Preconditioning Methods for Very Ill-conditioned Three-dimensional Linear Elasticity Problems

E. GRAHAM AND P. A. FORSYTH
Department of Computer Science
University of Waterloo

Contact address :
P. A. Forsyth
Department of Computer Science
University of Waterloo
Waterloo, ONT.
Canada N2L 3G1
Tel: (001 519) 888-4567 ext. 4415
Fax: (001 519) 885-1208
Email: paforsyt@yoho.uwaterloo.ca

Abstract

Finite element models of linear elasticity arise in many application areas of structural analysis. Solving the resulting system of equations accounts for a large portion of the total cost for large, three-dimensional models, for which direct methods can be prohibitively expensive. Preconditioned conjugate gradient (PCG) methods are used to solve difficult problems with small ($\ll 1$) average element aspect ratios. Incomplete LU (ILU) factorisations based on a drop tolerance parameter are used to form the preconditioning matrices. Various new techniques known as reduction techniques are examined. Combinations of these reduction techniques result in highly effective preconditioners for problems with very poor aspect ratios. Standard and hierarchical triquadratic basis functions are used on hexahedral elements, and test problems comprising a variety of geometries with up to 50,000 degrees of freedom are considered. Manteuffel's method of perturbing the stiffness matrix to ensure positive pivots occur during factorisation is used, and its effects on the convergence of the preconditioned system are discussed.

Keywords : 3D elasticity, PCG, preconditioning, reduction

Running title : Ill-conditioned PCG methods

Last Revision : April 9, 1998

1 INTRODUCTION

The most CPU-intensive part of any three dimensional elasticity computation involves the solution of the large sparse stiffness matrix. It is generally conceded that iterative methods will eventually displace the traditional direct methods for solution of these systems. Some recent articles on this topic include [1, 2, 3, 4, 5, 6, 7]. An extensive list of previous work is also given in [7].

Preconditioned conjugate gradient (PCG) methods are generally the methods of choice for the iterative solution of symmetric positive definite (SPD) stiffness matrices. However, the success of PCG methods is crucially dependent on the preconditioning technique used. For well-conditioned problems, a standard incomplete factorisation

based on either a level of fill [8] or a drop tolerance [9] works well. However, practical problems are often very poorly conditioned. The most common source of poor conditioning is when the elements have aspect ratios far from unity. Although it is the goal of most modern mesh generators to produce meshes where all the elements have good aspect ratios, this does not appear to the case in practice. Also, there are many problems where one of the physical dimensions of the object being modelled is much smaller than the other dimensions. In this case, a natural mesh for the problem typically has elements with extreme aspect ratios.

The objective of this article is to investigate some new approaches for preconditioning based on reduction techniques. These methods entail the generation of an intermediate matrix (based on the original stiffness matrix), and an incomplete (or complete) factorisation of this intermediate matrix is then used as a preconditioner. We give a more detailed explanation in Section 3, and the performance of these preconditioners will be tested on a variety of badly conditioned problems that challenge the robustness of iterative solvers. Performances will also be compared to that of a direct solver.

2 FORMULATION

The three dimensional linear elasticity problem for a homogeneous, isotropic material is given in terms of displacement vector $\boldsymbol{x}$ (with three degrees of freedom at each node) by the following system of equations :

$$\mu \nabla^2 \boldsymbol{x} + (\lambda + \mu) \nabla \nabla \cdot \boldsymbol{x} = 0 \,, \tag{1}$$

where the material properties are given by

$$\begin{aligned} \mu &= \frac{E}{2(1+\nu)} \,, \quad \text{and} \\ \lambda &= \frac{E\nu}{(1+\nu)(1-2\nu)} \end{aligned}$$

for Young's modulus E and Poisson's ratio ν.

In the following, we will be using the popular triquadratic basis functions defined on subparametic 20-node hexahedral elements [10]. Following the usual assembly procedure, we obtain the linear system

$$A\boldsymbol{x} = \boldsymbol{b} \tag{2}$$

where A is an SPD stiffness matrix. The test problems will be characterised by the average aspect ratio of the elements δ, which we define as the average (over the whole grid) of $\delta_e = d_s/d_l$, the ratio of the shortest to the longest side, for each element. Clearly, we expect that the problems will become more difficult as $\delta \to 0$.

3 PRECONDITIONING

Incomplete LU (ILU) factorisation is a robust, well-understood preconditioning technique, and has been extensively documented [2, 8, 7]. Criteria for discarding the fill-in entries that occur during the factorisation are generally based on the graph of the matrix (levels of fill) [11, 8], the magnitudes of the entries (drop tolerance) [9, 12], or both. There is a standard definition of a level-based ILU [7], whereas a drop tolerance ILU can vary according to the form of the dropping criterion. Dropping a fill-in entry according to its value is an intuitively clearer concept than that of a dropping criterion based on the nodal elimination graph, and in [7] it was concluded that for linear elasticity applications, a drop tolerance ILU generally outperforms a level-based ILU having the same number of nonzero entries.

We denote by $A^{(k)}$ the matrix obtained after $(k-1)$ stages of elimination. In this work, a fill-in entry $a_{ij}^{(k+1)}$ during the k^{th} stage will be discarded if

$$|a_{ij}^{(k+1)}| < \epsilon a_{ii}^{(k+1)} \tag{3}$$

where $\epsilon \in [0,1]$ is the drop tolerance parameter, otherwise it is kept. Note that using a drop tolerance of $\epsilon = 1$ permits no fill-in entries to remain in the preconditioner, whilst using a value of $\epsilon = 0$ gives a complete factorisation. Using ILU preconditioning, we thus obtain a preconditioner of the form

$$C = \bar{L}\bar{U} = A + R \tag{4}$$

where the magnitude of the *defect* matrix R (representing the accuracy of the approximation to A) is controlled by the value of ϵ. If A is an M-matrix, it is known that the condition number

$$\kappa_F = \kappa(C^{-1}A) = \|C^{-1}A\|_2 \|(C^{-1}A)^{-1}\|_2 \tag{5}$$

of the preconditioned system decreases with $\|R\|$, and our experience indicates that this is true for other stiffness matrices also. Recall that a bound for the number of iterations required for a given solution tolerance for a PCG method is given in terms of the condition number of the preconditioned system κ_F [13]. Hence one can prescribe an appropriate value for ϵ to suit the needs of the problem; for example, solving ill-conditioned problems requires a finer drop tolerance (to ensure good convergence) than solving well-conditioned problems [7].

However, there are some serious drawbacks to this method of preconditioning. As the number of unknowns (N) in the problem becomes larger, the number of nonzero entries in the preconditioner C grows considerably; for three-dimensional analyses, a single mesh refinement in each coordinate direction increases N eightfold, and the time and storage requirements for computing C rapidly exceed acceptable levels. Secondly, it is known that for a standard ILU factorisation, κ_F is $\mathcal{O}(h^{-2})$ (or $\mathcal{O}(h^{-1})$ for modified ILU factorisations, where applicable [14, 15, 3]); thus for a given convergence tolerance,

more PCG iterations are needed as the mesh is refined. Also, the cost of each PCG iteration is related to the number of nonzero entries in C, making the iterative solution phase expensive for large problems. In addition, when A is not an M-matrix, the occurrence of negative pivots [9, 16] in the preconditioner could cause the factorisation to be restarted, which for large problems would be prohibitively time-consuming.

3.1 Reduction techniques

Thus there is a need to improve the quality of drop tolerance ILU preconditioners, so that the number of nonzero entries is reduced, yet enough information is retained to yield a low condition number κ_F, and that the appearance of negative pivots is less common. To this end, we seek to alter the stiffness matrix A before computing C, and discard the less important entries (in terms of minimising κ_F), or perturb the diagonal entries to ensure the appearance of positive pivots.

Preconditioning techniques that entail generating an intermediate matrix before the incomplete factorisation is performed are referred to in this work as **reduction techniques**. A new matrix B is derived from the original stiffness matrix A, and an incomplete factorisation C of B is used to precondition A. The process of obtaining B from A is called a **reduction**, since B usually has fewer nonzero entries than A. Given that C is now *indirectly* approximating A, one would expect some penalty to be incurred in terms of minimising $\kappa_F = \kappa(C^{-1}A)$. We can, however, show that this penalty is bounded above [3]: since

$$X = YZ \quad \Rightarrow \quad \|X\|_p \leq \|Y\|_p \|Z\|_p$$

for any matrices X, Y, Z and any p-norm, by writing $C^{-1}A = C^{-1}BB^{-1}A$ and letting

$$\kappa_1 = \kappa(B^{-1}A), \quad \kappa_2 = \kappa(C^{-1}B)$$

we have that

$$\kappa_F \leq \kappa_1 \kappa_2. \tag{6}$$

Hence the difference between κ_2 and κ_F is bounded by a factor of κ_1. Note that κ_1 is fixed by the choice of reduction, whereas κ_2 can be made arbitrarily small by choosing $\epsilon \approx 0$.

Reduction techniques often yield intermediate matrices B that are *spectrally equivalent* to A with respect to certain problem parameters. Families of matrices $A(\rho)$ and $B(\rho)$ that vary with parameter ρ are said to be spectrally equivalent if there exist constants α, β ($\alpha \leq \beta$) such that

$$\alpha \leq \frac{\boldsymbol{x}^T A(\rho)\boldsymbol{x}}{\boldsymbol{x}^T B(\rho)\boldsymbol{x}} \leq \beta \quad \forall \quad \rho, \boldsymbol{x} \neq \boldsymbol{0} \tag{7}$$

where α and β are *independent* of ρ. It can be shown that if B is symmetric positive definite (SPD) then the eigenvalues of $B^{-1}A$ all lie in the interval $[\alpha, \beta]$ and thus $\kappa_1 \leq \frac{\beta}{\alpha}$.

We now describe three different reduction techniques, and detail the particular merits of each.

3.1.1 *H Reduction*

This technique applies to stiffness matrices arising through the use of hierarchical basis functions [17] in the finite element discretisation, and is also known as **two-level preconditioning**. It was first proposed by Axelsson and Gustafsson [18] and was extensively analysed by Jung, Langer and Semmler in [19]. For quadratic basis functions, the original matrix A can be reordered to take the block form

$$A = \begin{pmatrix} A_{vv} & A_{vm} \\ A_{mv} & A_{mm} \end{pmatrix} \tag{8}$$

where v and m denote the vertex and midside nodes respectively. Note that the matrix A_{vv} is the stiffness matrix that arises from a *linear* finite element discretisation, hence it is known to be SPD, and it can be shown that A_{mm} is also SPD [13]. Thus the decoupled matrix

$$B_H = \begin{pmatrix} A_{vv} & 0 \\ 0 & A_{mm} \end{pmatrix} \tag{9}$$

is itself SPD and could therefore be used as a starting point for an ILU factorisation preconditioner C.

It can be shown that A and B_H are spectrally equivalent with respect to the grid discretisation parameter h [18]. So, given a particular problem and an initial grid, one can conclude that $\kappa_1 = \kappa(B_H^{-1}A)$ will not grow as the grid is successively refined. It has also been observed that A and B_H are spectrally equivalent with respect to NEL (the number of elements in the grid), i.e. the grid can be enlarged by adding similar elements without affecting κ_1. These two properties together imply that κ_1 does not grow with N, the total number of degrees of freedom. It has been demonstrated, however, that constants α, β *are* dependent upon the type of element used in the discretisation, the degree of element deformation and on the value of Poisson's ratio ν, but not on the shear modulus G [19]. It was observed that the effect of ν on α, β was less significant than that of the element deformation, for realistic problems; in particular, the average element aspect ratio δ, and the angles between the faces of the elements (degree of distortion) were more influential.

Given that A_{vv} corresponds to a linear FE stiffness matrix, its condition number $\kappa(A_{vv})$ is $\mathcal{O}(h^{-2})$, and it can be shown that $\kappa(A_{mm})$ is independent of h [18], and also of N. The preconditioner C takes the form

$$C = \begin{pmatrix} C_{vv} & 0 \\ 0 & C_{mm} \end{pmatrix}$$

so if we choose $C_{vv} = A_{vv}$ (a *complete* factorisation of A_{vv}) and C_{mm} to be any incomplete factorisation of A_{mm}, we have

$$C^{-1}B_H = \begin{pmatrix} I_{vv} & 0 \\ 0 & C_{mm}^{-1}A_{mm} \end{pmatrix} \tag{10}$$

where I_{vv} represents the identity submatrix, and hence $\kappa_2 = \kappa(C^{-1}B_H)$ is also independent of h. So when using preconditioners of this form, the number of iterations needed for convergence will not grow as the grid is refined or enlarged. It has been observed from results in [19] that $\kappa(A_{mm})$ is *highly* dependent upon both ν and δ. Dickinson and Forsyth [7] have demonstrated the superiority of this preconditioning technique over standard ILU factorisation with no reduction, particularly for ill-conditioned problems.

3.1.2 *C Reduction*

This technique applies to matrices that are not Stieltjes matrices, as is usually the case when solving stress analysis problems, and is also known as **diagonal compensation**. It is a method of deriving a Stieltjes matrix B_C from the original matrix A via the removal of positive off-diagonal entries. A matrix is *Stieltjes* if it is symmetric, positive definite and has non-positive off-diagonal entries. It is readily observed that all Stieltjes are also M-Matrices, and thus standard ILU factorisations will automatically yield positive definite preconditioners. The idea was first proposed by Beauwens and Wilmet [14] and was more recently used by St. Georges and Warzee [3] during their investigation of various modified ILU factorisation schemes for Stieltjes matrices. The process of obtaining B_C from A entails dropping the positive off-diagonal entries and adding them to the diagonal, thereby preserving the row sums of A, and is fully described in [3], where it is shown that the resultant matrix B_C is SPD (and therefore a Stieltjes matrix).

Spectral equivalence properties for A and B_C have not been derived, thus one can assume that $\kappa_1 = \kappa(B_C^{-1}A)$ will vary as the mesh is refined or enlarged, and also according to other problem parameters. It can be shown that $\lambda_i(B_C^{-1}A) \in (0, 1]$ for each eigenvalue λ_i [13], and hence

$$\kappa_1 \leq \frac{1}{\lambda_{min}(B_C^{-1}A)}.$$

In [15] it is shown that with an appropriate modified ILU factorisation strategy, and some additional conditions imposed upon B_C such as diagonal dominance, it is possible to obtain a bound for $\kappa_2 = \kappa(C^{-1}B_C)$ of the form

$$\kappa_2 \leq m + l + 2$$

with constants $m, l \geq 0$. The value of m is determined by the factorisation strategy employed, and l is dependent on the ordering of the unknowns. A precise definition for l and some suggested ordering methods can be found in [15]. This bound has been considered as an algebraic generalisation of the $\mathcal{O}(h^{-1})$ bound developed by Axelsson using a geometric approach, and a variety of factorisation strategies that refine it were analysed and tested in [3].

3.1.3 *D Reduction*

This process can be applied to any stiffness matrix arising from problems with more than one degree of freedom per node. It was first applied to 3D structural analysis problems by Shlafman and Efrat [20] and proceeds in a similar vein to H reduction. The original matrix A is ordered by grouping together unknowns pertaining to the same nodal degree of freedom; thus for three-dimensional linear elasticity, each node has three associated degrees of freedom, corresponding to displacements in the x, y and z directions. The matrix A can then take the block form

$$A = \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{yx} & A_{yy} & A_{yz} \\ A_{zx} & A_{zy} & A_{zz} \end{pmatrix} \tag{11}$$

It should be noted that each A_{ii} block is the stiffness matrix for the corresponding one-dimensional problem, and is thus symmetric positive definite. Therefore the decoupled matrix

$$B_D = \begin{pmatrix} A_{xx} & 0 & 0 \\ 0 & A_{yy} & 0 \\ 0 & 0 & A_{zz} \end{pmatrix} \tag{12}$$

is also SPD, and hence a valid choice for ILU factorisation.

It can be shown that when A is a stiffness matrix for problems involving three-dimensional solid objects, A and B_D are spectrally equivalent with respect to NEL and h [20], and thus $\kappa_1 = \kappa(B_D^{-1}A)$ will not grow as the grid is refined or enlarged. Spectral equivalence with respect to other problem parameters has not been established.

Since each A_{ii} is itself a stiffness matrix, the corresponding $\kappa(A_{ii})$ will be $\mathcal{O}(h^{-2})$ and thus an ILU factorisation of B_D will yield $\kappa_2 = \kappa(C^{-1}B_D)$ as $\mathcal{O}(h^{-2})$. Thus for the number of iterations to remain constant with grid refinement, one must employ a complete factorisation of B_D, which is considerably more expensive than the complete factorisation of A_{vv} in equation (10). However, it was noted in [20] that the appearance of negative pivots during the factorisation was much less common than for standard ILU techniques, and hence modified ILU factorisations could conceivably apply.

3.2 Combining reduction techniques

It is clear that each of these techniques offers some advantage over conventional ILU preconditioning in terms of storing and computing the preconditioner C. For larger problems, however, the cost can still dominate the entire solution process. To remedy this, it is possible to combine two or more reduction schemes in a single preconditioning technique, in an attempt to attain the benefits of each, whilst further reducing the number of nonzeros in the matrix to be factored.

For C reduction in particular, the time and storage saved depends on the values of the matrix entries, and may not always be appreciable. St. Georges and Warzee [3]

suggested using D reduction followed by C reduction (referred to as **DC reduction**), to give the following Stieltjes matrix

$$B_{DC} = (B_D)_C = \begin{pmatrix} (A_{xx})_C & 0 & 0 \\ 0 & (A_{yy})_C & 0 \\ 0 & 0 & (A_{zz})_C \end{pmatrix}. \tag{13}$$

Factorisations of B_{DC} are much cheaper than those of B_C, and in addition the spectral equivalence of A and B_{DC} with respect to h and NEL has been established [3]. Thus one can achieve a bound for κ_F that is independent of h and NEL by completely factoring B_{DC}, or attain a bound of $\mathcal{O}(h^{-1})$ given an appropriate modified ILU factoring strategy.

For H reduction, the complete factorisation of A_{vv} (necessary for κ_F to be independent of h) becomes relatively expensive for larger problems ($N \geq 40,000$). In addition, when solving problems with a poor element aspect ratio, it has been recommended that a very low drop tolerance ILU, or even a complete factorisation, of A_{mm} be used [21, 7]. This will dominate the cost of solution for fairly low values of N, and make each iteration very expensive. To redress this, we can combine H and D reduction, giving

$$B_{HD} = \begin{pmatrix} (A_{vv})_D & 0 \\ 0 & (A_{mm})_D \end{pmatrix} \tag{14}$$

which we shall call **HD$_A$ reduction**. Given that A and B_D are spectrally equivalent for stiffness matrices A arising from three-dimensional solid problems, and that A_{vv} is itself the stiffness matrix arising from a linear FE discretisation, we deduce the spectral equivalence of $(A_{vv})_D$ and A_{vv}. Since we also know that $\kappa(A_{mm})$ is independent of h and NEL, we can show further that B_{HD} and B_H are spectrally equivalent with respect to these two parameters. Then since $\kappa(B_{HD}^{-1}A) \leq \kappa(B_{HD}^{-1}B_H) \cdot \kappa(B_H^{-1}A)$, which can be bounded independently of h and NEL, we must have spectral equivalence of A and B_{HD}. Hence using this technique, κ_F can be independent of h and NEL for the relatively small price of completely factoring $(A_{vv})_D$, and using any incomplete factorisation for $(A_{mm})_D$.

Although this approach is undoubtedly advantageous for large problems, there are still the effects of δ and ν to consider. Since both $\kappa(B_D^{-1}A)$ and $\kappa(B_H^{-1}A)$ are dependent on each of these parameters, it is likely that losing more information from A will have an adverse effect on κ_F. Therefore we also propose two further combinations, namely **HD$_v$ reduction** and **HD$_m$ reduction**, where only one of the matrix domains is decoupled, giving matrices

$$B_{HD_v} = \begin{pmatrix} (A_{vv})_D & 0 \\ 0 & A_{mm} \end{pmatrix}, \quad B_{HD_m} = \begin{pmatrix} A_{vv} & 0 \\ 0 & (A_{mm})_D \end{pmatrix}. \tag{15}$$

Clearly, greater savings in time and storage needed to compute C will accrue with the use of HD$_m$ reduction, since A_{mm} is much larger than A_{vv}, although the relative

sensitivity of each domain to changes in ν and δ has not yet been established; we will attempt to ascertain this from our numerical results.

3.3 Comparing reduction techniques

The question of which reduction technique to use is not one that can be easily answered in general, since for each there are a wealth of ILU factorisations to choose from, some of which are more suited to particular techniques (e.g. see [3] for examples of modified ILU factorisations suited for Stieltjes matrices). However, given that the factor $\kappa_1 = \kappa(B^{-1}A)$ in the bound on κ_F given in equation (6) is fixed once the reduction has taken place, some techniques may be inherently better than others for a given type of problem, e.g. one with poor aspect ratio elements. We postulate that the quantity κ_F can be bounded below by κ_1, giving

$$\kappa_1 \leq \kappa_F \leq \kappa_1 \kappa_2, \tag{16}$$

in other words, preconditioning by using a complete factorisation of the reduced matrix yields the lowest overall condition number, and hence the fewest iterations (but not necessarily the fastest solution time). We have observed this to be true during our experiments, and thus we can obtain some measure of the effectiveness of a given reduction technique by setting $C = B$, thereby giving $\kappa_2 = \kappa(C^{-1}B) = 1$ and $\kappa_F = \kappa_1$, and then using the actual number of PCG iterations taken for a given problem to gauge the relative size of κ_1. If the reduction is intrinsically *bad* (κ_1 is large) for a particular class of problem, then no preconditioning technique is likely to yield good results. Thus there is reason to measure the innate quality of each reduction in terms of the size of κ_1, with respect to various problem parameters such as Poisson's ratio ν and the average element aspect ratio δ. We present some results from such measurements in Section 5.1.

3.4 Avoidance of negative pivots

It is well known that convergence of the PCG method is only guaranteed if the preconditioning matrix C is symmetric positive definite. A necessary and sufficient condition for any real $n \times n$ matrix A to be SPD is that the pivots u_{ii} in the LU factorisation $A = LU$ are all positive [13]. From equation (4) we see that C will be SPD if and only if the pivots $\bar{u}_{ii}$ in the ILU factorisation $A = \bar{L}\bar{U} + R$ are all positive.

In this work, we have used a diagonal scaling of the original system (2) to give $DADy = Db$ where D is a diagonal matrix such that

$$d_{ii} = \frac{1}{\sqrt{a_{ii}}}, \qquad d_{ij} = 0 \quad \text{for } i \neq j.$$

This scales each element of A, leaving unit diagonals in each row; note that the matrix DAD is symmetric. The solution of (2) is then $x = Dy$, and this scaling is known to sometimes improve the condition number $\kappa(A)$ of A whilst leaving $\kappa(C^{-1}A)$ unchanged. We recall that for a stiffness matrix A of a problem with discretisation parameter h, we

have that $\kappa(A)$ is $\mathcal{O}(h^{-2})$ where $\lambda_{max}(A) = \mathcal{O}(h)$, $\lambda_{min}(A) = \mathcal{O}(h^3)$. It can be shown that with diagonal scaling, we have $\lambda_{max}(A) = \mathcal{O}(1)$, $\lambda_{min}(A) = \mathcal{O}(h^2)$. Moreover, it is computationally more efficient to work with unit diagonals, as the square root of a diagonal entry is needed for various calculations.

Generally speaking, it is not the case for stress analysis problems that all pivots $\bar{u}_{ii}$ will be positive, and two distinct approaches for remedying this situation have been devised. Jennings and Malik [9] proposed a method which we call **J&M Add**, whereby multiples of the dropped terms are dynamically added to the diagonal of the row being processed. This method is known to give an SPD preconditioner, but often overestimates the amount necessary to add to the diagonals, and in [7] it was shown that for linear elasticity analyses, an alternative algorithm devised by Manteuffel [16] is more suitable. This algorithm attempts to make the system more diagonally dominant, so that an ILU factorisation is more likely to succeed. If, during the course of the ILU, a negative pivot is encountered, the factorisation is aborted and restarted on the perturbed system

$$\hat{A} = A + \eta \cdot \mathrm{diag}(A) \quad \Rightarrow \quad \hat{A} = A + \eta I \quad \text{for scaled systems} \tag{17}$$

where I is the $n \times n$ identity matrix and $0 < \eta \ll 1$. If a negative pivot occurs a second time, a new attempt is made with η replaced by 2η, and so on. We will refer to this method as the **Restart** method, and it has been shown that for any SPD matrix A there exists some $\eta > 0$ such that an ILU factorisation of (17) will be successful [16].

It is believed that adding positive values to the diagonal, whether before or during the factorisation, can damage the convergence rate of the PCG method. Thus additions made to ensure positive pivots should be kept low in order to keep the number of iterations to a minimum. For the Restart method, we can give theoretical bounds on the condition number of the resulting preconditioned system. For the perturbed matrix $\hat{A}$ given in equation (17), we have

$$\kappa(C^{-1}A) \leq \kappa(C^{-1}\hat{A}) \cdot \kappa(\hat{A}^{-1}A)$$

where $\kappa(\hat{A}^{-1}A)$, the penalty for the perturbation, can be written as (assuming that the system has been scaled with unit diagonal)

$$\begin{aligned}
\kappa(\hat{A}^{-1}A) &= \kappa(A^{-1}\hat{A}) \\
&= \kappa(I + \eta A^{-1}) \\
&= \frac{\lambda_{max}(I + \eta A^{-1})}{\lambda_{min}(I + \eta A^{-1})} \\
&= \frac{1 + \eta \cdot \lambda_{max}(A^{-1})}{1 + \eta \cdot \lambda_{min}(A^{-1})} \\
&= \kappa(A) \left(\frac{\lambda_{min}(A) + \eta}{\lambda_{max}(A) + \eta} \right)
\end{aligned} \tag{18}$$

where $\lambda_{max}(A), \lambda_{min}(A)$ denote the maximal and minimal eigenvalues of A. Thus the bounds on the perturbation step penalty $\kappa(\hat{A}^{-1}A)$ are

$$1 \leq \kappa(\hat{A}^{-1}A) < \kappa(A)$$

for $\eta \in [0, \infty)$. Since $\kappa(\hat{A}^{-1}A)$ is strictly increasing with η, one would suspect that, outside of a certain safety range [1], the number of iterations needed to solve the system also increases with η. Furthermore, if $\eta \ll \lambda_{max}(A)$ (which will be the case for scaled systems) then from (18) we have

$$\kappa(\hat{A}^{-1}A) \approx 1 + \frac{\eta}{\lambda_{min}(A)}. \tag{19}$$

Thus for ill-conditioned problems where the minimal eigenvalue $\lambda_{min}(A) \approx 0$, such as problems with a small mesh size h, or poor aspect ratio elements, the penalty for perturbing the matrix will be severe unless a sufficiently small value for η can be found that makes the factorisation successful.

In this work, we will use the Restart method as given for scaled systems in (17) to ensure the appearance of positive pivots when necessary. For a drop tolerance ILU, the symbolic and numeric factorisation algorithms are combined, and both phases need to be repeated when the factorisation is restarted, which can be costly (especially for standard ILU preconditioners with no reduction). Hence it is necessary to find a suitable starting value for η, large enough that the factorisation succeeds before too long, and yet small enough to give good convergence in the iterative phase once the preconditioner has been computed. To this end, we choose $\eta = 10^{-4}$ for the reduction techniques, and $\eta = 10^{-3}$ for standard ILU preconditioners (where the extra factorisations are much more expensive). In addition, we have modified the algorithm slightly so that if, after five successive perturbations, the factorisation still fails, we replace η with 10η and start again; thus for a reduction technique with seven failed attempts before succeeding, the sequence of values used would be $\eta = 10^{-4}$, $2 \cdot 10^{-4}, \ldots, 5 \cdot 10^{-4}$, 10^{-3} and $2 \cdot 10^{-3}$, whereupon the factorisation would finish successfully.

The issue of the poorer performance of the J&M Add method was recently addressed by Hladík et al [2] who modified the algorithm to add only the necessary amounts to the diagonals. They concluded that for standard ILU preconditioners, their modification made for a substantial improvement in performance over the original algorithm, which we can verify from observations during our own experiments. We have not pursued their idea further here, but refer the interested reader to [2] for details.

4 TEST PROBLEMS

All of the test problems in this work are model problems arising from regular, structured grids that are partitioned into subparametric 20-node quadratic hexahedral elements. Table 1 summarises the test problem data, and detailed descriptions of each problem are given below.

[1] In [16] it is shown that the optimal value for η is slightly larger than the smallest for which the factorisation is successful, to avoid having pivots $\bar{u}_{ii} \approx 0$.

Table 1: *Summary of data for the various test problems used.*

	Degrees of Freedom	Nonzeros in $U(A)$	Relative Dimensions of Elements	Material Properties ν, E
Beam	13,503	986,370	$1 : \frac{1}{\delta} : \frac{1}{\delta}$	various, 100
Pipe	42,624	3,333,456	$1 : 75 : 114$	(inner) 0.45, 50 (outer) 0.3, 300
I-Beam	56,559	4,460,352	(flange centre) $1 : 1 : 20$ (flange sides) $1 : 20 : 20$ (web) $1 : 15 : 20$	0.3, 200
Washer	26,793	1,997,235	$1 : 164 : 125$	0.25, 20

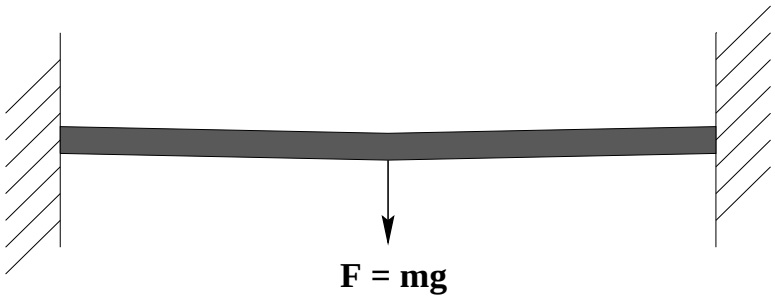

Figure 1: *Beam hanging under its own weight.*

4.1 Beam problem

This problem consists of a rectangular beam of variable thickness attached at each end to a wall, hanging under gravity as shown in Figure 1. The beam is held in tension, inducing a longitudinal strain of 1%. Its dimensions are 0.6 m $\times$ 0.15 m $\times$ 0.15δ m in the x, y and z directions respectively, and it is modelled on a $25 \times 7 \times 7$ grid, where the dimensions of each element are $1 \times \frac{1}{\delta} \times \frac{1}{\delta}$. Grids with $\nu = 0.4$ and $\delta = 1$, $\frac{1}{5}$, $\frac{1}{10}$, $\frac{1}{50}$, $\frac{1}{100}$ and $\frac{1}{500}$ are used to demonstrate the effects of poor element aspect ratio on the convergence of the iterative solution. Similarly, grids with $\delta = 1$ and $\nu = 0.4, 0.49, 0.499$ and 0.4999 show the effects of Poisson's ratio. A depiction of the surface mesh is shown in Figure 2; all such diagrams show modified versions of the actual grids used, for clarity.

The boundary conditions are designed to simulate the effects of the weight of the beam and the longitudinal strain. A downward force is prescribed at the centre of the underside of the beam (the centre of gravity) and the nodes adjacent to the left hand wall are fixed. Those adjacent to the right hand wall are displaced by 0.006 m in the x direction, to effect the 1% strain. The prescribed force was proportional to the cube of the beam thickness in each case, in order to adjust for the changing aspect ratio; in each case, the maximum vertical displacement was approximately 0.002 m.

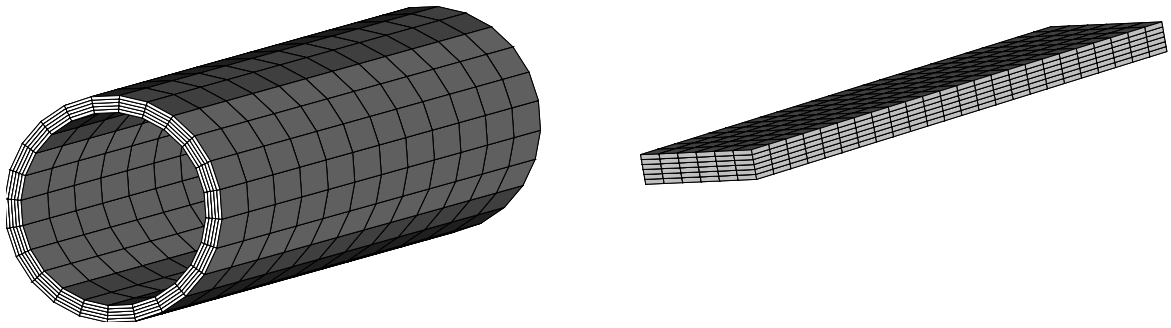

Figure 2: *Surface meshes of the pipe and beam.*

4.2 Pipe problem

This problem consists of a pipe made from two different materials, having inner radius 1.0 m, length 10.0 m and thickness 0.0175 m. The pipe comprises five concentric layers, of which the inner three layers are made from one material, and the outer two layers from another. A depiction of the surface mesh is shown in Figure 2.

Separate sets of boundary conditions simulate the pipe undergoing tension along its length, or torsion about its axis of symmetry. In each case, the nodes at one end of the pipe are fixed. For tension, an axial load is applied to the free end, to effect a longitudinal strain of approximately 3%. For torsion, a tangential planar force is applied to each surface element at the free end, resulting in an angular displacement of approximately 4°.

4.3 I-Beam problem

This problem consists of an I-beam of length 1.8 m and cross-sectional height 0.38 m, with a flange of width 0.24 m and thickness 0.04 m, and web thickness 0.04 m. The elements in the grid are of differing sizes and aspect ratios, as can be seen from the surface mesh in Figure 3.

Two sets of boundary conditions simulate the I-beam experiencing torsion and bending, respectively. In each case, the nodes at one end of the pipe are fixed. For torsion, a tangential planar force is applied to the free end, as for the equivalent pipe problem, resulting in a rotation of approximately 2°. For bending, a lateral force is applied at the free end, inducing a displacement of about 0.015 m.

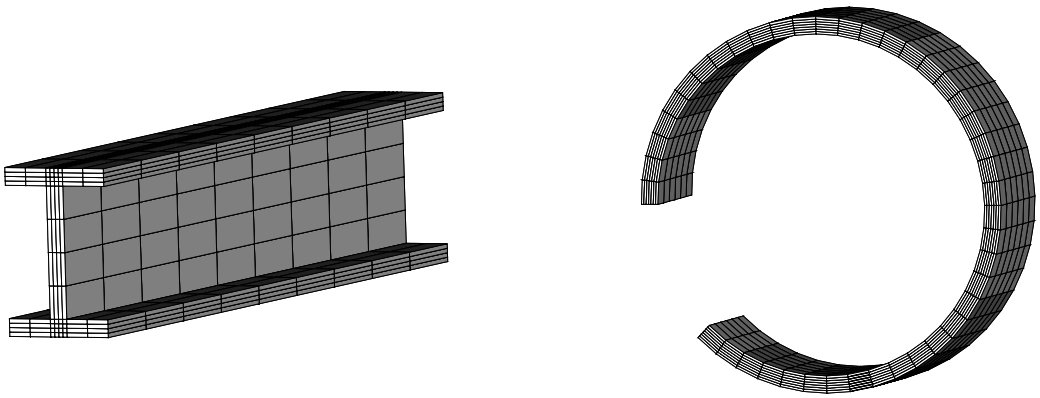

Figure 3: *Surface meshes of the I-beam and washer.*

4.4 Washer problem

This problem consists of a thin circular ring with an arc of sector angle 45° having been removed. The washer has inner radius 1.0 m, width 0.6 m and thickness 0.0056 m, and is shown in Figure 3.

Once again, two sets of boundary conditions simulate different loadings on the washer. A central line of nodes along the outside surface of the washer have been fixed, and a normal force has been applied to each of the ends, to create the effect of prising the washer open. The maximum displacement in any direction is approximately 0.017 m. Also, equal and opposite axial forces have been applied to the normal surface of the washer near each end, to produce lateral displacements of approximately 0.008 m.

5 RESULTS

Results are given for experiments that test the various reduction techniques (summarised in Table 2) with respect to the average element aspect ratio δ and Poisson's ratio ν. Many experiments were carried out; however, in the interests of brevity, details will be given for a representative sample of the tests which highlight typical results from all the experiments. Results for the beam problem are given first, and then successful techniques will be tested further on the larger, more geometrically complex problems.

Results are generally given in terms of CPU time and the number of iterations. CPU time includes computation of the preconditioner, including all repeated factorisation attempts, and iterative solution of the linear system, but not the assembly of the stiffness matrix. All tests were run on a Sun Sparc 10 workstation with a 32 bit word size, 256 Megabytes of RAM and 256 Megabytes of virtual memory. Standard basis functions were used in all experiments except those where H* reduction techniques were

Table 2: *Brief summary of the different reduction techniques.*

Reduction	Explanation
C	Remove positive off-diagonal entries and add to the diagonal
D	Decouple matrix with respect to degrees of freedom
H	Decouple matrix with respect to vertex and midside nodes
DC	Perform D-reduction and then C-reduction
HD$_A$	Perform H-reduction and then D-reduction
HD$_v$	Perform H-reduction, and then D-reduction on A_{vv} only
HD$_m$	Perform H-reduction, and then D-reduction on A_{mm} only

involved, in which case hierarchical basis functions were used instead, and a direct solve was performed on the vertex node domain. Whenever a direct solve was performed, the matrix (or submatrix) was ordered with the Minimum Degree ordering algorithm [22], otherwise the Reverse Cuthill McGee (RCM) algorithm [23] was used.

The tests were run using double-precision storage for both the stiffness and preconditioning matrices, except for the I-beam and pipe problems, where the preconditioner (only) was stored in single-precision to reduce memory requirements. The l_2 residual of any computed solution was calculated once the solution had been obtained, to validate the residual given by the PCG algorithm upon convergence. For the beam problems, the residual arising from a direct solve of a given problem was used as the convergence tolerance for the iterative solution. For the larger problems, the convergence criteria varied with each problem, and will be discussed later in this section. All iterative solutions used the zero vector as an initial guess.

Given that the matrices involved are symmetric, only the upper halves need be stored and processed, thereby reducing both storage costs and execution times.

5.1 Results for the beam problem

As mentioned in Section 3, we can give an estimate of the relative magnitude of $\kappa_1 = \kappa(B^{-1}A)$ for the different reduction techniques, by utilising a complete factorisation of the reduced matrix as a preconditioner, and thus gain an insight into the intrinsic quality of each reduction. Figures 4 and 5 show the solve time and number of iterations needed by each reduction as a function of the aspect ratio and Poisson's ratio, for the various incarnations of the beam problem, and also show the time taken for a direct solve of the original matrix. From them, we see that as the aspect ratio worsens, the hierarchical basis techniques H and HD$_m$ reduction are by far the best choices, and the only ones for which convergence was achieved in under 1000 iterations. The poorer performance of HD$_v$ and HD$_A$ reduction implies that $\kappa(A_{vv})$ is much more sensitive to δ than $\kappa(A_{mm})$; hence there is an additional need to completely factor A_{vv} in order to ensure good convergence of the PCG method, quite apart from achieving independence of h. The graphs also suggest that the cost of the extra iterations incurred by using

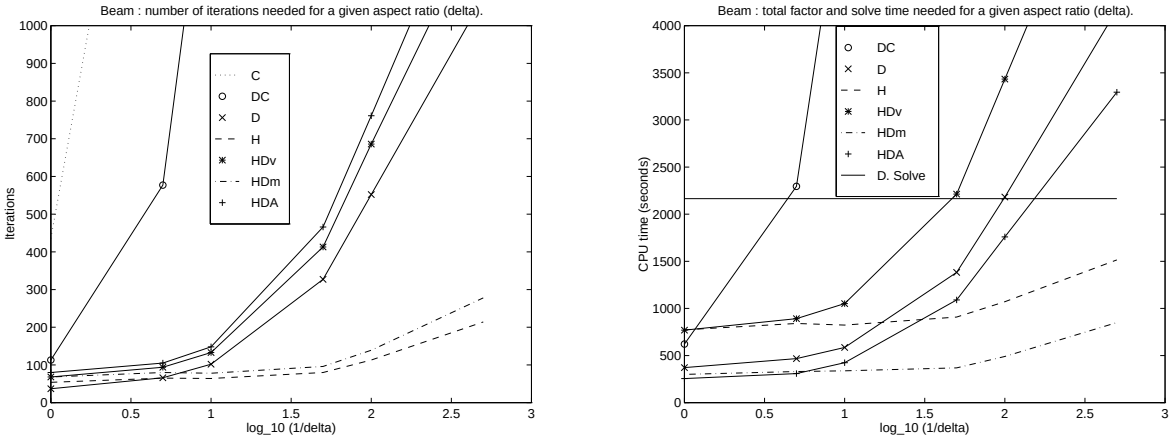

Figure 4: *Graphs of Iterations .vs. Aspect Ratio and Time .vs. Aspect Ratio for the beam problem using the various reduction techniques when completely factoring the reduced matrix.*

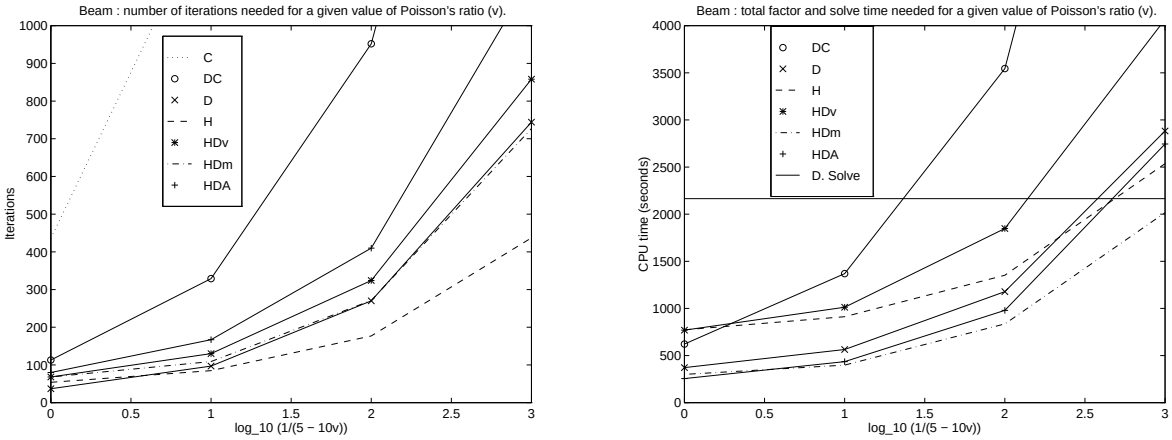

Figure 5: *Graphs of Iterations .vs. Poisson's Ratio and Time .vs. Poisson's Ratio for the beam problem, using the various reduction techniques when completely factoring the reduced matrix.*

HD_m reduction, as compared to H reduction, is compensated by the savings accrued whilst computing the preconditioner and performing the forward-back solve step each iteration. The results obtained from varying Poisson's ratio are similar, although we notice that D reduction performs relatively better.

We see that using C reduction, with or without D reduction, gives very poor results, even for well-conditioned problems. Part of the cause can be attributed to the fact that the reduced matrix B_C is not spectrally equivalent to the stiffness matrix A

Table 3: *Performance of H and HD_m reduction for the beam problem with varying aspect ratio δ, when incompletely factoring the reduced matrix. The incomplete factorisation applies to the midside node domain only; a complete factorisation of the vertex node domain was used in all cases. The number of factoring failures due to negative pivots, if any, is given in parentheses beside the number of iterations.*

H reduction	**CPU seconds (iterations)**				
Drop tol. ϵ	$\delta = \frac{1}{5}$	$\delta = \frac{1}{10}$	$\delta = \frac{1}{50}$	$\delta = \frac{1}{100}$	$\delta = \frac{1}{500}$
1	227 (75)	361 (142)	689 ($301^{(7)}$)	900 ($409^{(7)}$)	$*^{(4)}$
10^{-1}	234 (75)	370 (146)	542 ($224^{(8)}$)	1379 ($627^{(7)}$)	$*^{(7)}$
10^{-2}	244 (77)	348 ($123^{(10)}$)	596 ($238^{(11)}$)	1474 ($656^{(11)}$)	$*^{(11)}$
10^{-3}	291 (66)	307 (77)	457 ($159^{(8)}$)	864 ($349^{(7)}$)	$*^{(6)}$
10^{-4}	446 (65)	419 (64)	482 ($113^{(2)}$)	625 ($191^{(4)}$)	$*^{(3)}$
10^{-5}	622 (65)	607 (64)	539 (81)	841 (184)	2311 ($836^{(1)}$)
10^{-6}	736 (65)	722 (64)	711 (80)	761 (121)	2556 ($774^{(1)}$)
0	840 (65)	823 (64)	908 (80)	1071 (113)	1515 (214)

HD_m reduction	**CPU seconds (iterations)**				
Drop tol. ϵ	$\delta = \frac{1}{5}$	$\delta = \frac{1}{10}$	$\delta = \frac{1}{50}$	$\delta = \frac{1}{100}$	$\delta = \frac{1}{500}$
1	213 (82)	211 (81)	243 (99)	329 (145)	669 (341)
10^{-1}	213 (82)	211 (81)	383 ($170^{(6)}$)	802 ($404^{(7)}$)	$*^{(7)}$
10^{-2}	212 (80)	217 (82)	559 ($252^{(11)}$)	1260 ($660^{(11)}$)	$*^{(11)}$
10^{-3}	227 (80)	221 (78)	337 ($128^{(5)}$)	660 ($308^{(6)}$)	$*^{(6)}$
10^{-4}	251 (80)	244 (78)	269 (99)	410 (177)	$*^{(1)}$
10^{-5}	278 (80)	270 (78)	286 (96)	382 (147)	1649 ($797^{(1)}$)
10^{-6}	306 (80)	293 (78)	313 (96)	394 (140)	734 (333)
0	330 (80)	338 (78)	369 (96)	490 (139)	850 (278)

* Failed to converge within 1000 iterations.

with respect to the number of unknowns N, and thus the number of iterations depends on the problem size. However, DC reduction also gives poor results relative to the other methods (albeit markedly better than C reduction alone) even though $\kappa(B_{DC}^{-1}A)$ is independent of N, leaving us to conclude that it is the C reduction step that is causing the slow convergence. One reason for this could be that the amount added to the diagonals is too large in some sense, causing the same sort of performance deterioration that is sometimes encountered when a large perturbation is made to ensure positive pivots. Also, the advantages of C reduction involve using modified ILU schemes which lower the largest eigenvalue λ_{max} and thus control κ_F [3]. However, results in [19] indicate that as $\delta \to 0$ or $\nu \to 0.5$, it is the smallest eigenvalue that changes the most, getting increasingly closer to zero. The poor convergence of C-reduction has also been reported by Hladík et al in [2].

So far, it seems that the hierarchical techniques H and HD_m reduction are the

most successful for problems with poor aspect ratios or extreme values of ν, whilst HD_A reduction has some merit for larger, better-conditioned problems. To further compare H and HD_m reduction, we present results using incomplete factorisations of the reduced matrix in Table 3. For a given drop tolerance, HD_m reduction gives a lower overall solution time for all aspect ratios, and in many cases, the number of iterations is also lower, particularly as the aspect ratio worsens. We note that for the very poor aspect ratio problems, the number of factoring failures (and hence the number of perturbations to the original matrix) has a more significant effect on the number of iterations needed than the drop tolerance itself, with the fastest convergence obtained when the fewest perturbations are needed. This supports the theory given earlier, that for ill-conditioned problems, diagonal perturbations must be kept to a minimum, and explains the success of the coarse drop tolerance factorisation for $(A_{mm})_D$ using $\epsilon = 1$. We see that HD_m reduction generally resulted in fewer factoring failures, which concurs with the stabilising properties of D reduction claimed by Shlafman and Efrat in [20]. We conclude from the results in this section that HD_m reduction is the best choice for poor aspect ratio problems.

5.2 Results for the larger problems

Figure 6 shows results for the more successful reduction techniques, namely H, HD_m and HD_A reduction, for the larger, more complex problems.

From the graphs in Figure 6, we see that though the best choice of reduction varies from problem to problem, HD_m reduction consistently outperforms H reduction across the range of drop tolerances tested, with only the coarse drop tolerance factorisation of $\epsilon = 1$ used for the washer problem giving better results for H reduction. Again, the occurrence of negative pivots during factorisation was less common when using HD_m reduction, and the amount by which the matrix was perturbed was highly influential. HD_A reduction was not competitive for the poorer aspect ratio problems, but provided the fastest solution for the better-conditioned I-beam problem. We see that for the more ill-conditioned problems, the fastest solutions arise from using a fine drop tolerance factorisation of the midside node domain, and thus a more accurate approximation to B^{-1} for a preconditioner, as was reported in [7].

An interesting feature of the graphs is that they each have a hump in the vicinity of $\epsilon = 10^{-2}$, resulting from an excessive number of factoring failures occurring with this drop tolerance. For the I-beam problem, the convergence rate is not greatly affected, but the time taken for the additional attempts to factor the matrix is significant. For the other problems, the factorisation time is less important, but the iteration time is drastically affected, particularly for the washer problem. For the pipe problem, we notice a rapid upsurge in solution time for H reduction as $\epsilon \to 0$, which is largely spent in computing the now enormous preconditioner (even though no factoring failures occurred). This illustrates one of the principal drawbacks of H reduction, that the preconditioners are very expensive to compute and store for ill-conditioned problems, where fine drop tolerances are necessary. Fortunately, such drop tolerances are available without too

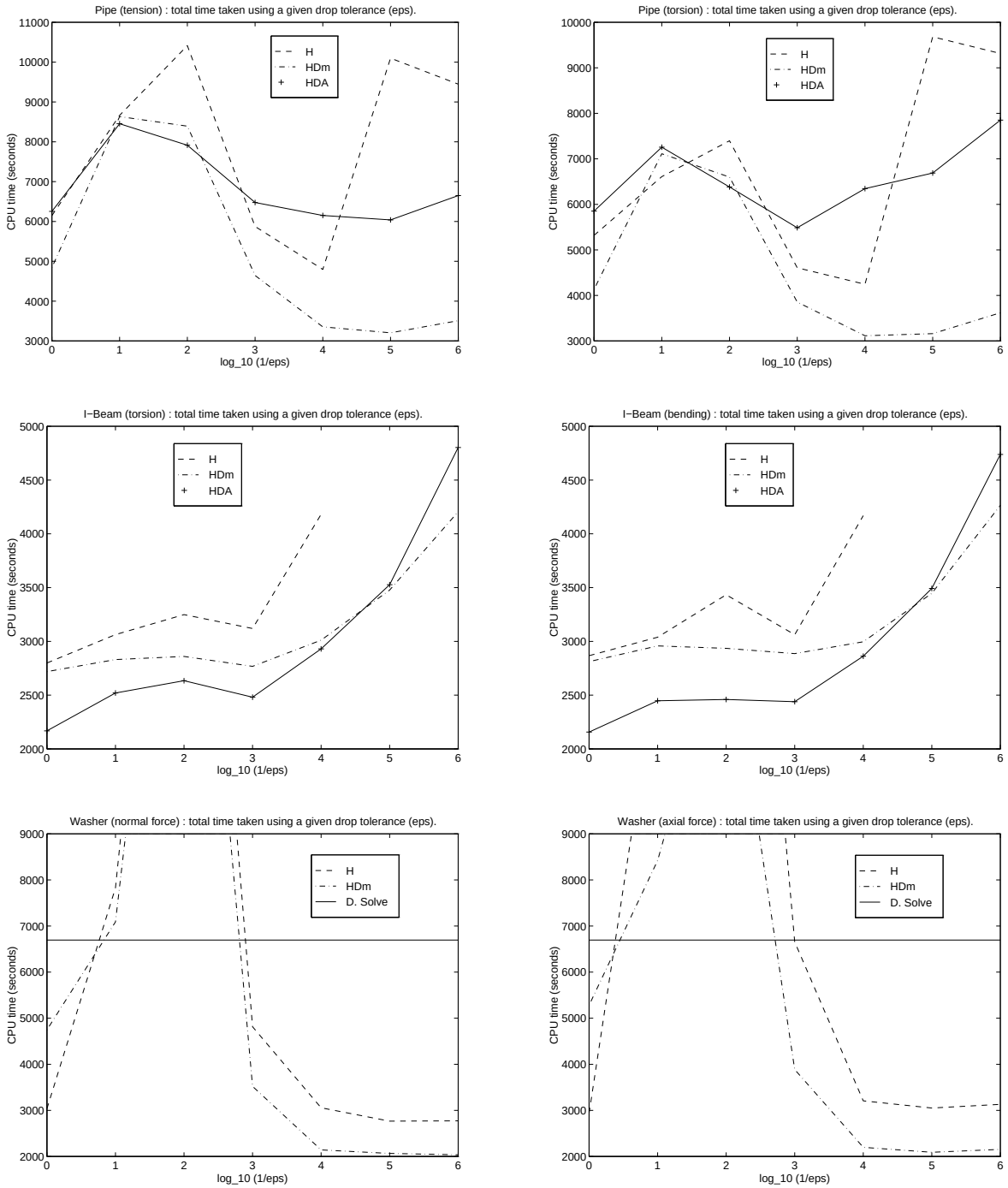

Figure 6: *Graphs of Time .vs. Drop Tolerance for both sets of boundary conditions for the pipe, I-beam and washer problems, respectively, using H, HD_m and HD_A reduction.*

much expense through the use of HD_m reduction.

For the I-beam problem, a coarse factorisation of the midside node domain was sufficient to provide good convergence (notice that for the I-beam problem, there was not enough memory available to store an H reduction preconditioner with $\epsilon < 10^{-4}$). The most time-consuming activity for H and HD_m reduction was the complete factorisation of the vertex node domain, and the fastest solutions in fact arose through using HD_A reduction, in spite of needing many more iterations for convergence. As the size of a given problem increases (with grid refinement, say), this factorisation will dominate the cost of solution for moderate to well-conditioned problems, and hence HD_A reduction is an extremely useful technique in such situations, as well as having considerable memory-saving benefits. We stress that this technique is not applicable for very ill-conditioned problems, however; for the washer problem, convergence was not achieved within 5000 iterations using any preconditioner arising from HD_A reduction, which is therefore absent from the graphs.

We notice that for the washer problem, the performance of direct and iterative solves is comparable; the smaller size and poorer aspect ratio of the problem favouring direct solution methods. We have found this to be indicative of a plateau reached by our preconditioning methods, with respect to the average aspect ratio. A modified version of the washer problem with an aspect ratio twice as poor could not be solved within 5000 iterations using any of the preconditioning techniques mentioned in this paper. We note that the existence of force-specified boundary conditions makes the problem even more difficult; with only force-specified boundary conditions, the stiffness matrix would be singular, and the condition number of the system infinite. By way of comparison, we reformulated the washer problem (with normal loads) to have only displacement-specified boundary conditions (at the nodes which were previously subject to surface forces), obtained by first solving the original washer problem to find the displacements. The maximum relative difference[2] in any direction for the two solutions (solved in double-precision by a direct method) was verified to be less than 1% at any given node, and the solution times for HD_m reduction are compared in Table 4. We found it generally true that iterative methods performed better on problems with displacement-specified boundary conditions, which must always be preferred, given the choice. The different types of loading for each problem did not have a significant impact on the convergence rates, however.

The convergence tolerance used for the PCG solutions varied with each problem, as mentioned earlier. A direct solution was not available for the I-beam problem due to memory restrictions, and even though one was available for the pipe problem, the solution quality (judged by the size of the residual) was very poor. Thus the convergence criteria for these problems have been changed, and a summary is given in Table 5. We found in general that for even moderately poor aspect ratios, it was essential to store the stiffness matrix entries in double-precision, as single-precision storage lead to

[2]This is defined as the maximum absolute difference divided by the maximum absolute displacement, for each direction.

Table 4: *Convergence for the washer problem (normal loads) with equivalent force-specified and distance-specified boundary conditions, using HD_m reduction. The convergence tolerance was set to the residual of the direct solution in each case, including adjustments.*

	CPU seconds (iterations)				
Drop tol. ϵ :	1	10^{-2}	10^{-4}	10^{-6}	0
Force-specified	4747 (1145)	*	2143 (504)	2037 (400)	2665 (412)
Distance-specified	2322 (578)	3680 (907)	1202 (257)	1279 (225)	1693 (224)

* More than 5000 iterations needed.

inaccurate solutions and much higher iteration counts. In some cases, the resulting PCG method did not converge at all, and similar behaviour was witnessed in [24]. Storing just the preconditioner in single-precision presented less of a difficulty, although for very bad aspect ratio problems some idiosyncrasies were apparent, such as the appearance of negative pivots during the direct solve of the pipe problem (thus accounting for the high residual). Iteration counts were also affected : for the washer problem (normal loading) using HD_m reduction, convergence could not be achieved within 5000 iterations using single-precision storage, compared to 412 using double-precision. (For the pipe problem, the maximum difference in using single and double-precision was less than 20 iterations, however.) We note that bad aspect ratios give rise to relatively small off-diagonal entries in the preconditioning matrices, thus the risk of losing necessary information through single-precision storage becomes significant as $\delta \to 0$. We also found that for the washer problem, the residual calculated upon convergence was significantly higher than that returned by the PCG algorithm, even with double-precision storage in use. Thus we lowered the convergence tolerance for each set of boundary conditions, to make the true residuals (calculated from the solutions themselves) comparable to a direct solve.

Table 5: *Convergence tolerance and residual data for the larger problems.*

Problem	**Residual from Direct Solve**	**PCG convergence tolerance used**
Pipe (tension)	$4.97 \cdot 10^{-5}$	10^{-8}‡
Pipe (torsion)	$1.03 \cdot 10^{-3}$	10^{-8}‡
I-Beam (both)	†	10^{-8}‡
Washer (normal)	$1.81 \cdot 10^{-14}$	$1.00 \cdot 10^{-16}$
Washer (axial)	$5.72 \cdot 10^{-15}$	$1.00 \cdot 10^{-17}$

† Insufficient memory available.

‡ l_2 residual reduction

In summary, we can recommend HD_m reduction with a drop tolerance $\epsilon = 10^{-5}$ or 10^{-6} for problems with $\delta \leq \frac{1}{20}$, and HD_A reduction with $\epsilon = 1$ otherwise (espe-

Table 6: *Best performance of each of the reduction techniques for the Pipe (tension), I-Beam (torsion) and Washer (normal force) problems. The drop tolerance used to attain this performance is also given; note that for H, HD_m and HD_A reduction, the incomplete factorisation applies to the midside node domain only.* Memory *is measured as the number of megabytes needed to store the stiffness matrix (in double-precision) and the preconditioner (in single-precision for the I-Beam and Pipe problems, and double-precision for the Washer problem).*

Reduction	**CPU seconds (iterations)**			**Memory in megabytes**		
	Pipe	I-Beam	Washer	Pipe	I-Beam	Washer
H	4,795 (285) $[10^{-4}]$	2,799 (83) [1]	2,766 (357) $[10^{-5}]$	103 Meg	117 Meg	74 Meg
HD_m	3,204 (277) $[10^{-5}]$	2,719 (86) [1]	2,037 (400) $[10^{-6}]$	86 Meg	108 Meg	54 Meg
HD_A	6,039 (820) $[10^{-5}]$	2,168 (252) [1]	**	69 Meg	73 Meg	N/A
D	8,612 (524) [0]	6,683 (277) $[10^{-5}]$	**	124 Meg	158 Meg	N/A
DC	**	**	**	N/A	N/A	N/A
C	**	**	**	N/A	N/A	N/A
None	**	12,198 (342) $[10^{-4}]$	**	N/A	210 Meg	N/A
Direct Solve	13,739	†	6,695	293 Meg	N/A	236 Meg

** More than 14,400 seconds taken.

† Insufficient memory available.

cially for very large, moderately-conditioned problems). HD_m reduction consistently outperformed H reduction in our experiments, and has additional memory-saving attributes, as does HD_A reduction. These are exemplified in Table 6, where the best performance attained by each reduction technique is given (using $\epsilon = 1, 10^{-1}, \cdots, 10^{-6}$ and 0), accompanied by the memory requirements for each solution. We see clearly that hierarchical basis techniques are by far the best choice, saving both time and memory in comparison to the other methods. Techniques involving the reduction of off-diagonal entries (C-reduction) did not produce good preconditioners, for any of the problems tested. Indeed, we note that for the very ill-conditioned washer problem, only hierarchical basis preconditioners could compete with a direct solve. The superiority of these techniques over standard ILU factorisation is also noteworthy.

6 CONCLUSIONS

The main conclusions of this work can be summarised as follows :

- For problems with poor element aspect ratios, preconditioners based on a hierarchical decoupling of the stiffness matrix (H and HD_m reduction) give the best results. The combination of hierarchical and degree of freedom decoupling is particularly successful. Note that HD_m reduction (hierarchical decoupling followed by degree of freedom decoupling of the midside nodes, with a complete factorisation of the vertex node domain, and an incomplete factorisation of the midside node domain) is always faster than standard H reduction, and requires less memory.

- For larger problems with better element aspect ratios, HD_A reduction (hierarchical decoupling followed by degree of freedom decoupling of both midside and vertex node domains) works very well, requiring only 60% - 70% of the memory needed for H reduction.

- For extremely poor aspect ratio problems, the amount added to the diagonal to ensure positive pivots is crucially important, and more influential than the value of the drop tolerance. Adding the minimum amount to the diagonal can save large amounts of time during the iterative phase.

In this article, we sought to find the limits of applicability of the iterative methods discussed, with respect to the condition number of the stiffness matrix. A robust technique (i.e. one which works well for problems with poor aspect ratios) consists of hierarchical decoupling, and requires a fairly small drop tolerance for the reduced midside node domain, and a direct solve of the vertex node domain. For problems with element aspect ratios $\delta \geq \frac{1}{100}$, the above choices of preconditioner were quite effective, and the resulting PCG methods were faster than direct methods. However, for problems with extreme aspect ratios ($\delta < \frac{1}{100}$), it must be admitted that the iterative methods used in this work did not always perform satisfactorily. Of course, there are also limitations in terms of problem size, under which our methods fare well in comparison to direct methods, which can quickly exceed the physical capabilities of a given machine.

The decision of which type of method to use, therefore, should be made according to both the size of the problem and the condition number of the resulting stiffness matrix. Given that most practical problems arising in industry are likely to be better conditioned than those solved in this work, and will often be larger, we can comfortably recommend the use of iterative solvers in general.

References

[1] J. Fish and R. Guttal. Adaptive solver for the p-version of the finite element method. *Int. J. Num. Meth. Eng.*, 40:1767–1784, 1997.

[2] I. Hladí k, M.B. Reed, and G. Swoboda. Robust preconditioners for linear elasticity FEM analyses. *Int. J. Num. Meth. Eng.*, 40:2109–2127, 1997.

[3] P. Saint-Georges, G. Warzee, R. Beauwens, and Y. Notay. High-performance PCG solvers for FEM structural analysis. *Int. J. Num. Meth. Eng.*, 39:1313–1340, 1996.

[4] M.C. Dracopoulos and M.A. Crisfield. Coarse/fine mesh preconditioners for the iterative solution of finite element problems. *Int. J. Num. Meth. Eng.*, 38:3297–3313, 1995.

[5] V.E. Bulgakov and G. Kuhn. High performance multilevel iterative aggregation solver for large finite element structural analysis problems. *Int. J. Num. Meth. Eng.*, 38:3529–3544, 1995.

[6] M. Suarjana and K.H. Law. A robust incomplete factorization based on value and space constraints. *Int. J. Num. Meth. Eng.*, 38:1703–1719, 1995.

[7] J.K. Dickinson and P.A. Forsyth. Preconditioned conjugate gradient methods for three-dimensional linear elasticity. *Int. J. Num. Meth. Eng.*, 37:2211–2234, 1994.

[8] E.F. D'Azevedo, P.A. Forsyth, and W.-P. Tang. Towards a cost effective ILU preconditioner with high level fill. *BIT*, 32:442–463, 1992.

[9] A. Jennings and G.M. Malik. Partial elimination. *J. Inst. Maths. Appl.*, 20:307–316, 1977.

[10] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element method.* McGraw-Hill, London, fourth edition, 1989.

[11] E.F. D'Azevedo, P.A. Forsyth, and W.-P. Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM. J. Matrix Anal. Applic.*, 13:944–961, 1992.

[12] M.A. Ajz and A. Jennings. A robust incomplete choleski conjugate gradient algorithm. *Int. J. Num. Meth. Eng.*, 20:949–966, 1984.

[13] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, New York, 1994.

[14] R. Beauwens and R. Wilmet. Conditioning analysis of positive definite matrices by approximate factorizations. *J. Comput. Appl. Math.*, 26:257–269, 1989.

[15] R. Beauwens. Modified incomplete factorisation strategies. In O. Axelsson and L Kotolina, editors, *PCG Methods*, number 1457 in Lecture Notes in Mathematics, pages 1–16, Berlin, 1990. Springer.

[16] T.A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Math. Comp.*, 34(150):473–497, 1980.

[17] O.C. Zienkiewicz, J.P. de S.R. Sago, and D.W. Kelly. The hierarchical concept in finite element analysis. *Comp. Struct.*, 16(1):53–65, 1983.

[18] O. Axelsson and I. Gustafsson. Preconditioning and two-level multigrid methods of arbitrary degree of approximation. *Math. Comp.*, 40:219–242, 1983.

[19] M. Jung, U. Langer, and U. Semmler. Two-level hierarchically preconditioned conjugate gradient methods for solving linear elasticity finite element equations. *BIT*, 29:748–768, 1989.

[20] S. Shlafman and I. Efrat. Using Korn's inequality for an efficient iterative solution of structural analysis problems. In R. Beauwens and P. de Groen, editors, *Iterative Methods in Linear Algebra*, pages 575–581, North-Holland, 1992. Elsevier Science.

[21] C. Farhat and N. Sobh. A coarse/fine preconditioner for very ill-conditioned finite element problems. *Int. J. Num. Meth. Eng.*, 28:1715–1723, 1989.

[22] J.W.H. Liu. Modification of the minimum-degree algorithm by multiple elimination. *ACM Transactions on Mathematical Software*, 11(2):141–153, 1985.

[23] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices.* Oxford University Press, Oxford, 1986.

[24] M. Papadrakakis and N. Bitoulas. Accuracy and effectiveness of preconditioned conjugate gradient algorithms for large and ill-conditioned problems. *Comp. Meth. Appl. Mech. Eng.*, 109:219–232, 1993.