)_A &= (RAv,v)_A + [B_0A_0\Pi^*v,\Pi^*v]_{A_0} \\ &\ge |(RAv,v)_A| - |[B_0A_0\Pi^*v,\Pi^*v]_{A_0}| \\ &\ge \alpha_0(h)\rho_A^{-1}\|Av\|^2 - \lambda_1\|\Pi^*v\|_{A_0}^2 \\ &\ge \alpha_0(h)\rho_A^{-1}\|Av\|^2 - \lambda_1\beta_1(h)\|v\|_A^2. \end{aligned}$$

For $v = v_A^N$ we have

$$|(BAv,v)_A| = \lambda_A^N|(BAv,v)| \ge \alpha_0(h)\rho_A^{-1}(\lambda_A^N)^2(v,v) - \lambda_1\beta_1(h)\lambda_A^N(v,v).$$

This implies

$$(BAv_A^N, v_A^N) \ge (\alpha_0(h) - \lambda_1\beta_1(h))(v_A^N, v_A^N)$$

or
$$\lambda_{\max}(BA) \geq (\alpha_0(h) - \lambda_1\beta_1(h)).$$
From (2.2),
$$\begin{aligned}(BAv, v)_A &\leq |(RAv, Av)| + \lambda_1\|\Pi^* v\|^2_{A_0}\\ &\leq \alpha_1(h)\rho_A^{-1}\|Av\|^2 + \lambda_1\|\Pi^* v\|^2_{A_0}.\end{aligned}$$

For $v = v_A^1$ we have,
$$|(BAv_A^1, v_A^1)_A| = \lambda_A^1(BAv_A^1, v_A^1) \leq \alpha_1(h)\rho_A^{-1}(\lambda_A^1)^2(v_A^1, v_A^1) + \lambda_1\beta_1(h)\|v_A^1\|_A^2$$
or
$$|(BAv_A^1, v_A^1)| \leq \frac{1}{\lambda_A^1}[\alpha_1(h)\rho_A^{-1}(\lambda_A^1)^2 + \lambda_1\beta_1(h)\lambda_A^1](v, v).$$
This implies that
$$\lambda_{\min}(BA) \leq (\alpha_1(h)\rho_A^{-1}\lambda_A^1 + \lambda_1\beta_1(h)).$$
Hence, (2.9) follows. □

As can be seen from §3 there are a number of cases where the assumptions of the above theorem are violated. In particular, for many cases of practical interest, the parameters α_0, α_1, β_1 and γ_0 are not constants but depend on the structure of the two grids $\mathcal{V}$ and $\mathcal{V}_0$.

3 Examples.

In this section, we investigate three examples. The first two (§3.1 and §3.2) examples are in one and two space dimensions. They use geometrically chosen grids which are mapped onto uniform grids. The preconditioner is shown to be unhelpful in §3.3. The third example (§3.4) shows a case in which the method is useful. It has a nearly uniform grid that is mapped onto a similar uniform grid.

For clarity of presentation, we drop the explicit dependence of the various parameters on h. For all the discussion in this section it should be assumed that the parameters are functions of h (i.e., α_0 refers to $\alpha_0(h)$ and so on).

3.1 1-D example with a geometric grid.

The effectiveness of the above method is clearly dependent on the value of the constants and the form of the function f. Consider a simple example to show that an inappropriate selection of $\mathcal{V}_0$ can lead to a preconditioner that is far from optimal.

Consider the simple two point boundary value problem defined on the interval $[0, 1]$:

(3.1) $$-u''(x) = g(x); \qquad u(0) = u(1) = 0.$$

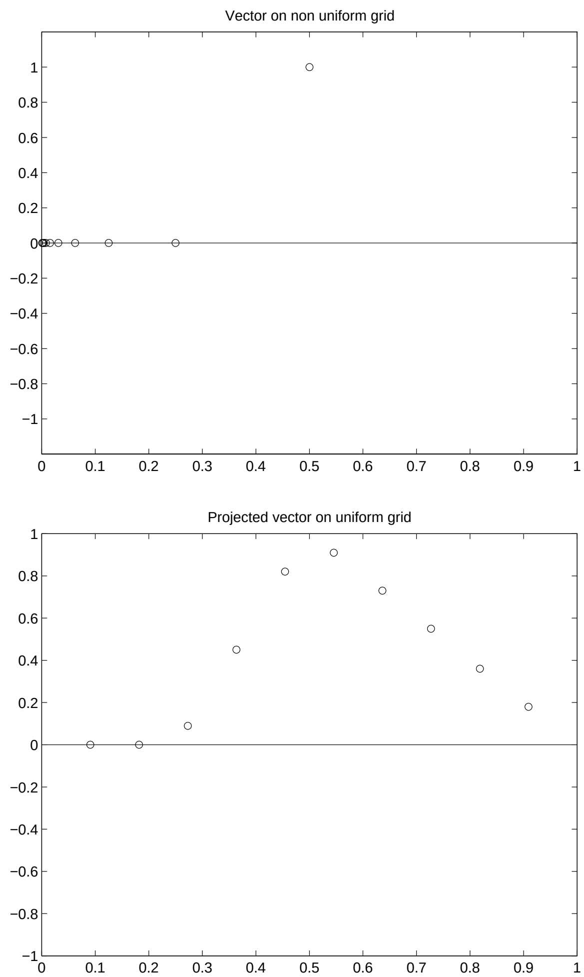


Figure 3.1: Vector on $\mathcal{V}$ and $\mathcal{V}_0$ for k=10

$\mathcal{V}$ here is the non-uniform grid given by

$$\mathcal{V} = \left(\frac{2^0}{2^k}, \frac{2^1}{2^k}, \ldots, \frac{2^{k-1}}{2^k} \right). \tag{3.2}$$

This has dimensionality k. This type of grid is common for boundary layer problems. Consider the case when the auxiliary grid $\mathcal{V}_0$ is chosen to be the uniformly spaced grid having the same dimensionality on $[0, 1]$:

$$\mathcal{V}_0 = \left(\frac{1}{k+1}, \frac{2}{k+1}, \ldots, \frac{k}{k+1} \right).$$

The operator A in this case is given by a tridiagonal matrix with entries

$$a_{i,j} = \begin{cases} 0, & if \quad |i-j| > 1, \\ 2^{2k+1}, & if \quad i = j = 1, \\ 2^{2k-2i+4}, & if \quad i = j \neq 1, \\ -\frac{1}{3} 2^{2k-2i+5}, & if \quad j = i-1, \\ -\frac{1}{3} 2^{2k-2i+4}, & if \quad j = i+1. \end{cases} \tag{3.3}$$

The corresponding operator A_0 on $\mathcal{V}_0$ is given by a tridiagonal matrix A_0 with entries

$$a^0_{i,j} = \begin{cases} 0, & if \quad |i-j| > 1, \\ 2\,(k+1)^2, & if \quad i = j, \\ -\,(k+1)^2, & if \quad |i-j| = 1. \end{cases}$$

We use simple linear interpolation to map between the two domains. We now proceed to show that the function f depends on the value of k. This in turn implies that the condition number of the preconditioned matrix is $O(k)$.

A simple application of Gershgorin's Disk theorem shows that $\rho_A = O(2^{2k})$. We now try to evaluate the various constants in the theorem for this particular $\mathcal{V}$ and $\mathcal{V}_0$. We assume that Jacobi is used as a smoother. Although this eases the computation of the constants, the argument is not dependent of the form of the smoother. A calculation of the spectral radius of the Jacobi matrix provides an estimate of the values of α_0 and α_1:

$$\begin{aligned} \alpha_0 &= O\left(2^{2k}/k\right) \\ \alpha_1 &= O\left(2^{2k}\right). \end{aligned}$$

Since the preconditioner B_0 is derived from the application of the multigrid method on a uniform grid we know that

$$\lambda_0 = \lambda_1 = O(1).$$

An analysis of the two grids also reveals that

$$\beta_0^{-1} = O(1)$$

and

$$\beta_1 = O\left(\frac{1}{k^2}\right).$$

Similarly we see that

$$\gamma_0^{-1} = O(\rho_A) = O(2^{2k}).$$

Consider the vector $v \in \mathcal{V}$ such that $v = e_k$. Since the last element of the vector is 1 we can explicitly compute $\|v\|_A^2$. A simple calculation shows that

$$\|v\|_A^2 = (Ae_k, e_k) = 16.$$

Now look at the vector $w = Pv$, which is given by

$$w_i = \begin{cases} 0, & if \quad i < \frac{k+1}{4}, \\ 1 - \frac{2}{k+1} + \frac{4i}{k+1}, & if \quad \frac{k+1}{4} \le i < \frac{k+1}{2}, \\ 1, & if \quad i = \frac{k+1}{2}, \\ 2 - \frac{2i}{k+1}, & if \quad i > \frac{k+1}{2}. \end{cases}$$

See Figure 3.1 for an illustration of this. A similar computation shows that

$$\|Pv\|_{A_0}^2 = \|w\|_{A_0}^2 = (A_0 w, w) = 6(k+1).$$

This implies that $f(h) = f(\frac{1}{k}) = o(k)$. As a result the function f grows at least as fast as k.

3.2 2-D example with a geometric grid

We now do a similar analysis for the problem in two dimensions. Consider the model problem defined on the square $[0,1]\times[0,1]$:

$$-u_{xx} - u_{yy} = g(x); \qquad u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0.$$

The non-uniform grid in this case is given by $\mathcal{V} = \mathcal{M} \times \mathcal{M}$, where

$$\mathcal{M} = \left(\frac{2^0}{2^k}, \frac{2^1}{2^k}, \ldots, \frac{2^{k-1}}{2^k} \right).$$

This has dimensionality k^2.

Now consider the problem where the auxiliary grid $\mathcal{V}_0$ is chosen to be the uniformly spaced grid having the same dimensionality on $[0,1] \times [0,1]$, i.e., $\mathcal{V}_0 = \mathcal{M}_0 \times \mathcal{M}_0$ where

$$\mathcal{M}_0 = \left(\frac{1}{k+1}, \frac{2}{k+1}, \ldots, \frac{k}{k+1} \right).$$

Figure 3.2 contains an example of $\mathcal{V}$ and $\mathcal{V}_0$ when $k = 10$.

For each point (i,j) in both grids, define the function

$$index(i,j) = (i-1)k + j$$

and the inverse mappings

$$X(index) = \left\lfloor \frac{index}{k} \right\rfloor + 1$$

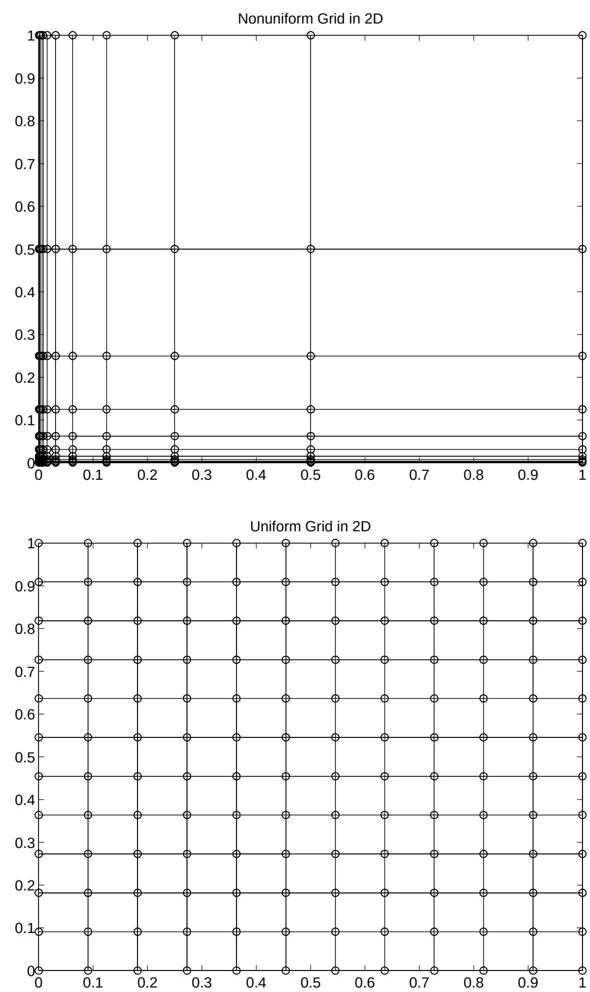


Figure 3.2: $\mathcal{V}$ and $\mathcal{V}_0$ in the 2D case, k=10

and

$$Y(index) = index - (X(index) - 1)k.$$

For each point p in $\mathcal{V}$, define the following constants:

$$\begin{aligned} h_l(p) &= \begin{cases} 2^{-k}, & if \quad X(p) = i = 1, \\ 2^{i-2-k}, & if \quad X(p) = i \neq 1, \end{cases} \\ h_r(p) &= \quad 2^{i-1-k}, \qquad X(p) = i, \end{aligned}$$

$$
\begin{aligned}
h_t(p) &= \begin{cases} 2^{-k}, & if \quad Y(p) = j = 1, \\ 2^{j-2-k}, & if \quad Y(p) = j \neq 1, \end{cases} \\
h_b(p) &= \quad 2^{j-1-k}, \qquad Y(p) = j.
\end{aligned}
$$

The operator A in this case is given by a pentadiagonal matrix of size $k^2 \times k^2$ with entries

$$
a_{p_1,p_2} = \begin{cases}
\frac{2}{h_l(p_1)h_r(p_1)} + \frac{2}{h_t(p_1)h_b(p_1)}, & if \quad X(p_1) = X(p_2) \quad and \quad Y(p_1) = Y(p_2), \\
\frac{-2}{h_l(p_1)(h_l(p_1)+h_r(p_1))}, & if \quad X(p_2) = X(p_1) - 1 \quad and \quad Y(p_1) = Y(p_2), \\
\frac{-2}{h_r(p_1)(h_l(p_1)+h_r(p_1))}, & if \quad X(p_2) = X(p_1) + 1 \quad and \quad Y(p_1) = Y(p_2), \\
\frac{-2}{h_t(p_1)(h_t(p_1)+h_b(p_1))}, & if \quad Y(p_2) = Y(p_1) - 1 \quad and \quad X(p_1) = X(p_2), \\
\frac{-2}{h_b(p_1)(h_t(p_1)+h_b(p_1))}, & if \quad Y(p_2) = Y(p_1) + 1 \quad and \quad X(p_1) = X(p_2).
\end{cases}
$$

The corresponding operator on $\mathcal{V}_0$ is given by the following pentadiagonal matrix A_0:

$$
a^0_{p_1,p_2} = \begin{cases}
4\,(k+1)^2, & if \quad X(p_1) = X(p_2) \quad and \quad Y(p_1) = Y(p_2), \\
-\,(k+1)^2, & if \quad X(p_2) = X(p_1) - 1 \quad and \quad Y(p_1) = Y(p_2), \\
-\,(k+1)^2, & if \quad X(p_2) = X(p_1) + 1 \quad and \quad Y(p_1) = Y(p_2), \\
-\,(k+1)^2, & if \quad Y(p_2) = Y(p_1) - 1 \quad and \quad X(p_1) = X(p_2), \\
-\,(k+1)^2, & if \quad Y(p_2) = Y(p_1) + 1 \quad and \quad X(p_1) = X(p_2).
\end{cases}
$$

We use linear interpolation to map between the two domains. In this case the value of the function f depends on k. This in turn implies that the condition number of the preconditioned matrix is not constant. The calculations of the constants is very similar to that of the one dimensional case, albeit more tedious.

An application of Gershgorin's Disk Theorem again shows that $\rho(A) = O(2^{2k})$. An estimate of the spectral radius of the Jacobi matrix provides an estimate of the values of α_0 and α_1:

$$
\begin{aligned}
\alpha_0 &= O\left(2^{2k}/k\right) \\
\alpha_1 &= O\left(2^{2k}\right).
\end{aligned}
$$

Similarly we can show that that

$$
\lambda_0 = \lambda_1 = O(1).
$$

An analysis of the two grids reveals that

$$\beta_0^{-1} = O(1)$$

and

$$\beta_1 = O\left(\frac{1}{k^2}\right).$$

Similarly

$$\gamma_0^{-1} = O(\rho_A) = O(2^{2k}).$$

The value of the function f again grows at least linearly with k.

By replacing the above *expressions* (which are not constants) in the estimate we can see that the estimate of the condition number of the resulting system is not really constant, but can grow as rapidly as

$$\begin{aligned} \kappa(BA) &\leq (2^{2k} + \frac{c_1}{k^2})((\frac{2^{2k}}{k})^{-1} 2^{2k} + c_2 k) \\ &= O(k\, 2^{2k}). \end{aligned}$$

In §3.3, we will show a similar lower bound (up to a polynomial factor) for the condition number.

3.3 Common discussion for first two examples

The examples of §3.1 and §3.2 show that it is trivial to map data onto a uniform grid where the assumptions of the Theorem 2.1 are not satisfied. This can result in the construction of an extremely badly conditioned linear system.

We now show that the bounds derived are tight to within a polynomial for the examples considered. To avoid excessive algebra we use the one dimensional example of §3.1 to illustrate the case.

We make an additional assumption concerning the form of B_0. In the following discussion we assume that $\lambda_{\max}(B_0)$ is a constant. This assumption is justified as B_0 is a preconditioner for A_0 and $\lambda_{\min}(A_0)$ is a constant. Specifically, we assume that

$$||B_0||_\infty \leq \kappa_1.$$

From (2.1), we have

$$BA = RA + \Pi B_0 \Pi^t A.$$

Let e_i denote the i^{th} canonical vector. So

$$(BAe_1, e_1) = (RAe_1, e_1) + (\Pi B_0 \Pi^t A e_1, e_1).$$

Using the form of A in (3.3) we see that

$$Ae_1 = 2^{2k+1} e_1 - \frac{1}{3} 2^{2k+3} e_2.$$

Since the smoother R is Jacobi's method,

$$Re_1 = \frac{2}{3} e_2$$

and

$$Re_2 = \frac{2}{3}e_1 + \frac{2}{3}e_3.$$

This implies that

$$|(RAe_1, e_1)| = \frac{2}{9}2^{2k+3}.$$

It is easy to show that for the grids in question that

$$(\Pi B_0 \Pi^t A e_1, e_1) = 0, \qquad \forall k > 4.$$

Since

$$|(BAe_1, e_1)| = \frac{2}{9}2^{2k+3},$$

we have that

$$\lambda_{\max}(BA) \geq \frac{2}{9}2^{2k+3}. \tag{3.4}$$

We can easily show that

$$|(RAe_k, e_k)| = \frac{128}{9}.$$

From the projection scheme we get that

$$(\Pi v^*, e_k) = v^*_{\frac{k+1}{2}} \qquad v^* \in \mathcal{V}_0$$

Therefore

$$\begin{aligned}
\delta &= (\Pi B_0 \Pi^t A e_k, e_k) \\
&= (B_0 \Pi^t A e_k)_{\frac{k+1}{2}} \\
&= \left[B_0 \Pi^t (16 e_k - \frac{64}{3} e_{k-1}) \right]_{\frac{k+1}{2}} \\
&\leq \|B_0\|_\infty (16 \|\Pi^t e_k\|_\infty + \frac{64}{3} \|\Pi^t e_{k-1}\|_\infty) \\
&= \kappa_1 (16 + \frac{64}{3}) \\
&= O(1).
\end{aligned}$$

As a result,

$$\|(BAe_k, e_k)\| \leq \frac{128}{9} + O(1) = O(1).$$

This implies that

$$\lambda_{\min}(BA) \leq O(1). \tag{3.5}$$

From (3.5) and (3.4),

$$\kappa(BA) \geq \frac{2}{9c^*}2^{2k+3}, \qquad c^* = O(1).$$

The bound derived in this case is tighter than the bound derived in Theorem 2.1. The lower bound in the general case is given by

$$\kappa(BA) \geq (\frac{2^{2k}}{k} - \frac{1}{k^2})(c^* + \frac{1}{k^2})^{-1} = O(2^{2k}/k).$$

The upper bound derived in the previous section is tight up to a polynomial factor for this pair of grids. We present numerical evidence in §4 to validate our claims.

3.4 1-D example with nearly matched grids

We now investigate the case where the uniform and nonuniform grids are not as mismatched as in the previous examples. We restrict out analysis to the one dimensional case (3.1) for clarity of presentation. As in §3.1, a two dimensional example on a tensor product set of grids produces quite similar results.

The nonuniform grid in the following example is given by a set of $2n$ points. The grid is defined such that the spacing between the last n points is exactly half of the inter-point spacing between the first n points. As a result,

$$\mathcal{V} = \left(\frac{2}{3n+1}, \frac{4}{3n+1}, \ldots, \frac{2n}{3n+1}, \frac{2n+1}{3n+1}, \ldots, \frac{3n}{3n+1}\right) \tag{3.6}$$

The uniform grid is given by a grid having $2n$ uniformly spaced points on the interval $[0, 1]$. The uniformly spaced grid is given by

$$\mathcal{V}_0 = \left(\frac{1}{2n+1}, \frac{2}{2n+1}, \ldots, \frac{2n}{2n+1}\right).$$

A set of C^0-piecewise linear functions is used as the finite element basis on the two grids (see Figure 3.3). As before, we use Jacobi as a smoother in this case. Linear interpolation is used to map between the two domains. We assume that all functions are sufficiently smooth. An estimate of the spectrum of the Jacobi matrix on the non-uniform grid provides us with the following estimates of the values of α_0 and α_1:

$$\begin{aligned} \alpha_0 &= O(1) \\ \alpha_1 &= O(n). \end{aligned}$$

Since we assume that the equation is solved exactly on the uniform grid, we have

$$\lambda_0 = \lambda_1 = 1.$$

From the smoothness of the functions and the two grids it follows that

$$\beta_0^{-1} = O(1 + \frac{1}{n}) = O(1)$$

and

$$\beta_1 = O(1 + \frac{1}{n}) = O(1).$$

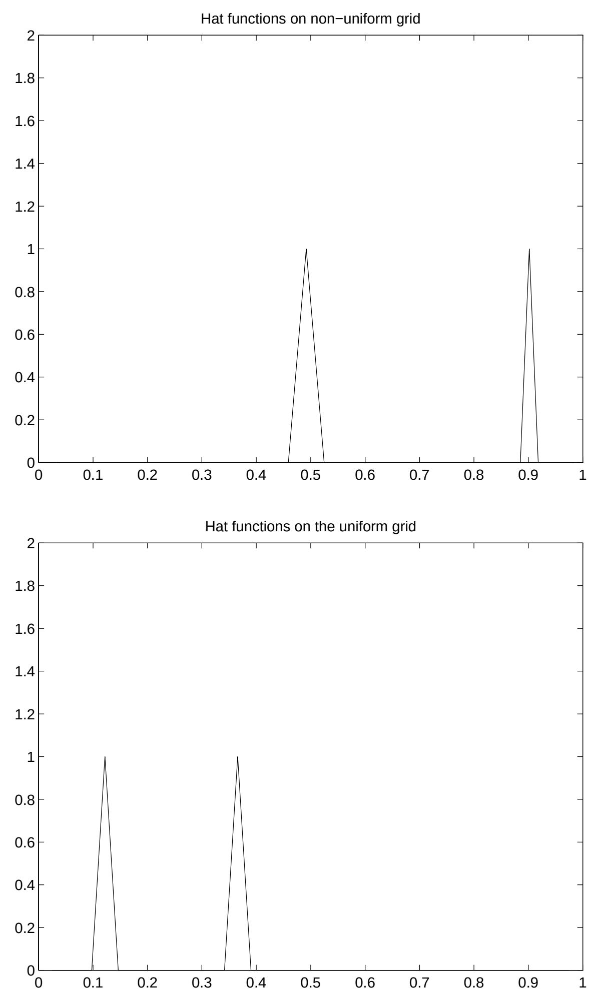


Figure 3.3: Linear hat functions on $\mathcal{V}$ and $\mathcal{V}_0$

Similarly

$$\gamma_0^{-1} = O(1).$$

By replacing the constants in the expression for the bounds on the condition number we see that

$$O(1) \leq \kappa(BA) \leq O(n). \tag{3.7}$$

Note that the upper bound in (3.7) would be $O(1)$ if $\mathcal{V}$ and $\mathcal{V}_0$ were nested. At first glance, (3.7) appears not to be sharp. If $\mathcal{V}$ and $\mathcal{V}_0$ in this case were nested, the upper bound would be $O(1)$. Table 4.1 indicates that the upper bound is actually $O(n)$.

Table 4.1: $\kappa(BA)$ for examples in §3.1 and §3.4

k	$\kappa(BA)$	
	§3.1	§3.4
5	1.8×10^4	8.5
10	1.0×10^7	7.6
15	6.9×10^{11}	10.1
20	3.6×10^{14}	7.9
25	2.4×10^{19}	14.3
30	1.2×10^{22}	20.1
40	5.5×10^{28}	21.1
50	6.5×10^{31}	23.5
75	2.4×10^{44}	38.8
100	2.7×10^{55}	67.5

4 Numerical experiments.

We performed experiments involving the grids $\mathcal{V}$ and $\mathcal{V}_0$ as outlined in §3. The matrices Π and Π^t correspond to the use of linear interpolation between the two domains. For ease of computation B_0 was chosen to be A_0^{-1}, i.e., the multigrid solution on the uniform grid corresponded to an exact solution of the transferred problem. We used Jacobi as the smoother. The condition number of BA was estimated using Hager's method (see [5]).

The first column of Table 4.1 corresponds to the case of $\mathcal{V}$ in (3.2) and the second column of Table 4.1 corresponds to $\mathcal{V}$ in (3.6).

As can be seen from Table 4.1, the condition number of the matrix BA grows exponentially for the case where the grids are very poorly matched. However, for the case where the grids are somewhat similar we find that the condition number grows quite slowly as a function of k. In fact, for values of k that might be used in practice, the condition number remains relatively constant.

5 Conclusions.

Although the method suggested in [10] can be applied in a variety of situations, great care must be taken in the application of the method. A naive application of the method can lead to a situation where the condition number of the system is not bounded by a constant as more mesh points are added leading to the loss of optimality inherent in the multigrid method.

In the worst case scenario, the auxiliary grid $\mathcal{V}_0$ must have a uniform mesh spacing equivalent to the smallest one in $\mathcal{V}$. This is precisely the computational grid that unstructured grid methods are designed to avoid. While an optimal preconditioner can be constructed from a condition number viewpoint, it is not optimal from a computational complexity viewpoint.

The inequalities of Theorem 2.1 are bounded in a large number of cases, especially when the two grids involved have the same underlying structure. In this case the data can be mapped onto a lower dimensional auxiliary space which might provide a reduction in the computation requirements. At this stage we point out that, in practice, the method will probably work on structural problems for stress. The unstructured grids are nearly structured and high order basis elements are used. In this case the problem is normally mapped to an auxiliary space which uses lower dimensional basis elements. As a result the explosive growth in the condition number, arising from the mapping process, is not seen.

In [4], $\mathcal{P}_1$ conforming finite elements are used to construct optimal preconditioners for higher order finite elements. The conforming finite elements are shown to provide optimal preconditioners in the case where the underlying grids are the same. In [2], the authors discuss a method for using a coarse grid for preconditioning a finer grid, assuming that the auxiliary problem is conforming.

As has been shown in [10], the method can be applied to obtain an optimal preconditioner in a large number of cases. In particular, when the grids involved are nearly structured, as are all of the examples presented in [10], the process of mapping one grid to the other is a fairly well conditioned process. In cases such as this, the method offers the advantage of working with uniform grids in the auxiliary space instead of the nonuniform grids generated during the discretization of the problem. As long as the spectral properties of the discretized operators in both spaces reflect the original spectrum of the differential operator the assumptions of the theorem are satisfied.

In the case where the auxiliary grid has the same underlying discretization as the original grid but uses lower dimensional basis elements, the projected vectors on the new grid are close approximations to the original vectors. As a result the mapping from one grid to the other does not incur a substantial loss of information.

However in the case where the grids involved are physically different, the error transferred to the auxiliary grid may be much greater than the error in the original grid. In this case the spectral properties of the operator on the nonuniform grid are as much an artifact of the discretization scheme used as of the original differential operator. The error involved in projecting data from one grid to the other is dependent on the ratio of the grid spacing in both grids. Provided this ratio is bounded (as in ordinary multigrid), the transformation can be made without a dramatic increase in the condition number. However when the ratios vary, from $2^{-k}/k$ to $2^{-1}/k$ in the example presented above, the projection itself is an extremely ill-conditioned operation. As a result the error on the auxiliary grid bears little or no resemblance to the error on the primary grid. This causes the entire method to be very badly conditioned leading to exponential condition numbers that we see in the above example.

REFERENCES

1. D. Braess. *Algebraic multigrid for elliptic problems of second order*, Computing, 55 (1995), pp. 379–393.

2. J. H. Bramble, J. E. Pasciak, and X. Zhang. *Two level preconditioners for 2m'th order elliptic finite element problems*, East-West J. Numer. Math., 4 (1996), pp. 99–120.
3. A. Brandt. *Lecture notes*, ICASE Workshop on Multigrid Methods, Hampton, VA, 1978.
4. S. C. Brenner. *Preconditioning complicated finite elements by simple finite elements*, SIAM J. Sci. Comput., 17 (1997), pp 1269–1274.
5. N. J. Higham. *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (algorithm 674)*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.
6. P. Leinen. *Data structures and concepts for adaptive finite element methods*, Computing, 55 (1995), pp. 325–354.
7. D. J. Mavriplis. *Turbulent flow calculations using unstructured and adaptive meshes*, Int. J. Numer. Meth. Fluids, 13 (1991), pp. 1131–1152.
8. N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas. *Seventh Copper Mountain Conference on Multigrid Methods*, Vol. CP 3339, NASA, Hampton, VA, 1996.
9. U. Rüde. *Mathematical and Computational Techniques for Multilevel Adaptive Methods*, Frontiers in Applied Mathematics **13**, SIAM, 1993.
10. J. Xu. *The auxiliary space method and optimal multigrid preconditioning techniques for unstructured meshes*, Computing, 56 (1996), pp. 215–235.