AIAA AVIATION Forum
June 15-19, 2020, VIRTUAL EVENT
AIAA AVIATION 2020 FORUM

10.2514/6.2020-3022

Node Numbering for Stabilizing Preconditioners Based on Incomplete LU Decomposition

W. Kyle Anderson,* Stephen L. Wood,† Kevin E. Jacobson,‡
NASA Langley Research Center, Hampton, VA 23681, USA

Matrix-reordering strategies that reduce the occurrence of instabilities often associated with preconditioners based on incomplete LU decomposition are presented. Several example matrices, extracted from turbulent-flow simulations, and from structural mechanics, are used to demonstrate the occurrence of instabilities and to investigate the underlying cause. The current work demonstrates that appropriate schemes for reordering unknowns can provide a very effective strategy for stabilizing the preconditioner, thereby enabling a Krylov-subspace method to solve linear systems that were heretofore unsolvable. Several new numbering strategies are presented, one of which provides a tunable parameter to control the bandwidth of the renumbered matrix. Using the discussed methodology greatly increases the robustness and reliability of the simulations.

I. Introduction

The need to solve large linear systems of equations is ubiquitous in physics-based simulations based on finite-element, finite-volume, and finite-difference methods. In many of these simulations, iterative linear solvers based on Krylov-subspace methods (e.g. Refs. [1–6]), are used because of their verifiable mathematical properties and demonstrated robustness. Independent of the particular Krylov-subspace method, preconditioners are inevitability used to reduce the number of search directions required to reach convergence. One technique that is often used as a preconditioner is based on an incomplete lower/upper (LU) decomposition of the original matrix to form an approximate inverse [7]. For this technique, the size of the nonzero entries in the original matrix are assumed to be order ϵ^1, where ϵ is a small value, and all other elements are initially assumed to be ϵ^∞ (zero). The matrix is then symbolically decomposed, using row operations, into lower and upper contributions, during which the relative size of the entries that represent fill are estimated in terms of ϵ^k, and terms smaller than a preset power of k are dropped. Using this technique, the structure of the matrix used for preconditioning is determined as a preprocessing step without consulting the actual values in the matrix. This technique is extremely common among practitioners and is often quite reliable. However, for many applications, the underlying assumptions are simply not accurate. Furthermore, because the structure of the lower and upper factors is determined as a preprocessing step, pivoting during LU decomposition is not performed, except when a block structure is maintained and pivoting may be used during the inversion of a block. Unfortunately, as nicely illustrated in Refs. [8–10], pivoting plays an important role in maintaining stability for elimination, as well as backsubstitution. As such, one might expect that forgoing pivoting may lead to potential adverse consequences.

In Ref. [11], Chow provides a thorough discussion on the reliability of preconditioners based on incomplete LU decomposition, and describes various means by which these preconditioners fail. Prevalent among the failure modes discussed in Ref. [11] is the instability of the backsubstitution phase, which is often characterized by long recurrences. Many widely used algorithms for numbering unknowns in a mesh, [12–27], are specifically designed to cluster entries along the diagonal. When using these techniques in conjunction with incomplete LU decomposition, the hope is that smaller bandwidths imply fewer discarded unknowns during the factorization, thereby providing a more accurate approximate inverse that translates into fast and robust convergence. To verify this behavior, numerous studies have been conducted to compare ordering strategies for incomplete LU decompositions to identify methodologies that are favorable in terms of convergence [28–33]. Among the methods often included in these studies, the Cuthill-McKee [19] and reverse Cuthill-McKee [18] strategies repeatedly rank among the best performers in terms of reduced search directions and rapid convergence.

Unfortunately, as demonstrated later, the strategy of minimizing bandwidth can also promote instability of both the elimination and backsubstitution phases for matrices that are not diagonally dominant. The purpose of the current work

*Senior Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
†Research Scientist, Computational AeroSciences Branch, AIAA Senior Member.
‡Research Aerospace Engineer, Aeroelasticity Branch, AIAA Member.

is to demonstrate the role that ordering the unknowns has on stability, and to explore means for systematically ordering the variables to improve the reliability of incomplete LU decomposition and backsubstitution. The particular type of instability being considered is demonstrated and rationale underlying new ordering strategies is discussed. Several algorithms for reordering are then presented, and results are shown to demonstrate the effectiveness of the procedures.

II. Problem Description

Before discussing node ordering strategies, an initial example is first presented to illustrate the type of instability that is the motivation of the current investigation. The particular demonstration is for turbulent flow over a simple bump configuration similar to that shown in Fig. 1. The freestream Mach number for this simulation is $M_\infty = 0.2$ and the Reynolds number, based on the chord of the bump, is 3×10^6. The unknowns in the mesh, which is comprised of $1,351,680$ tetrahedral elements and $242,729$ grid points, have been ordered using a reverse Cuthill-McKee strategy [18] to reduce the bandwidth. The solution to the compressible Navier-Stokes equations, strongly coupled with the Spalart-Allmaras turbulence model [34], is obtained using the stabilized finite-element flow solver described in Ref. [35]. To solve the nonlinear equations, a Newton-type approach is used, where the linear system at each step is solved using GMRES [1] and an ILU(k) preconditioner similar to that described in [7]. Consistent with other implementations of ILU(k), no pivoting is performed between rows. However, because six variables are solved at each grid point, a block structure is maintained so that each row is actually comprised of six scalar rows. During LU decomposition, while pivoting is not used between (blocked) rows, pivoting is performed within the 6×6 diagonal blocks as they are inverted.

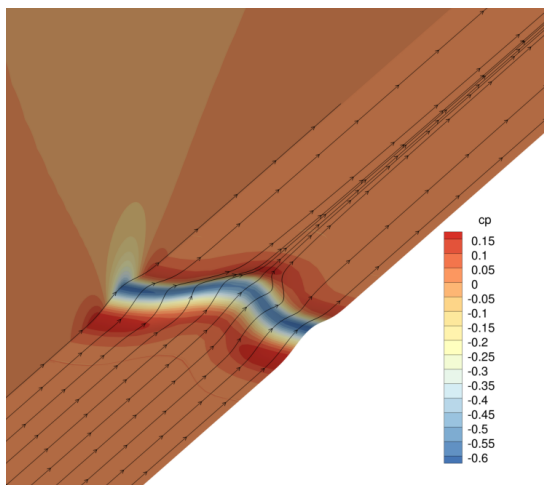
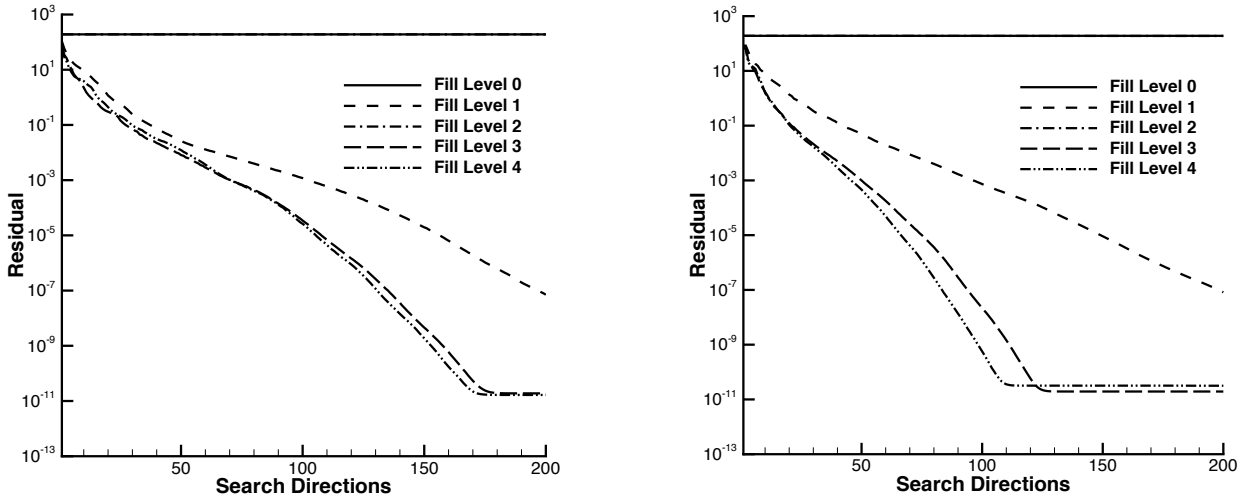


Fig. 1 Pressure contours for flow over subsonic bump: $M_\infty = 0.2$**,** $Re = 3 \times 10^6$**.**

The matrix for this example was extracted from a multiprocessor simulation at a particular iteration where the linear solver was unable to reduce the residual except by very small amounts. Specifically, the output from GMRES [1], after several hundred search directions using reverse Cuthill-McKee for ordering, and ILU(2) as the preconditioner, indicated that the ratio of the final residual relative to the initial residual was only reduced to slightly less than unity. Attempting to use other fill levels resulted in erratic behavior, in that convergence could be not be achieved using ILU(0) or ILU(2), but could be obtained using ILU(1), ILU(3), and ILU(4). Because using the entire matrix would be cumbersome for unit testing, the matrix used for preconditioning each partition was written to a file and convergence was examined for each partition individually. Here, it was discovered that the linear systems on all partitions converged quite readily except for one. The behavior of the linear system on this single partition accurately reproduced the behavior observed during the full simulation, and was therefore saved for later study. The matrix reflected the set of unknowns resident on the partition, which contained $30,619$ grid points, with 6 variables stored at each node, for a total of $183,714$ unknowns. Of the $183,714$ total rows, only $5,323$ were diagonally dominant. Because the solution procedure used in the current study takes advantage of the block structure, the Frobenius norm of the entries was also calculated, where only 23 of the $30,619$ rows were diagonally dominant.

The convergence histories using ILU(0)-ILU(4) are shown in Fig. 2, where results are included using both Cuthill-McKee [19] and reverse Cuthill-McKee [18] orderings. In the figures, the residual depicted for monitoring convergence

is the estimate from the least squares problem in GMRES. For both orderings, ILU(0) and ILU(2) fail to reduce the residual, as evidenced by the nearly horizontal lines in the figure. In fact, while the L_2 norm of the residual estimated from solving the least-squares problem indicates a very slight decrease, the actual L_2 norm of the residual, directly computed as a post-processing step, increases by several orders of magnitude, depending on the fill level. In addition to the disappointing convergence, this result also illustrates that the often quoted feature that GMRES reduces the residual with each additional search direction is a property of the algorithm that assumes exact arithmetic and is not necessarily indicative of the actual behavior when roundoff error may significantly influence the results. The root of the problem is that the instability in the preconditioner results in values so large, that orthogonalization against previous vectors in the Krylov subspace is not achievable because of subtractive cancellation error.



(a) Cuthill-McKee

(b) Reverse Cuthill-McKee

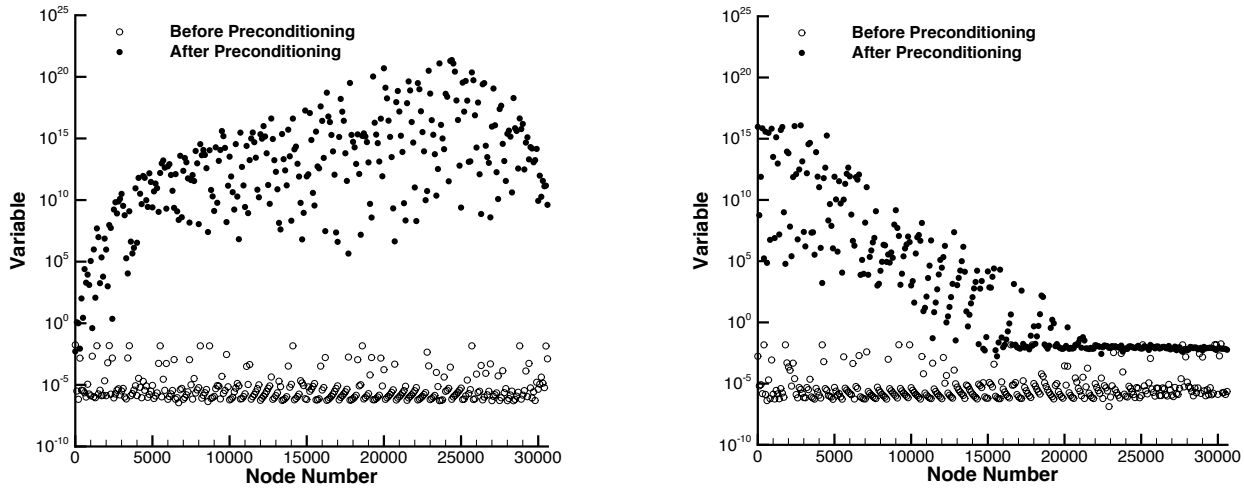
Fig. 2 GMRES convergence for Cuthill-McKee and reverse Cuthill-McKee orderings.

To verify that the poor behavior for this case is not implementation dependent, results are shown in Table 1 obtained using three separate GMRES implementations [36–38]. The current implementation, used in most of the results later presented, is based on Ref. [36], and is denoted in the tables simply as "GMRES". Here, results are shown both with and without two additional orthogonalization steps. Conversely, SLAT [37], dynamically monitors orthogonalization as the computations proceed, and automatically applies orthogonalization as needed. As seen in the table, the estimated residual (denoted as (E)) obtained using an ILU(0) preconditioner with all implementations, decreases very slightly from its initial value of approximately 191, to about 190.4. However, direct computation of the actual L_2 norm of the residual (denoted as (A)), indicates that with all implementations, the residual actually increases by as much as a factor of 29. Similar poor results are obtained for ILU(2), where all implementations again fail to reduce the residual, this time increasing it by as much as six orders of magnitude. In contrast, the residuals obtained using ILU(1) and ILU(3) are all reduced, and the estimated and actual residuals agree very closely. Although not shown, results obtained using reverse Cuthill-McKee are very similar.

To confirm that the erratic convergence is attributable to the preconditioner, the elements of the Krylov vector on the first preconditioned search direction have been examined before and after applying the preconditioner using ILU(0). As seen in Fig. 3, the magnitude of the variables before applying the preconditioner are on the order of 10^{-5}, whereas the variables after applying the preconditioner are on the order of 10^{25} for Cuthill-McKee and 10^{15} for reverse Cuthill-McKee. Because the variables after applying the preconditioner ideally represent changes in the dependent variables, the large values occurring after applying the preconditioner are indicative of instabilities in either the LU decomposition or the backsubstitution. Experiments using 64-bit and 128-bit arithmetic reveal that for both Cuthill-McKee and reverse Cuthill-McKee, the elimination phase is unstable, as indicated by very large relative differences in the lower and upper factors depending on the number of significant digits used in the calculations. Specifically, the root mean square (RMS) of relative differences between 64-bit and 128-bit computations is 0.7816×10^3 for Cuthill-McKee, and 0.1675×10^4 for reverse Cuthill-McKee. The maximum relative differences for these same computations are 0.2286×10^7 and 0.3495×10^7 for Cuthill-McKee and reverse Cuthill-McKee, respectively.

Table 1 Final residuals for Cuthill-McKee ordering (initial residual = 0.1910×10^3).

		ILU(0)	ILU(1)	ILU(2)	ILU(3)
GMRES	(E)	0.1904×10^3	0.7106×10^{-7}	0.1907×10^3	0.1889×10^{-10}
	(A)	0.3466×10^3	0.7106×10^{-7}	0.1975×10^5	0.1814×10^{-9}
GMRES w/ortho	(E)	0.1904×10^3	0.7106×10^{-7}	0.1906×10^3	0.1581×10^{-12}
	(A)	0.1017×10^4	0.7106×10^{-7}	0.4476×10^5	0.1818×10^{-9}
SLAT	(E)	0.1904×10^3	0.7986×10^{-7}	0.1907×10^3	0.2480×10^{-11}
	(A)	0.1537×10^4	0.7986×10^{-7}	0.2362×10^6	0.1751×10^{-9}
PETSC	(E)	0.1903×10^3	0.7106×10^{-7}	0.1907×10^3	0.4651×10^{-12}
	(A)	0.5579×10^4	0.7106×10^{-7}	0.1061×10^9	0.1876×10^{-9}



(a) Cuthill-McKee (b) Reverse Cuthill-McKee

Fig. 3 Variables before/after preconditioning.

III. Error Propagation

To motivate the ordering strategies that will be proposed, two complementary points of view are presented. The first approach is to note that in a system of linear equations, each row of the matrix represents a hyperplane, with each entry along every row representing the component of a vector normal to the hyperplane [9, 10]. During forward elimination, the matrix is transformed through a series of row operations to reduce the system of equations to an equivalent set, whose structure is amenable to easy solution by forward/backward substitution.

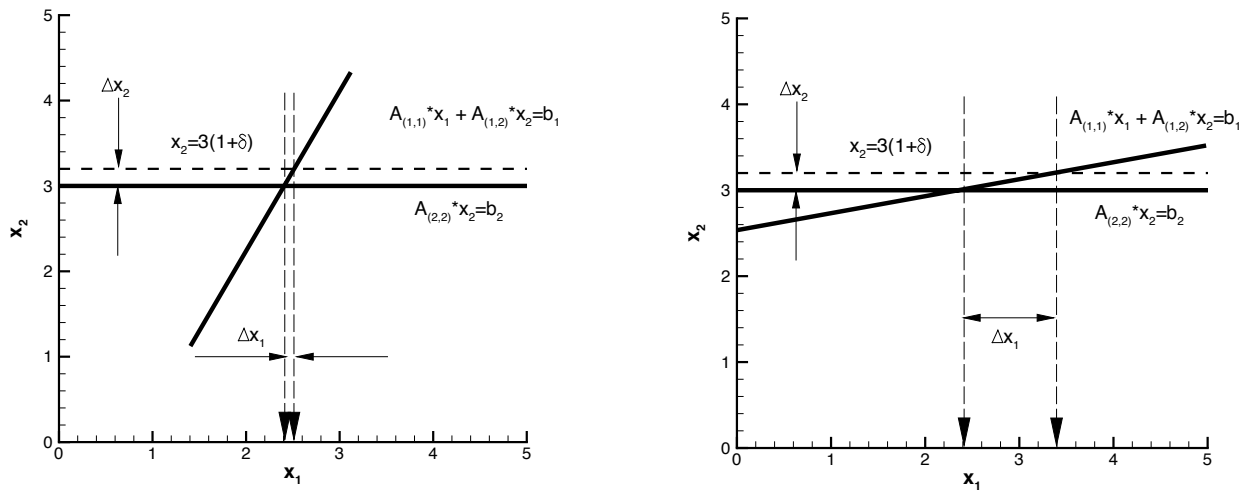
An important aspect of the solution process is the propagation of errors during the LU decomposition and backsubstitution phases. Errors introduced at any point in either phase may be magnified or damped during subsequent operations, depending on the orientation of the hyperplanes. Nominally, if the angle between hyperplanes is greater than 45°, a previously introduced error will diminish, whereas for angles smaller than 45°, the error will magnify.

To graphically illustrate this point, Fig. 4 depicts backsubstitution for a simple 2×2 system of equations, where Fig. 4(a) and Fig. 4(b) depict the cases with well-oriented and poorly-oriented hyperplanes, respectively. In each case, x_2 is first obtained by simply dividing b_2 by $A_{2,2}$. In the scenario presented, the exact solution for x_2, without roundoff error, is assumed to be 3.0, as illustrated by the solid horizontal line. However, in the presence of roundoff error, an approximate solution, given as $3 * (1 + \delta)$, is the actual result of the computation and is shown as the dotted horizontal line in the figure. The error is represented by the total difference between the exact and actual solutions for x_2, and is designated as Δx_2. If one now assumes exact arithmetic for subsequent computation, the solution for x_1 occurs at the intersection of the horizontal dotted line with the oblique solid line, which represents the second equation. Even

assuming exact arithmetic, an error in x_1, denoted at Δx_1, is inherited from the computation of x_2, as can be seen in Fig. 4(a). The issue is to determine whether the inherited error will magnify, or be damped, as the solution process continues. In Fig. 4(a), which depicts well-oriented hyperplanes, the error is reduced, whereas for poorly-oriented hyperplanes, as depicted in 4(b), the error is clearly magnified.

One means for controlling the order of computation so that sequential operations occur on well-oriented hyperplanes is through pivoting [7, 39]. However, for preconditioners based on incomplete LU decomposition, the matrix structure is predetermined without consideration to the actual values in the matrix, thereby precluding pivoting. An important observation on the relative orientation of hyperplanes is that rows in the matrix that do not have common nonzero entries in the same columns are, in fact, orthogonal. Therefore, by numbering the unknowns to minimize commonality between columns, well-oriented hyperplanes can be obtained without ever considering the actual values in the matrix. Because the matrix is sparse, considerable latitude exists for numbering the nodes to achieve this objective.

One should note, however, that the underlying connectivity of the mesh determines how many rows in any column are nonzero. As such, reordering the unknowns does not modify the underlying connectivity, but can introduce significant gaps between subsequent rows that share entries in any given column. By exploiting this principle, it should be possible to dampen errors between rows, instead of having them systematically accumulate as computations are conducted row-by-row. In fact, as illustrated in Fig. 4(a), with well-oriented hyperplanes, an error introduced during a floating point computation can be expected to actually diminish as the row operations are performed. A quantitative example using a nonsparse 4×4 system of equations is provided in Ref. [10].



(a) Well-aligned hyperplanes

(b) Poorly-aligned hyperplanes

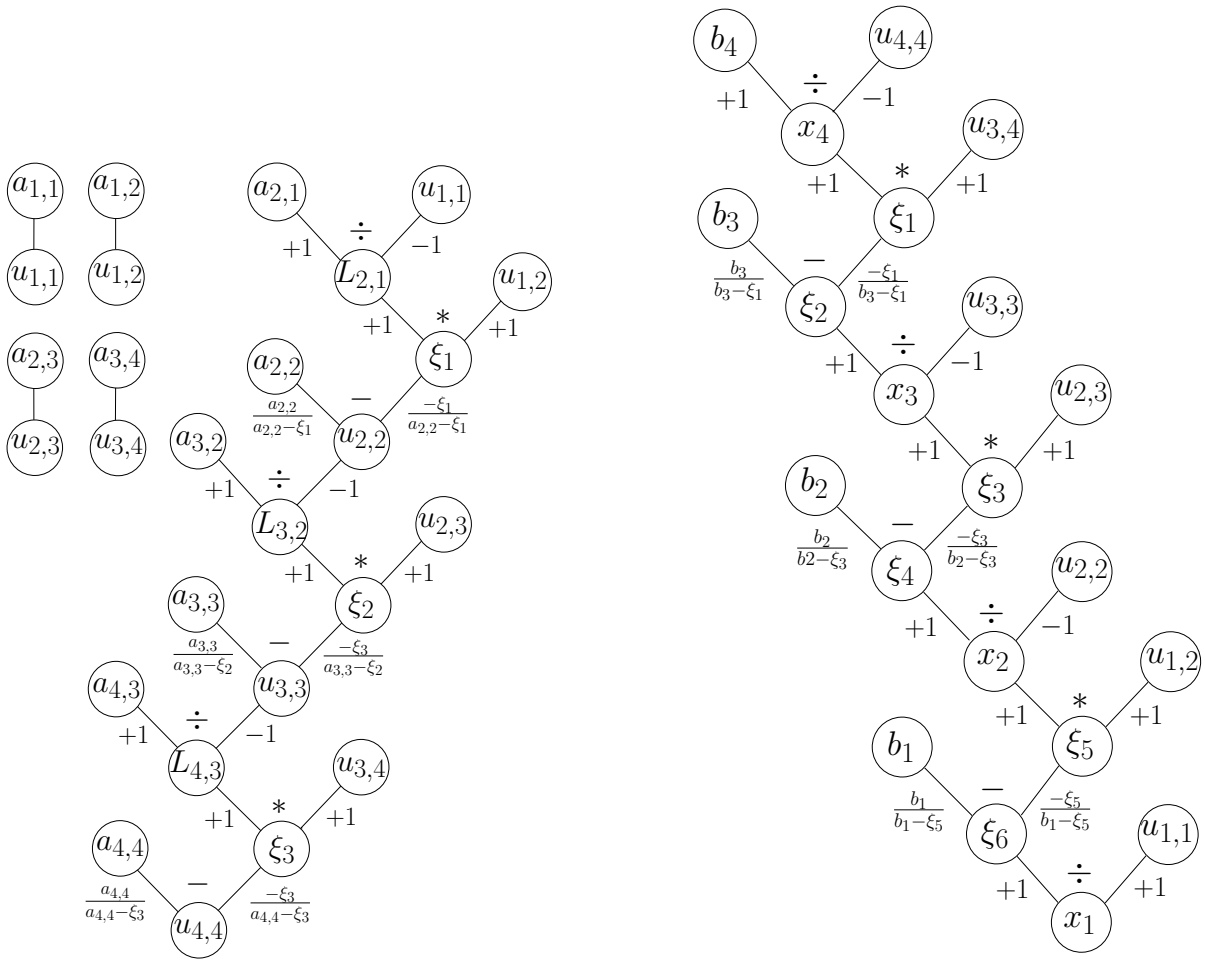
Fig. 4 Well-oriented and poorly-oriented hyperplanes.

Complementary insight into node ordering to reduce instability can be obtained through use of the chain rule to determine how an error, introduced at any point in the computation, affects subsequently-computed values. Consider the matrix in Eq. (1), which represents the structure that would result from a one-dimensional finite-element analysis, where the residual at any node depends solely on its nearest neighbors, and the unknowns are numbered sequentially from left to right using natural ordering. As seen in Eq. (1), the matrix has a tridiagonal structure. Although in this case an incomplete LU decomposition is exact, the matrix structure is useful for illustrative purposes. In particular, while the ordering depicted in Eq. (1) is advantageous from the point of view of minimizing bandwidth, its effect on propagation of errors may be quite different.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} \\ 0 & 0 & a_{4,3} & a_{4,4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \tag{1}$$

Insight into the propagation of errors can be gleaned from the computational graphs, shown in Figs. 5(a) and 5(b), which illustrate relative error propagation during LU decomposition and backsubstitution, respectively, using the analysis

methodologies described in Refs. [40, 41]. The nodes in the graphs indicate inputs, (b_i, $a_{i,j}$), outputs, (x_i, $L_{i,j}$, $u_{i,j}$) or intermediate computed values, (ξ_i), whereas the lines between the nodes are delineated by the condition numbers, which indicate how relative round-off errors amplify or decay during each arithmetic operation [40]. The sensitivity of relative error at any output variable that is attributable to errors at any other variable is obtained by multiplying the condition numbers along the paths between the relevant input and output nodes. If more than one path is available, these products are summed across all the paths [40]. The process is a graphical manifestation of the chain rule, so that the error at any output node can be properly attributed to errors originating at any node in the graph.



(a) Computational graph for LU decomposition

(b) Computational graph for backsubstitution

Fig. 5 Computational graphs for LU decomposition and backsubstitution for Eq. (1).

As seen in the figures, long continuous "chains" are present in both scenarios, so that during LU decomposition, round-off errors introduced in the computation of $L_{2,1}$ propagate uninterrupted all the way to $u_{4,4}$. Likewise, during backsubstitution, errors introduced while computing x_4 propagate without a break in continuity all the way to x_1. While long chains are not necessarily indicative of poorly-oriented hyperplanes, they enhance the possibility for errors produced early in the process to propagate through the entire length of the computation, potentially being magnified at several locations along the way. Particular interest in the figures corresponding to locations where subtraction cancellation occurs because, as noted in Ref. [42], "subtractive cancellation brings earlier errors into prominence." Specifically, if the denominator in a condition number is small, one can expect errors to be locally magnified at these points. Because of the long chains, it is conceptually possible that errors accumulate without an upper bound.

For the small matrix being used for illustration, Table 2 demonstrates that the computation of almost every element of the LU decomposition inherits errors that may have been magnified by previous subtractive cancellation errors. To

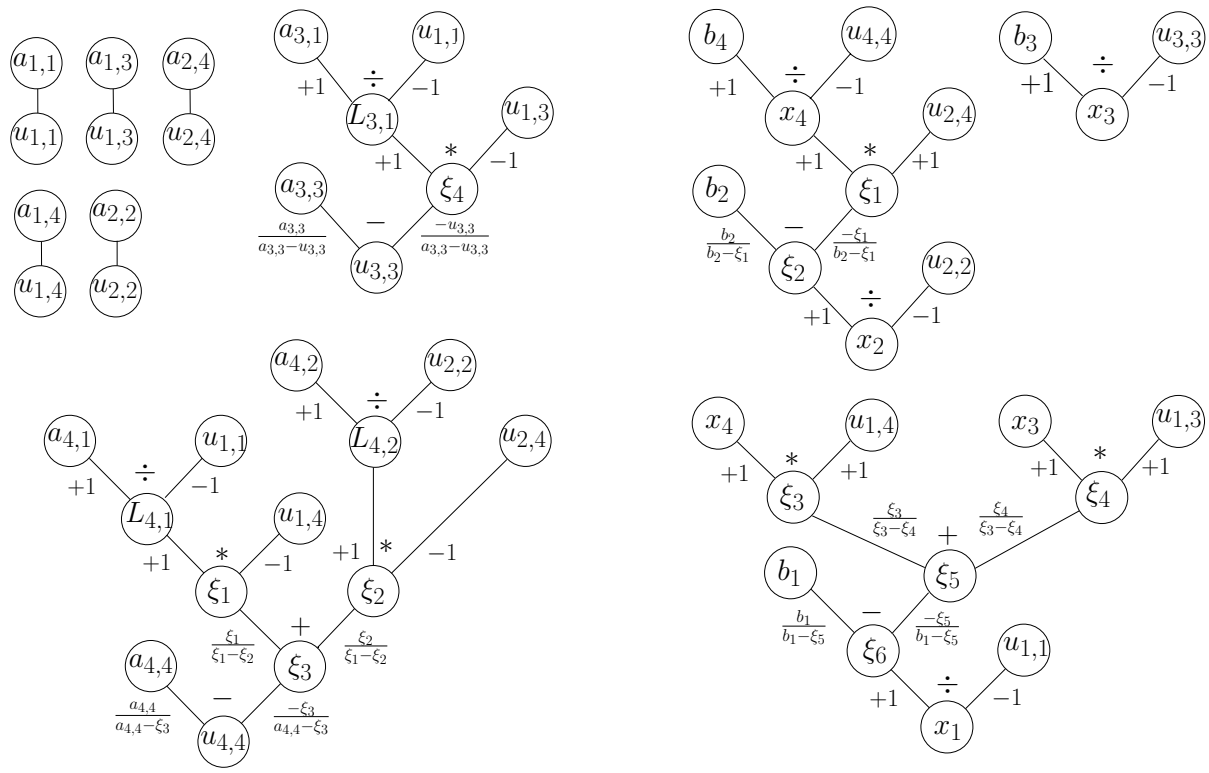
clarify the table, the left column indicates any variable currently being computed, whereas the variables along the top row have been previously computed and are, as such, the sources of propagating round-off error. For example, as seen in Fig. 5(a), $L_{2,1}$ is the first result computed during LU decomposition, and unless the result has no roundoff error, can reasonably be expected to be a source of error in subsequent arithmetic operations. In the later computation of, say, $u_{3,3}$, the error originally attributable to $L_{2,1}$ has potentially been magnified two additional times. As seen in the table, virtually every variable risks using round-off errors that have previously been amplified. Note that while Table 2 represents incomplete LU decomposition, the access patterns for backsubstitution are similar, so that the corresponding table (not shown) has identical values.

Table 2 Number of upstream subtractions for elements of LU decomposition of 1D tridiagonal example.

Variable	$L_{2,1}$	ξ_1	$U_{2,2}$	$L_{3,2}$	ξ_2	$U_{3,3}$	$L_{4,3}$	ξ_3	$U_{4,4}$
$L_{2,1}$	-	-	-	-	-	-	-	-	-
ξ_1	0	-	-	-	-	-	-	-	-
$U_{2,2}$	1	1	-	-	-	-	-	-	-
$L_{3,2}$	1	1	0	-	-	-	-	-	-
ξ_2	1	1	0	0	-	-	-	-	-
$U_{3,3}$	2	2	1	1	1	-	-	-	-
$L_{4,3}$	2	2	1	1	1	0	-	-	-
ξ_3	2	2	1	1	1	0	0	-	-
$U_{4,4}$	3	3	2	2	2	1	1	1	-

As hypothesized earlier, rearranging the nodes to promote favorable orientation of hyperplanes may be beneficial for improving the stability of both LU decomposition and backsubstitution. To illustrate this point, consider now the matrix depicted in Eq. (2). This matrix is again derived from a hypothetical finite-element discretization but, in contrast to the prior matrix, the nodes are ordered from left to right as $(3, 1, 4, 2)$, respectively. The corresponding computational graphs for LU decomposition and backsubstitution are shown in Figs. 6(a) and 6(b), respectively. The long chains that occurred when using the previous ordering (Figs. 5(a) and 5(b)) are now broken into multiple segments, and errors originating upstream of the breaks do not propagate across these boundaries. For comparison purposes, Table 3 shows results similar to those in Table 2. By reordering the points to more evenly distribute nonzero entries throughout the matrix, the number of variables that inherit round-off errors from previous computations is significantly reduced, with most computations inheriting no round-off errors at all.

$$\begin{bmatrix} a_{1,1} & 0 & a_{1,3} & a_{1,4} \\ 0 & a_{2,2} & 0 & a_{2,4} \\ a_{3,1} & 0 & a_{3,3} & 0 \\ a_{4,1} & a_{4,2} & 0 & a_{4,4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \tag{2}$$



(a) Graph for LU decomposition

(b) Graph for backsubstitution

Fig. 6 Computational graphs for random ordering.

Table 3 Number of upstream subtractions for elements of LU decomposition with random ordering.

Variable	$L_{3,1}$	ξ_4	$U_{3,3}$	$L_{4,2}$	ξ_2	ξ_3	$L_{4,1}$	ξ_1	$U_{4,4}$
$L_{3,1}$	-	-	-	-	-	-	-	-	-
ξ_4	0	-	-	-	-	-	-	-	-
$U_{3,3}$	1	1	-	-	-	-	-	-	-
$L_{4,2}$	0	0	0	-	-	-	-	-	-
ξ_2	0	0	0	0	-	-	-	-	-
ξ_3	0	0	0	1	1	-	-	-	-
$L_{4,1}$	0	0	0	0	0	0	-	-	-
ξ_1	0	0	0	0	0	0	0	-	-
$U_{4,4}$	0	0	0	2	2	1	2	2	-

IV. Node Ordering

In section III, the benefits of small chain lengths for producing well-oriented hyperplanes have been explained. To evaluate potential ordering strategies, figures are presented in this section to illustrate the maximum chain lengths associated with specific orderings. For future reference, Fig. 7 and Table 4, demonstrate the means used for presenting later results. Here, Fig. 7 depicts a small mesh comprised of only 10 grid points, whereas Table 4 illustrates the corresponding matrix structure associated with backsubstitution resulting from an incomplete LU decomposition with no fill (ILU(0)). In the table, the numbers along the diagonal are reference values to facilitate quickly identifying each row, and the "x" values denote nonzero entries. Because the example in this figure is for performing backsubstitution, the entries in the lower diagonal are left blank. To the right of the matrix in the table is the longest chain associated with the unknown on each row during backsubstitution. This value is obtained by simply counting upward from each row the number of subsequent rows influenced, directly or indirectly (including the node itself), by an error conceptually introduced during the computation of x_i. Also shown in parenthesis is the ratio of the number of effected rows, divided by the row number where the error is assumed to be introduced. This last value will be referred to in subsequent text as the "chain length." For example, considering row three, an error introduced at this node will not affect the solution of x_2, but will affect the solution of x_1, so that the resulting fraction is 0.667. Values of 1.0 indicate that the maximum possible number of remaining rows during backsubstitution can potentially be vitiated.

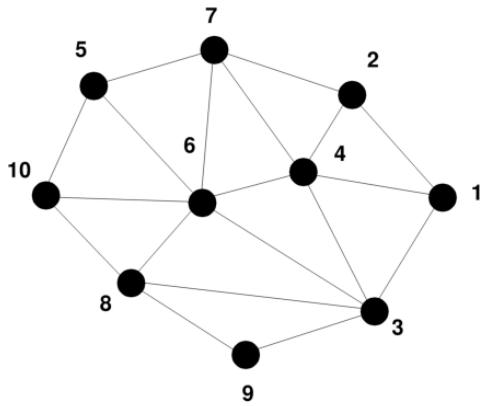


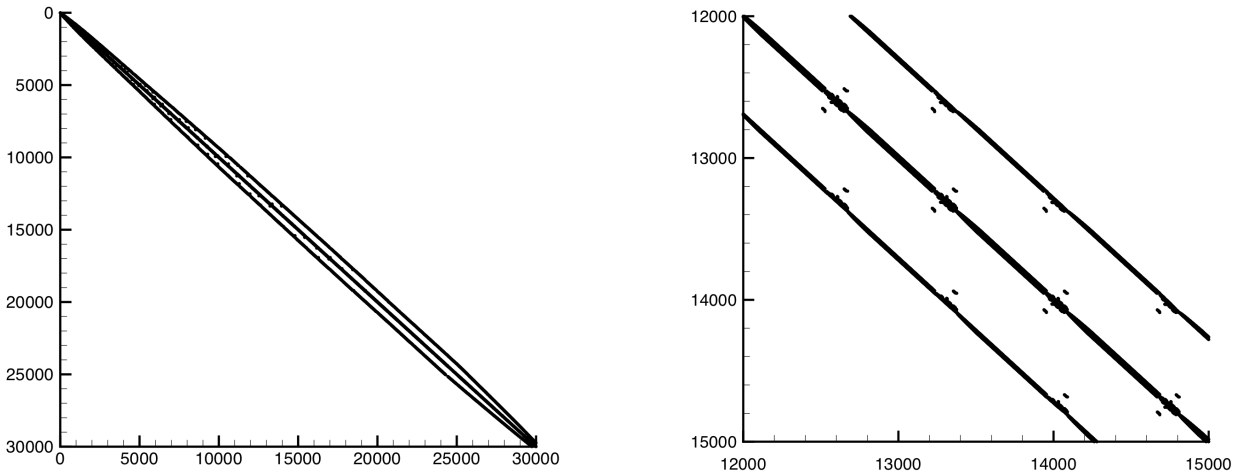
Fig. 7 Example mesh.

Table 4 Upper triangular matrix associated with Fig.7.

$$
\begin{array}{c}
\begin{bmatrix}
1 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 2 & 0 & x & 0 & 0 & x & 0 & 0 & 0 \\
 & & 3 & x & 0 & x & 0 & x & x & 0 \\
 & & & 4 & 0 & x & x & 0 & 0 & 0 \\
 & & & & 5 & x & x & 0 & 0 & x \\
 & & & & & 6 & x & x & 0 & x \\
 & & & & & & 7 & 0 & 0 & 0 \\
 & & & & & & & 8 & x & x \\
 & & & & & & & & 9 & 0 \\
 & & & & & & & & & 10
\end{bmatrix}
\end{array}
\begin{array}{l}
\rightarrow \quad 1\ (1.000) \\
\rightarrow \quad 2\ (1.000) \\
\rightarrow \quad 2\ (0.667) \\
\rightarrow \quad 4\ (1.000) \\
\rightarrow \quad 1\ (0.200) \\
\rightarrow \quad 6\ (1.000) \\
\rightarrow \quad 7\ (1.000) \\
\rightarrow \quad 7\ (0.875) \\
\rightarrow \quad 8\ (0.889) \\
\rightarrow \quad 8\ (0.800)
\end{array}
$$

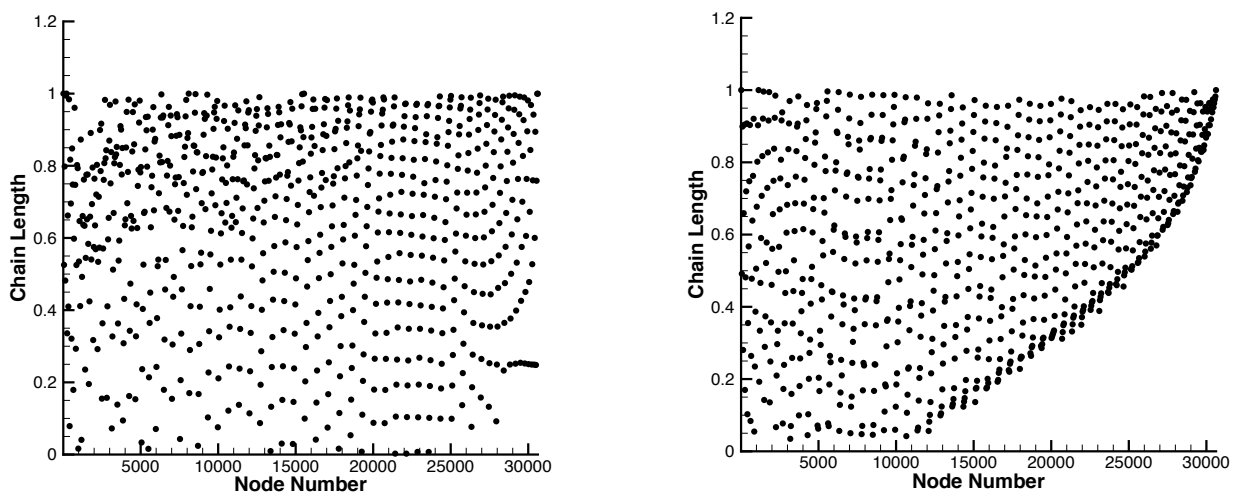
Many reordering strategies attempt to minimize the number of entries discarded during an incomplete LU decomposition by clustering entries in the matrix as close to the diagonal as possible. As seen in Fig. 8, which illustrates the matrix structure obtained using Cuthill-McKee for ordering the test problem introduced in section II, the numbering

strategy appears to be quite successful in that the bandwidth is narrow for all rows of the matrix. However, the resulting chain lengths, depicted in Fig. 9 (for every 50th row to reduce congestion on the plot), clearly indicate that while not all rows have maximal chain lengths, rows with maximal chain lengths are quite common, and are distributed across all rows of the matrix.



(a) Far view (b) Medium view

Fig. 8 Matrix structure for Cuthill-McKee.

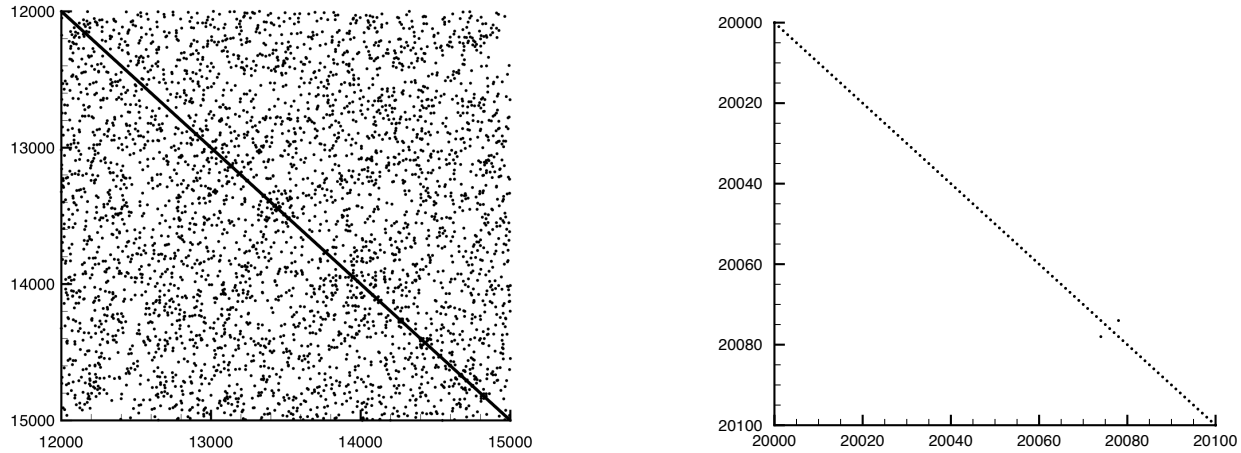


(a) Cuthill-McKee (b) Reverse Cuthill-McKee

Fig. 9 Chain lengths for Cuthill-McKee and reverse Cuthill-McKee orderings.

One reordering strategy that may facilitate the creation of well-oriented hyperplanes is to randomly order the unknowns, so that rows with common nonzero column entries will be scattered throughout the matrix. To demonstrate, the unknowns for the example problem have been randomly ordered, with the resulting distribution of entries within the matrix shown in Fig. 10. By closely examining Figs. 10(a) and 10(b), it is apparent that the nonzero entries are uniformly distributed throughout the matrix, and one can surmise that consecutive rows do not share many, if any, common entries. Because every entry on a row indicates a component of the hyperplane normal to the associated coordinate direction, the absence of common entries is a good indicator that the adjacent rows are likely well aligned.

Figure 11 displays the chain lengths obtained using random ordering. Here, the same scale as previously used for the Cuthill-McKee-based orderings in Fig. 9 is reused to illustrate the dramatic reduction obtained from random ordering.



(a) Medium view (b) Closeup

Fig. 10 Matrix structure for random ordering.

The figure shows that by using random ordering, even the longest chains are only a very small fraction of the maximum extent possible.

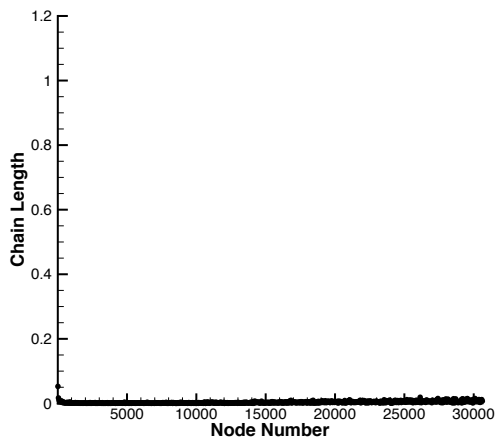
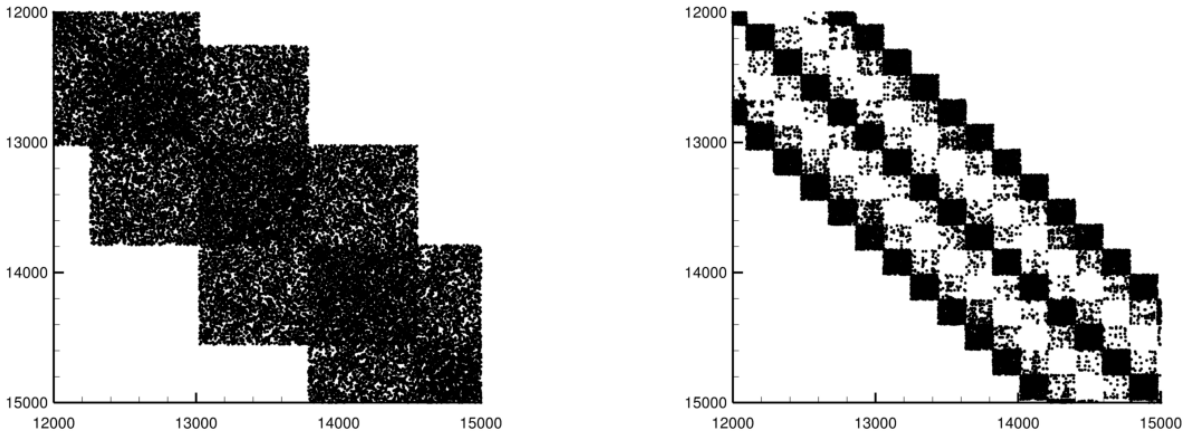


Fig. 11 Chain lengths for random ordering.

The figures above confirm that randomly numbering the unknowns is an effective means for breaking long chains, thereby improving the orientation of the hyperplanes. However, for cases where instability is not an issue, a narrow bandwidth generally leads to faster convergence with fewer search directions. A strategy to benefit from the advantages of both a narrow bandwidth, and random ordering, is to combine the two approaches. One means for achieving this, referred to as "q-ordering", is to first number the unknowns using a procedure that minimizes bandwidth, followed by a post-processing step where random ordering is applied across sets of sequential rows. In the current implementation, the number of rows considered at a time is specified as a percentage of the bandwidth after the bandwidth-minimization stage is completed. In the results that follow, the number of rows is determined in terms of $1/p$, where p is referred to here as the "prune width" and is used to control the balance between randomization (low p) and smaller bandwidths (large p). For example, $p = 4$ implies that the number of rows considered for simultaneous randomization is $1/4$ of the original bandwidth. Specific examples are presented in Fig. 12. As seen, a prune width of 1 results in a matrix with very well defined segments that appear to be nominally square. The resulting bandwidth is 1526, which is approximately twice the bandwidth of 765 obtained using the Cuthill-McKee orderings. As the prune width is increased, the bandwidth is decreased, so that with a prune width of 4, shown in Fig. 12(b), the bandwidth is 958, which is approximately 25% larger than for Cuthill-McKee.



(a) Prune width 1

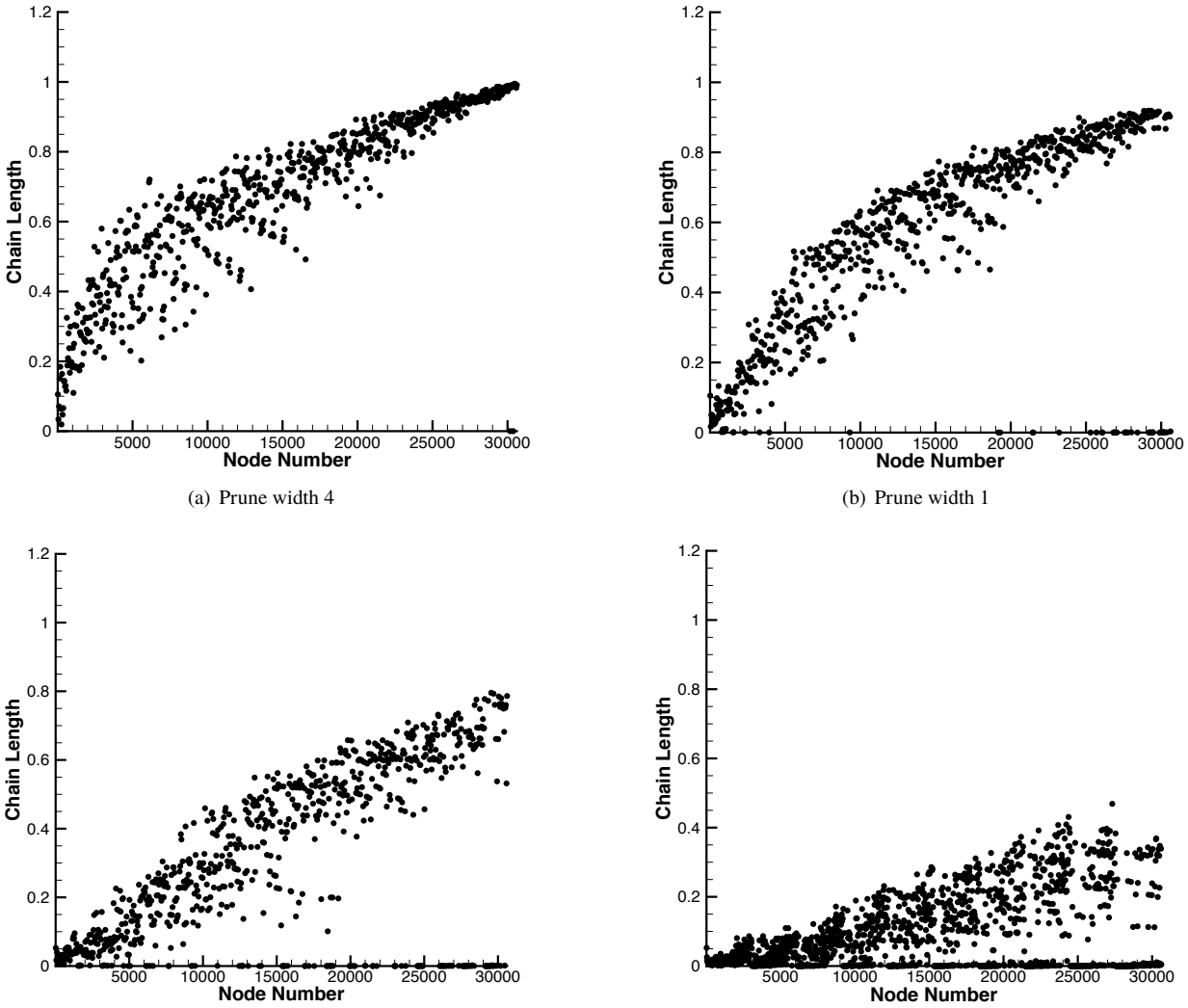
(b) Prune width 4

Fig. 12 Matrix structure for q-ordering.

Examining the chain lengths for q-ordering in Fig. 13, it is readily observable that the longest chains occur near the highest-numbered rows in the matrix and systematically decrease with decreasing row numbers. Importantly, as the prune width is decreased, the chain lengths are systematically reduced, indicating improved orientation of the hyperplanes.

As demonstrated above, introducing randomness into the ordering of unknowns has proven to be quite beneficial at reducing chain lengths, thereby also improving the orientation of hyperplanes. However, other strategies can be devised that are also effective. One methodology that has been effective in practice is referred to here as "k-ordering". Unlike q-ordering, where a bandwidth minimization ordering is first used, followed by a post-processing step of selective randomization of the unknowns, k-ordering is a single-step process similar to Cuthill-McKee. In this procedure, nodes are added in levels, where each level is determined by identifying all nodes that directly connect to nodes in the previous level. The first level is comprised of a single mesh point that has been chosen to be one of minimum degree. The nodes in each level are sorted using the procedure described in more detail in the appendix. The matrix structure obtained using this ordering strategy is substantially different than either Cuthill-McKee or q-ordering, as seen in Fig. 14. Here, a distinctive cross pattern is formed, along with a downward pointing diamond underneath. Although not shown, as more nodes are introduced because of increased fill, the diamond is typically systematically populated. While somewhat anecdotal, in practice, this technique typically provides more stability and reliable convergence than the Cuthill-McKee orderings. However, it does not offer the tunable parameter to control bandwidth that q-ordering provides, so that q-ordering is recommended for the most difficult cases.

One final ordering strategy presented is to order nodes by placing them into groups (colors) so that no node in a group is connected by an edge in the mesh. As with q-ordering, the application of this procedure is typically a two-step process where the mesh is first ordered using a methodology intended to reduce the bandwidth. As seen in Fig. 15, the entries in the matrix are spread out, but some structure is apparent from the bands of nonzero entries. The chain lengths for ILU(0) for the example test problem are shown in Fig. 15(b) and indicate that this may also be a viable method for ordering unknowns to form well-oriented hyperplanes.



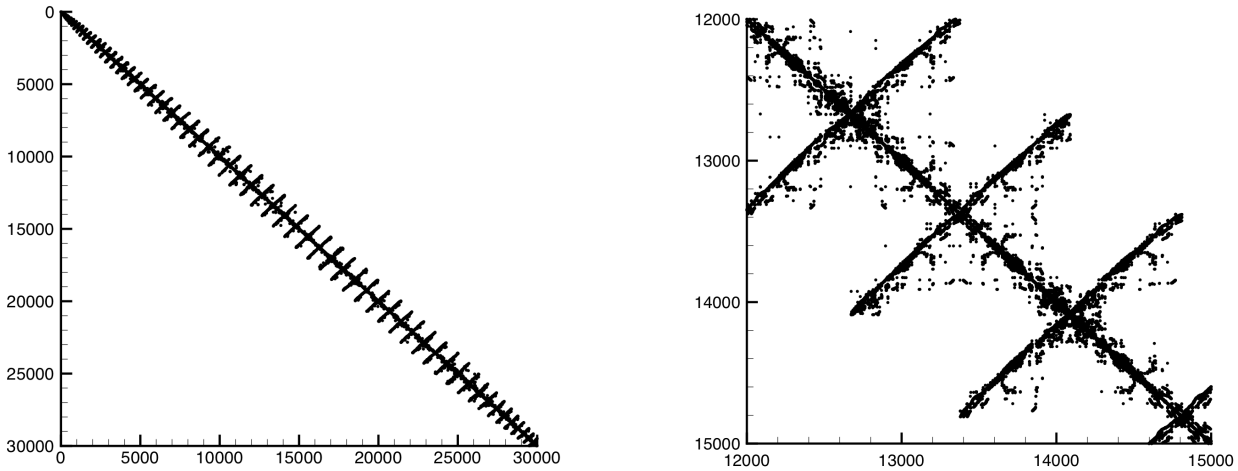
(a) Prune width 4

(b) Prune width 1

(c) Prune width 1/2

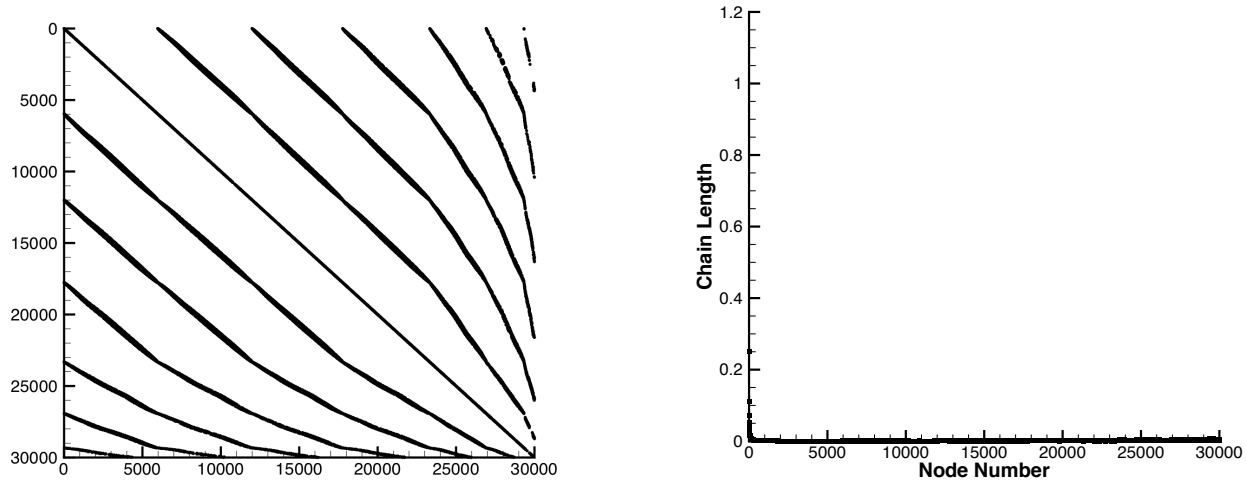
(d) Prune width 1/4

Fig. 13 Chain lengths for q-ordering.



(a) Far view (b) Medium view

Fig. 14 Matrix structure for k-ordering.

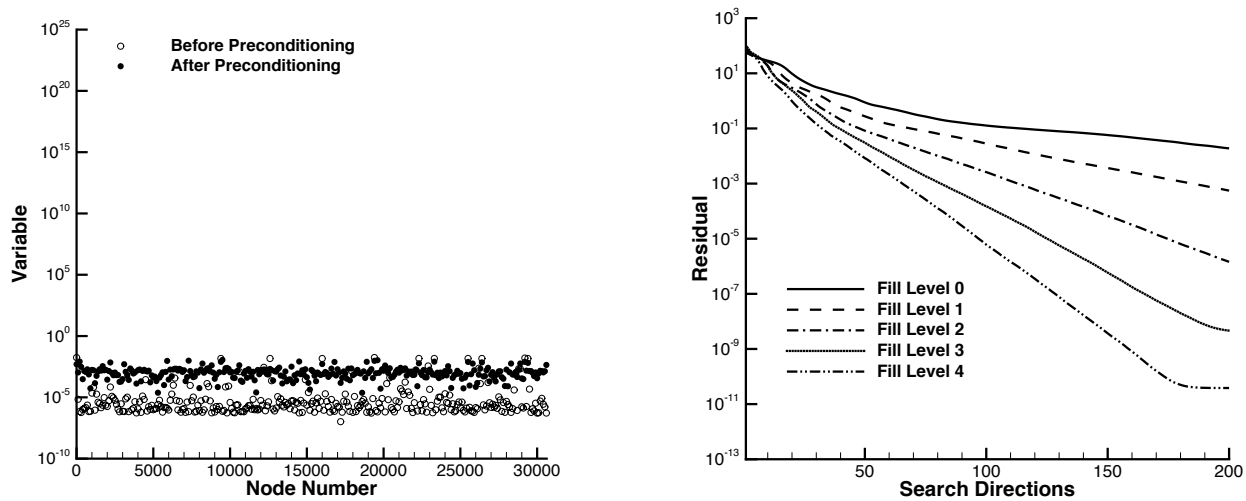


(a) Matrix structure (b) Chain lengths

Fig. 15 Matrix structure and chain lengths for color-based ordering.

V. Results

To examine the effectiveness of the proposed ordering strategies, several test cases from fluid dynamics and structural analysis problems are considered. The first case is a continuation of the example problem described earlier in section II. To demonstrate that random ordering does, in fact, stabilize the preconditioner for this problem, Fig. 16(a) shows the components of the first preconditioned Krylov vector before and after applying the preconditioner, similar to that shown in Fig. 3 for Cuthill-McKee. Figure 16(a) demonstrates that by using random ordering, the preconditioner is now stable. With the preconditioner now stabilized, Fig. 16(b), demonstrates that GMRES is able to reduce the residual, and that consistent gains in convergence are observed as the fill level is increased. While random ordering is not normally regarded as a good preconditioner, or even one that should be considered, the results below confirm that this method is quite effective at creating well-oriented hyperplanes that consequently stabilize the preconditioner.



(a) Stability of preconditioning

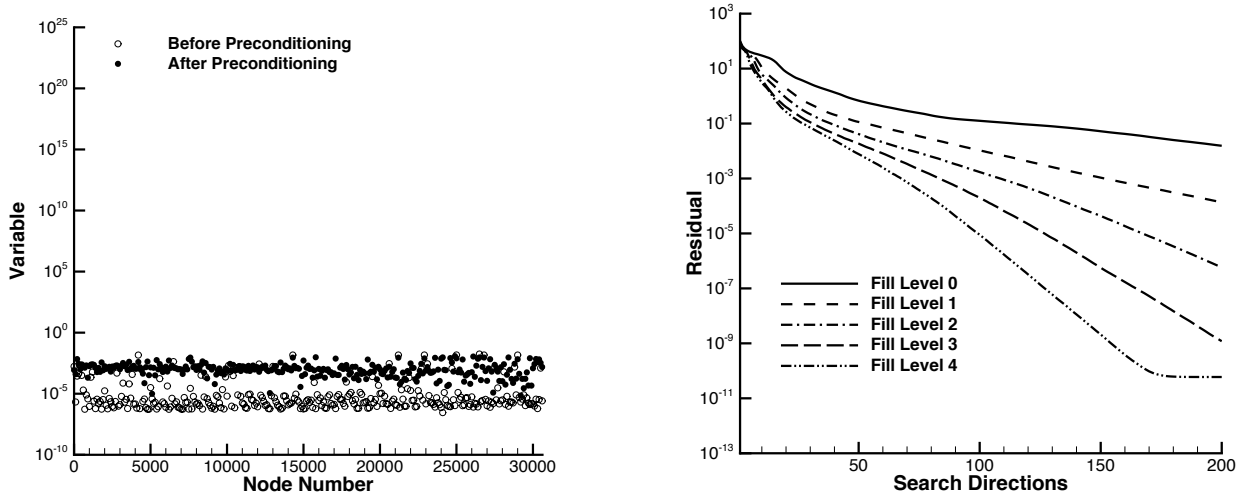
(b) Convergence

Fig. 16 Stability of preconditioner and GMRES convergence with random ordering.

The benefits of q-ordering are similarly readily apparent, as demonstrated in Fig. 17, which shows the variables before and after applying the preconditioner, as well as convergence history for a prune widths of 4. As with random ordering, the preconditioner is stable, and GMRES successfully reduces the residual. Although not shown, similar results are obtained for a prune width of 1.0, as well as for k-ordering and ordering based on node coloring.

During the description of this initial test case (section II), the instability of the preconditioner had been attributed to the LU decomposition, as evidenced by vary large differences between factorizations computed using 64-bit and 128-bit arithmetic. Similar studies have been done using the alternative ordering strategies described above, and are summarized below. Table 5 compares the RMS and maximum relative differences in the lower- and upper- factorization components computed using 64-bit and 128-bit accuracy for several ordering strategies. Large relative errors result from both Cuthill-McKee and reverse Cuthill-McKee orderings. In stark contrast, random ordering, k-ordering, and q-orderings with prune widths of 1, 4, and 12 all demonstrate only small relative differences, indicating that the LU decomposition is stable. With stable LU decomposition, GMRES successfully converges the linear problem.

The second test case is also from a fluid-dynamic application, and is used to demonstrate a few salient features of the ordering principles being proposed. This case, illustrated in Fig. 18 is for the high-lift common research model (CRM) [43] at a Mach number of 0.2, an angle of attack of 16°, and an Reynolds number of 3.26×10^6 based on the mean aerodynamic chord. The mesh is comprised of $5,714,675$ nodes and $33,699,514$ tetrahedrons. Residual convergence is shown in Fig. 19(a) with the corresponding history of the CFL number shown in Fig. 19(b). For all these runs, ILU(2) is used as the preconditioner and results using several ordering strategies are shown. As seen in Fig. 19(a), convergence of the residual is achieved in about 220 iterations using reverse Cuthill-McKee ordering, but only about 150 iterations are required with q-ordering and a prune width of $1/2$. In the figure, the solid circles indicate iterations where the magnitude of the variables after preconditioning exceeded those entering the preconditioner by more than 10 orders of magnitude. While somewhat arbitrary, exceeding this threshold correlates well with failure of the linear system to converge. Figure 19(a) indicates that the preconditioner failed approximately 11 times when using Cuthill-McKee



(a) Stability of preconditioning

(b) Convergence

Fig. 17 Stability of preconditioner and GMRES convergence with q-ordering (prune width 4).

Table 5 LU decomposition errors for bump problem with ILU(0).

Ordering	RMS Relative Difference	Max Relative Difference	Convergence
Cuthill-McKee	$0.7816 \times 10^{+03}$	$0.2286 \times 10^{+07}$	X
Reverse Cuthill-McKee	$0.1675 \times 10^{+04}$	$0.3495 \times 10^{+07}$	X
Random	0.2063×10^{-05}	0.4487×10^{-02}	✓
q(1)	0.3484×10^{-05}	0.1149×10^{-01}	✓
q(4)	0.1108×10^{-05}	0.2289×10^{-02}	✓
q(12)	0.1604×10^{-05}	0.3269×10^{-02}	✓
k-ordering	0.3417×10^{-05}	0.1151×10^{-01}	✓
Coloring	0.5651×10^{-05}	0.2158×10^{-01}	✓
Reverse Coloring	0.1918×10^{-05}	0.3904×10^{-02}	✓

ordering, primarily when the residual was nearing convergence and the CFL number was above 1000. This trend is not universal as failures are often encountered earlier in the iterative process, especially on poor-quality meshes. Observe from Fig. 19(b) that once the preconditioner begins encountering instability, the CFL number fails to increase. In contrast, results obtained using q-ordering with prune widths of 4 and $1/2$ both converge very rapidly beginning at iterations 150 and 170, respectively, and the preconditioner is never flagged as being unstable. At the same time that the residual is rapidly decreased, the CFL numbers rapidly increase, reaching almost 1×10^6 before convergence is reached. Although not shown, the trends in computer time closely mimic the convergence history, so that the total run time using Cuthill-McKee ordering is ultimately about 50% larger that using q-ordering with either prune width.

The solutions above have been obtained using ILU(2) as the preconditioner. Tests conducted on one of the partitions where the preconditioner failed using reverse Cuthill-McKee indicated that using ILU(0) was a stable alternative. After verifying that the LU decomposition with ILU(0) was stable by comparing factorizations obtained using 64-bit and 128-bit arithmetic, an experiment referred to here as the "null-space" test, is used to demonstrate the propagation of errors during backsubstitution. Specifically, once the LU decomposition is computed for each ordering, backsubstitution is then performed after first setting all entries on the right-hand side to zero except for on a single row. For these experiments, the right-hand side of row $30,000$ is modified so that the exact solution for the variables at that node are order ϵ (later normalized to 1.0). Completing the backsubstitution process and observing the solution at all subsequent nodes allows the imposed perturbation to be tracked through the system. Because this system is previously verified to be

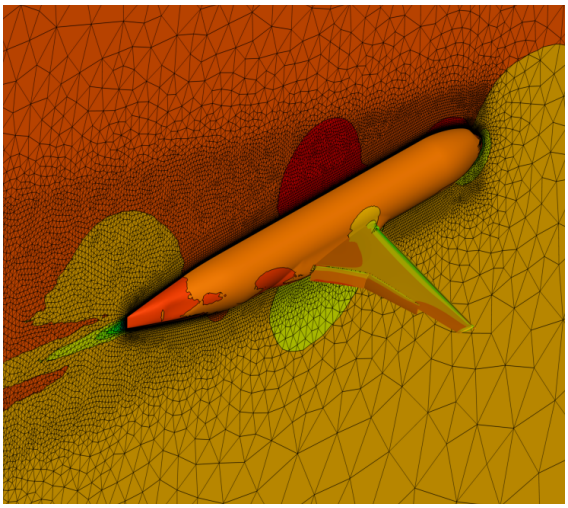


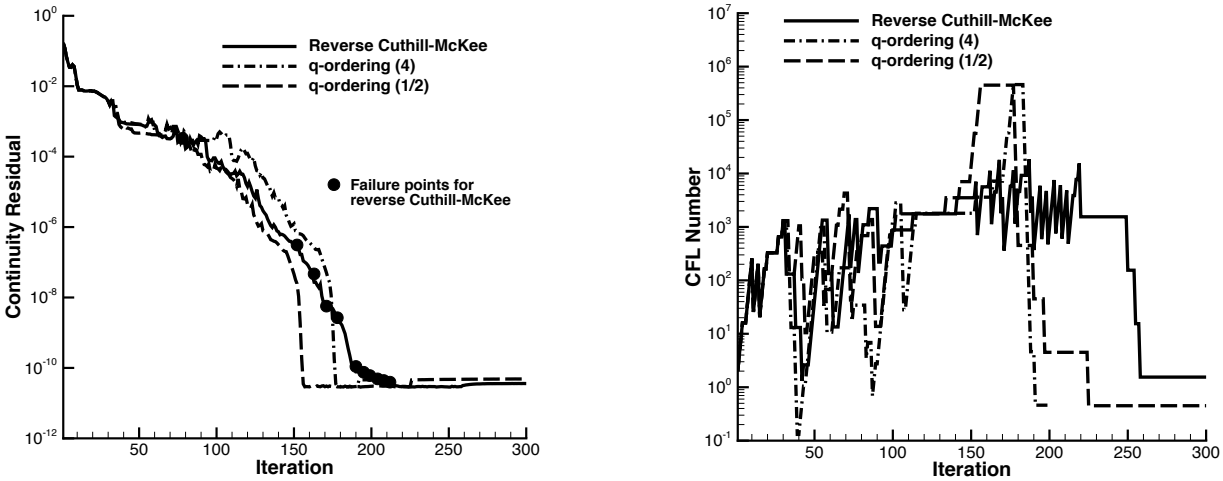
Fig. 18 Solution contours for flow over high-lift common research model: $M_\infty = 0.2$**,** $Re = 3.26 \times 10^6$**,** $\alpha = 16°$**.**

stable, the errors should not be amplified. The value in the experiment is in observing how far the error propagates.

Recall from Fig. 4 that with well-oriented hyperplanes, one expects errors introduced at any point during backsubstitution to diminish as the solution on lower-numbered rows is obtained. Also recall from Fig. 9 that chain lengths using Cuthill-McKee tend to be longer than those for q-ordering (Fig. 13), which are themselves longer than for random ordering (Fig.11). As such, one would expect that an error introduced at any specific location during backsubstitution would propagate further using Cuthill-McKee than for q-ordering, and that errors for q-ordering would propagate farther than with random ordering.

Figure 20 verifies that the expected results are obtained. As seen in Fig. 20(a), which indicates nonzero solutions obtained using Cuthill-McKee as the ordering, the perturbations introduced at row $30,000$ do indeed propagate all the way to the first row. In contrast, when using q-ordering and a prune width of $1/2$, the perturbations are clearly damped as the backsubstitution advances. With random ordering, the damping effect is even greater, with only a few nodes having nonzero values at all. Note that in the figures, results for Cuthill-McKee are plotted at only one of every 50 rows to reduce clutter. In contrast, with random ordering, all the values with magnitudes greater than 1×10^{-6} are shown, which clearly demonstrates that perturbations are effectively damped using random ordering. This experiment confirms that the hyperplanes are very well oriented when random ordering is used, but less well oriented as the chain lengths increase.

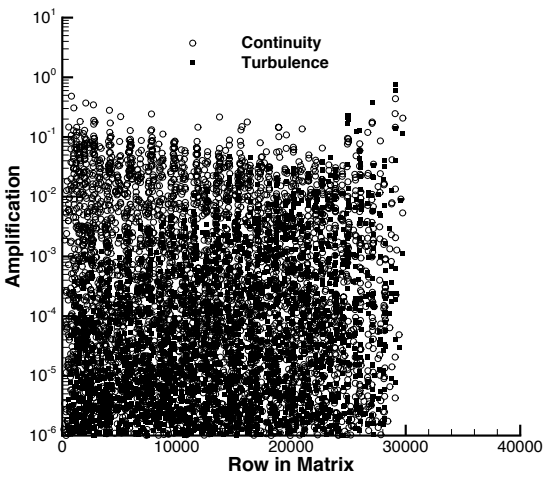
The results thus far have been for fluid-dynamics problems, where the solution of the linear system is one component of an outer Newton-based algorithm for solving the nonlinear equations. Within the context of the nonlinear problem, failure of the linear solver is not necessarily catastrophic. For example, in Fig. 19 for the high-lift common research model, although solving the linear system failed at numerous locations, the nonlinear residual ultimately converges because failure to solve the linear system can be compensated for by simply rejecting the nonlinear step and lowering the CFL number. In fact, in Ref. [35], failures in the linear system were not even monitored, and the determination of whether to accept or reject the step was a decision solely left up to the nonlinear controller. Since that time however, that laissez-faire approach has been replaced so that a nonlinear step is rejected if the linear solver fails. Circumstantial evidence indicates that, at least for the solver currently being used, accepting failed linear solves can lead to future difficulties that are sometimes problematic to overcome. Whether or not to accept or reject these steps is still a matter of research, and likely depends on the details of the solver.



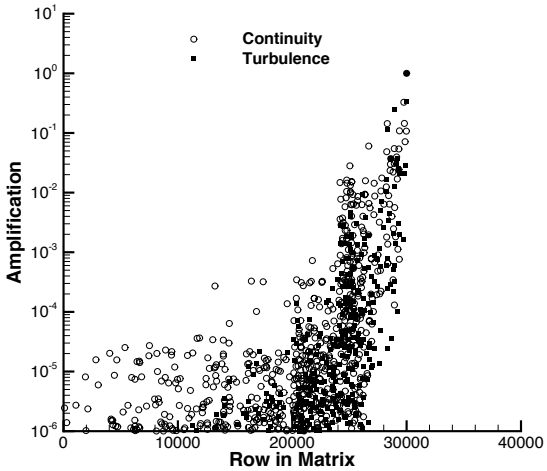
(a) Residual convergence

(b) CFL history

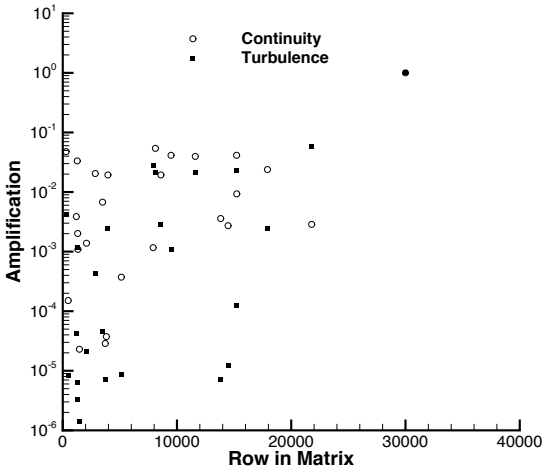
Fig. 19 Iterative convergence and CFL history for high-lift common research model.



(a) Reverse Cuthill-McKee



(b) q-ordering (1/2)

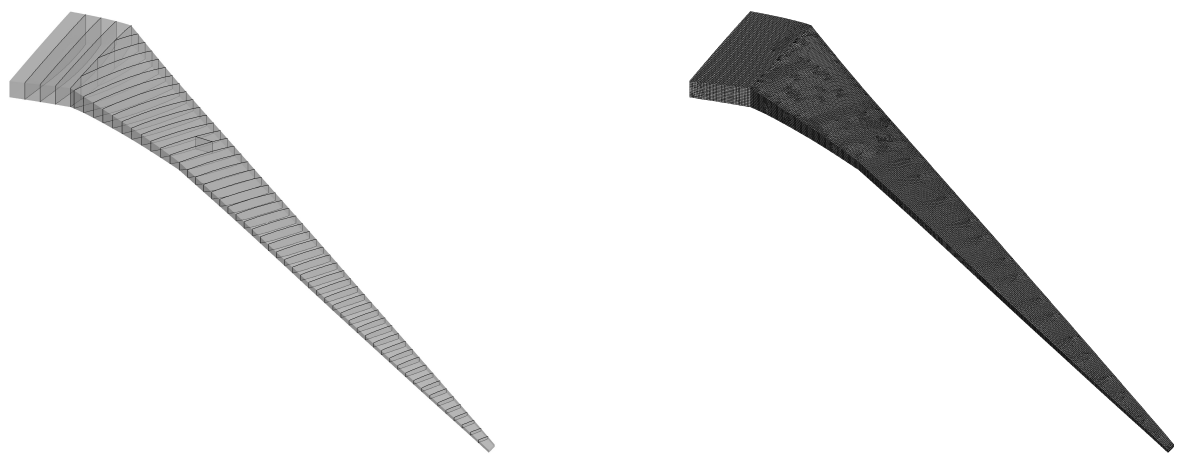


(c) Random ordering

Fig. 20 Null-space test.

For many applications, such as structural analysis and design, solving the linear system is the entire focus of the research, and options to modify the matrix by adjusting a CFL-type parameter as discussed above are not available options. These matrices are often symmetric and positive definite, but may still be very ill conditioned and can be difficult, if not impossible, to solve using iterative methods. For these problems, direct solvers with full or partial pivoting are normally the preferred approach.

To evaluate the capabilities of the current ordering schemes to solve a difficult structural problem, a stiffness matrix generated with TACS [44, 45] for the 13.5 aspect ratio version of the undeformed Common Research Model (uCRM) [46] is considered. The model and corresponding mesh, shown in Fig. 21, is composed of 23, 399 shell elements (22, 108 nodes) based on the mixed-interpolation of tensorial components (MITC) formulation [47] to model the ribs, spars, and skins of the wingbox of the uCRM. At each mesh point, there are 6 variables, yielding a total of 132, 648 unknowns in the linear system, which has only 3910 diagonally dominant rows. As with the fluid-dynamic applications, the matrix is solved by taking advantage of the block structure so that when evaluated using the Frobenius norm, only 564 rows, out of 22, 108, are diagonally dominant.



(a) Ribs, spars, and skin for undeformed common research model

(b) Mesh for undeformed common research model

Fig. 21 Structural model for common research model.

Table 6 presents a summary of results obtained using GMRES, 100 search directions, and ILU(0) as a preconditioner. For each ordering, the RMS and maximum relative difference for all the lower and upper factors in the LU decomposition computed using 64-bit and 128-bit arithmetic are shown. As seen, large relative differences are present for the Cuthill-McKee orderings and, as indicated in the last column, the residual after 100 search directions of GMRES had increased by almost a factor of 3. In contrast, with random ordering, color-based ordering, and q-ordering with prune widths below 4, only small RMS errors are present in the LU factors and, after 100 search directions, the residual is successfully reduced. For k-ordering and for q-ordering with prune width of 4, noticeably larger errors are present in the LU factors. Although not shown, the ratio of values before and after applying the preconditioner for q-ordering with a prune width of 4 are on the order of 1.0×10^8. On the other hand, for k-ordering, the values before and after applying the preconditioner are roughly the same magnitude, and show no sign of instability. Although k-ordering converges for this case, and q-ordering with a prune width of 4 does not, the levels of error present with k-ordering may portend incipient instability as will be demonstrated below.

Because increasing the fill level adds numerous entries into the matrix, the resulting decreased sparsity can generally be expected to increase the chain lengths, thereby adversely affecting the orientation of the hyperplanes. This supposition is verified for this case as exhibited in Table 7, which shows results similar to those in Table 6, except obtained with ILU(1) as the preconditioner. In the table, random ordering and q-ordering with low prune widths are all able to successfully reduce the residual, whereas all the other orderings lead to failure in solving of the linear system.

Observe that the RMS and maximum relative errors for q-orderings with prune widths of 1 and 4 are very similar,

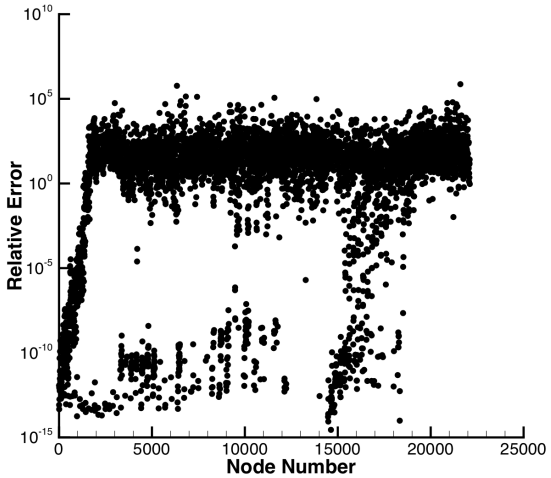
Table 6 LU decomposition errors for structural problem with ILU(0).

Ordering	RMS Relative Difference	Max Relative Difference	Convergence	Reduction
Cuthill-McKee	$0.2788 \times 10^{+02}$	$0.5499 \times 10^{+05}$	X	$0.2880 \times 10^{+01}$
Reverse Cuthill-McKee	$0.2312 \times 10^{+04}$	$0.3995 \times 10^{+07}$	X	$0.6188 \times 10^{+01}$
Random	0.1826×10^{-03}	$0.4259 \times 10^{+00}$	✓	$0.3204 \times 10^{+00}$
q(1/4)	0.5337×10^{-03}	$0.1408 \times 10^{+01}$	✓	$0.3337 \times 10^{+00}$
q(1/2)	0.1622×10^{-02}	$0.2005 \times 10^{+01}$	✓	$0.5406 \times 10^{+00}$
q(1)	0.5576×10^{-02}	$0.1158 \times 10^{+02}$	✓	$0.3353 \times 10^{+00}$
q(4)	$0.1106 \times 10^{+04}$	$0.2892 \times 10^{+07}$	X	$0.3418 \times 10^{+03}$
k-ordering	$0.2183 \times 10^{+02}$	$0.2804 \times 10^{+05}$	✓	$0.6276 \times 10^{+00}$
Node Coloring	0.1313×10^{-04}	0.1875×10^{-01}	✓	$0.2981 \times 10^{+00}$
Reverse Node Coloring	0.4407×10^{-04}	$0.1171 \times 10^{+00}$	✓	$0.2940 \times 10^{+01}$

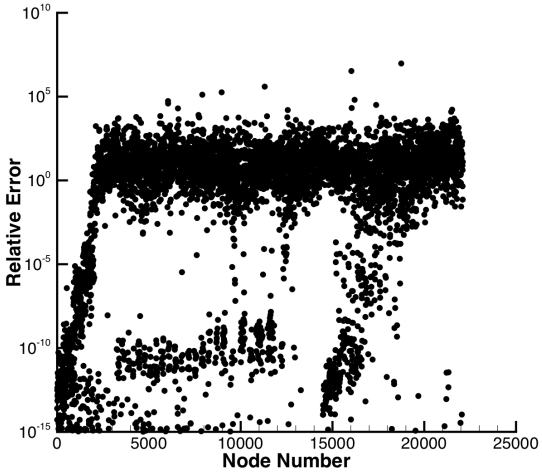
and yet using a prune width of 1 converges whereas using a prune width of 4 does not. The explanation for this behavior can be observed through plotting the largest relative errors in the factorization along each row. For example, Fig. 22 demonstrates the errors for the upper portion of the matrix after factorization, with the lower portion (not shown) appearing very similar. As seen, with reverse Cuthill-Mckee the factorization rapidly goes unstable so that by the 500th row, the relative errors computed using 64-bit and 128-bit arithmetic are consistently above $10,000$. However, as more randomness is introduced, the errors are progressively lowered, so that with a prune width of $1/4$ and random ordering, the errors are all nominally about 1×10^{-10}. The explanation for achieving convergence using a prune width of 1, whereas failing for a prune width of 4, is evident in Figs. 22(b) and 22(c) where one can clearly see that with a prune width of 4, the LU decomposition immediately goes unstable. In contrast, with a prune width of 1, the computation still appears to be marginally unstable, but growing much less rapidly than with a prune width of 4. An additional observation for this case is that the orderings based on node-coloring algorithms both failed using ILU(1). The reason is that as the fill level is increased, the new entries are added along the bands that are apparent in Fig. 15, thereby coupling the nearby rows at the detriment of the quality of the hyperplane orientation. Finally, although not shown, experiments with random ordering using fill levels ranging from ILU(0) to ILU(5) indicated that improved convergence was obtained as the fill level was increased. Specifically with ILU(0), the residual was reduced by a factor of 3 after 100 search directions, whereas using ILU(5), the residual was reduced by more than two orders of magnitude.

Table 7 LU decomposition errors for structural problem with ILU(1).

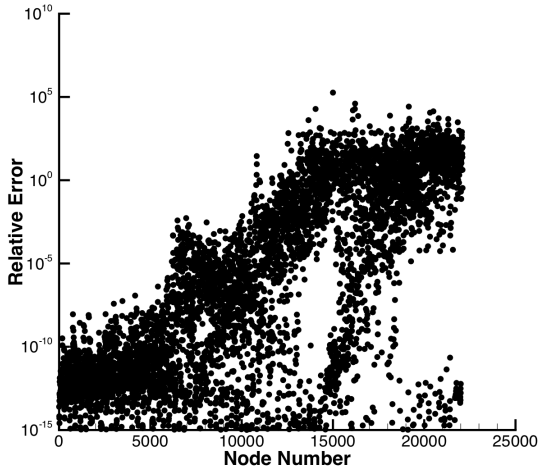
Ordering	RMS Relative Difference	Max Relative Difference	Convergence	Reduction
Cuthill-McKee	$0.1342 \times 10^{+04}$	$0.1862 \times 10^{+07}$	X	$0.8204 \times 10^{+01}$
Reverse Cuthill-McKee	$0.1283 \times 10^{+04}$	$0.2359 \times 10^{+07}$	X	$0.4237 \times 10^{+01}$
Random	0.4128×10^{-03}	$0.9542 \times 10^{+00}$	✓	$0.2880 \times 10^{+00}$
q(1/4)	0.2039×10^{-02}	$0.5459 \times 10^{+00}$	✓	$0.3185 \times 10^{+00}$
q(1/2)	$0.1691 \times 10^{+01}$	$0.4596 \times 10^{+04}$	✓	$0.3197 \times 10^{+00}$
q(1)	$0.1034 \times 10^{+04}$	$0.3590 \times 10^{+07}$	✓	$0.2804 \times 10^{+00}$
q(4)	$0.1032 \times 10^{+04}$	$0.1686 \times 10^{+07}$	X	$0.1162 \times 10^{+02}$
k-ordering	$0.6032 \times 10^{+05}$	$0.2042 \times 10^{+09}$	X	$0.5921 \times 10^{+02}$
Node Coloring	$0.6495 \times 10^{+02}$	$0.1208 \times 10^{+06}$	X	$0.3493 \times 10^{+03}$
Reverse Node Coloring	$0.1052 \times 10^{+04}$	$0.3439 \times 10^{+07}$	X	$0.1164 \times 10^{+04}$



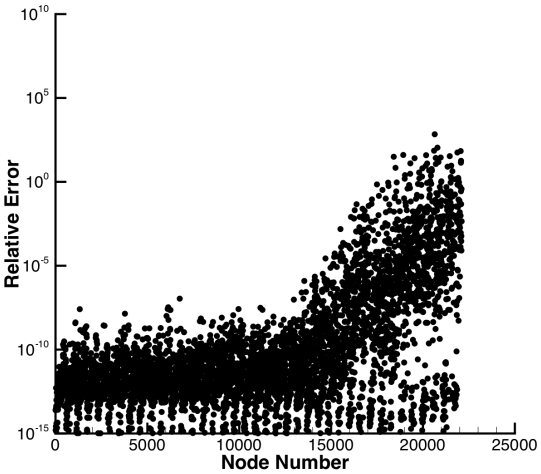
(a) Reverse Cuthill-McKee



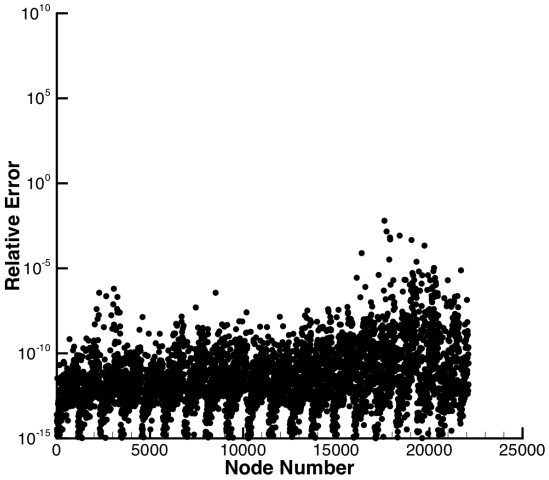
(b) q-ordering prune width 4



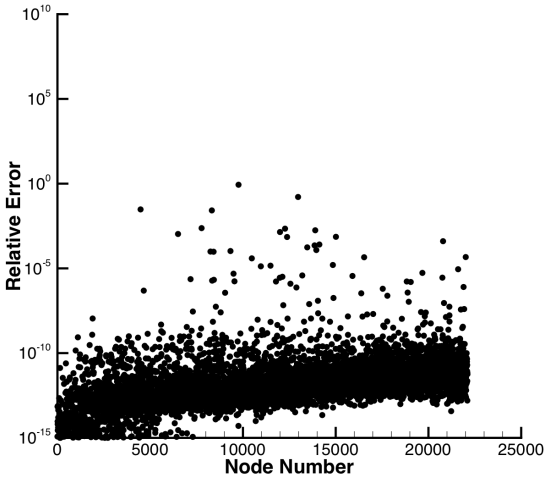
(c) q-ordering prune width 1



(d) q-ordering prune width 1/2



(e) q-ordering prune width 1/4



(f) Random ordering

Fig. 22 Relative errors in upper factorization.

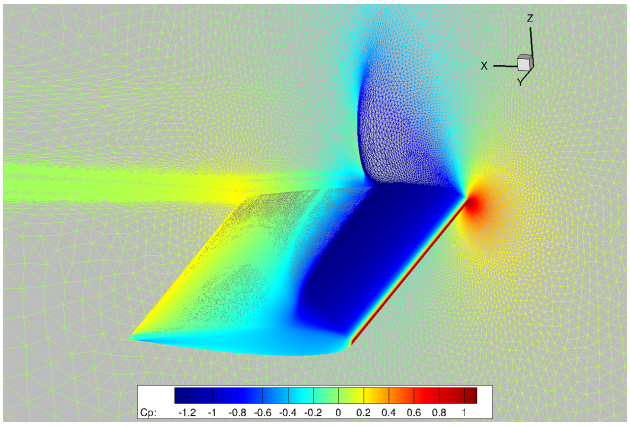


Fig. 23 Pressure contours for the Benchmark 0012 Wing at Mach 0.78 and an angle of attack of $4.0°$**.**

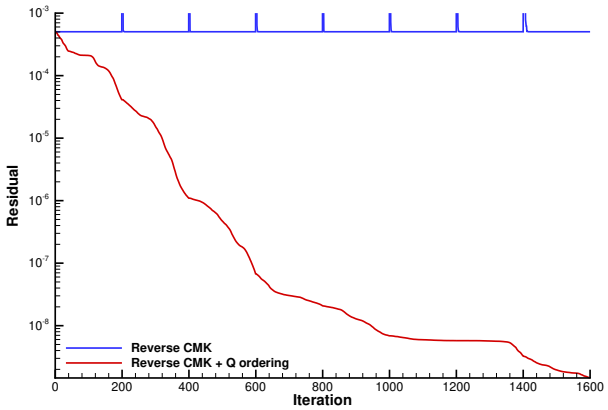


Fig. 24 Linear solver convergence for the LFD analysis of Benchmark 0012 Wing at Mach 0.78 and an angle of attack of $4.0°$**.**

A final demonstration of the ordering procedures described above is conducted for the Benchmark 0012 aeroelastic wing tested in the Transonic Dynamics Tunnel to study transonic flutter [48]. Stanford et al.[49], recently performed detailed computational analysis of this wing using FUN3D [50]. As part of that work, flutter analyses were conducted using a linearized frequency domain (LFD) solver, described in Ref. [51], which solves a linear system to compute the linearized response of the flow field to a periodic perturbation at a given frequency. In the frequency domain, the complex-valued system is solved with SLAT [37] using 200 Krylov dimensions and 1600 total search directions. Q-ordering is used to solve LFD problems about the transonic flow field in Figure 23, computed by the finite-element flow solver on an adapted mesh [52, 53]. When using reverse Cuthill-Mckee, 85 of the 400 partitions of the transonic LFD problem are flagged as having an unstable preconditioner based on the variables after preconditioning exceeding those entering the preconditioner by more than 10 orders of magnitude. This leads to essentially no reduction in the estimated residual as seen in Figure 24. However, when q-ordering with a prune width of 0.1 is applied to the same linear problem, none of the partitions are flagged as having an unstable preconditioner, and the linear system converges more than 5 orders of magnitude as seen in Figure 24.

VI. Summary

The importance of ordering unknowns for stabilizing preconditioners based on incomplete LU decomposition has been demonstrated. Several ordering strategies have also been presented that naturally facilitate forming well-oriented hyperplanes between nearby rows of the matrix, which is a key element for promoting stability for both factorization and backsubstitution. Although contrary to well-accepted paradigms for ordering unknowns, a simple random-number generator is demonstrated to be a very effective means for realizing this objective. To control the bandwidth, unknowns

can first be ordered using standard procedures such as Cuthill-McKee, followed by sequentially "randomizing" groups of adjacent rows. Through numerous examples, the principles underlying the development of this technique are demonstrated, and the effectiveness of the proposed procedures is verified using fluid-dynamics and structural analysis applications.

VII. Acknowledgments

This research was sponsored by the NASA Transformational Tools and Technologies (TTT) Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate. The authors would like to thank Ryan Glasby and Taylor Erwin for providing feedback on k-ordering, Bret Stanford for providing the NACA 0012 wing case, and also Matthew O'Connell for providing the PETSc mini-app.

VIII. Appendix: k-ordering

The strategy described below evolved from experiments examining how newly-developed ordering strategies affect the convergence and robustness of Krylov-subspace methods. One such ordering strategy was obtained by first identifying a starting node, and subsequently numbering the remaining nodes in the mesh by levels, where a level is comprised of all the of nodes immediately connected to the previous level. After each level is initially filled, the nodes that the sorting algorithm provided in the pseudocode below is used to rearrange nodes within the level so that nodes with low connectivity tend to migrate toward the top of the list without actually sorting the nodes. While sorting nodes according to degree is certainly an option, numerical experiments using a test matrices indicated that fully sorting is not as effective for stabilizing the preconditioner as is the algorithm below. While this method is certainly ad hoc, it has proven to be effective for many fluid-dynamic applications. For very difficult problems, such as solving the adjoint equations or for the linearized frequency domain problems, q-ordering or random ordering are generally more effective.

Algorithm 1 Pseudocode for sorting nodes on a level

```
n = number of nodes in mesh
list = list to be sorted
deg = degree for every node
start = starting index for this level
end = ending index for this level

function sortlist(n,list,deg,start,end)
array list(n)
array deg(n)
for i = start to end do
   n1 = list(i)
   for j = i + 1 to end do
      n2 = list(j)
      if deg(n2) ≤ deg(n1) then
         swap = list(i)
         list(i) = list(j)
         list(j) = swap
      end if
   end for
end for
end function
```

References

[1] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[2] Axelsson, O., "Conjugate Gradient Type Methods for Unsymmetric and Inconsistent Systems of Linear Equations," *Linear Algebra and its Applications*, Vol. 29, 1980, pp. 1 – 16. doi:https://doi.org/10.1016/0024-3795(80)90226-8, URL http://www.sciencedirect.com/science/article/pii/0024379580902268, special Volume Dedicated to Alson S. Householder.

[3] Eisenstat, S. C., Elman, H. C., and Schultz, M. H., "Variational Iterative Methods for Nonsymmetric Systems of Linear Equations," *SIAM Journal on Numerical Analysis*, Vol. 20, No. 2, 1983, pp. 345–357. URL http://www.jstor.org/stable/2157222.

[4] Saad, Y., "Practical Use of Some Krylov Subspace Methods for Solving Indefinite and Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 5, No. 1, 1984, pp. 203–228. doi:10.1137/0905015, URL https://doi.org/10.1137/0905015.

[5] Elman, H. C., "Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations," Ph.D. thesis, Yale University, New Haven, CT, USA, 1982. AAI8222744.

[6] van der Vorst, H. A., "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, No. 2, 1992, pp. 631–644. doi:10.1137/0913035, URL https://doi.org/10.1137/0913035.

[7] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[8] Poole, G., and Neal, L., "Gaussian Elimination: When is Scaling Beneficial?" *Linear Algebra and its Applications*, Vol. 162-164, 1992, pp. 309 – 324. doi:https://doi.org/10.1016/0024-3795(92)90382-K, URL http://www.sciencedirect.com/science/article/pii/002437959290382K.

[9] Poole, G., and Neal, L., "A Geometric Analysis of Gaussian Elimination. I," *Linear Algebra and its Applications*, Vol. 149, 1991, pp. 249 – 272. doi:https://doi.org/10.1016/0024-3795(91)90337-V, URL http://www.sciencedirect.com/science/article/pii/002437959190337V.

[10] Neal, L., and Poole, G., "A Geometric Analysis of Gaussian Elimination. II," *Linear Algebra and its Applications*, Vol. 173, 1992, pp. 239 – 264. doi:https://doi.org/10.1016/0024-3795(92)90432-A, URL http://www.sciencedirect.com/science/article/pii/002437959290432A.

[11] Chow, E., and Saad, Y., "Experimental Study of ILU Preconditioners for Indefinite Matrices," *Journal of Computational and Applied Mathematics*, Vol. 86, No. 2, 1997, pp. 387–414. doi:10.1016/S0377-0427(97)00171-4.

[12] Smyth, W., "Algorithms for the Reduction of Matrix Bandwidth and Profile," *Journal of Computational and Applied Mathematics*, Vol. 12-13, 1985, pp. 551 – 561. doi:https://doi.org/10.1016/0377-0427(85)90048-2, URL http://www.sciencedirect.com/science/article/pii/0377042785900482.

[13] Bernardes, J. A. B., and de Oliveira, S. L. G., "A Systematic Review of Heuristics for Profile Reduction of Symmetric Matrices," *Procedia Computer Science*, Vol. 51, 2015, pp. 221 – 230. doi:https://doi.org/10.1016/j.procs.2015.05.231, URL http://www.sciencedirect.com/science/article/pii/S187705091501039X, international Conference On Computational Science, ICCS 2015.

[14] Chagas, G. O., and de Oliveira, S. L. G., "Metaheuristic-based Heuristics for Symmetric-matrix Bandwidth Reduction: A Systematic Review," *Procedia Computer Science*, Vol. 51, 2015, pp. 211 – 220. doi:https://doi.org/10.1016/j.procs.2015.05.229, URL http://www.sciencedirect.com/science/article/pii/S1877050915010376, international Conference On Computational Science, ICCS 2015.

[15] Sloan, S. W., and Randolph, M. F., "Automatic Element Reordering for Finite Element Analysis with Frontal Solution Schemes," *International Journal for Numerical Methods in Engineering*, Vol. 19, No. 8, 1983, pp. 1153–1181. doi:10.1002/nme.1620190805, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620190805.

[16] D'Azevedo, E. F., Forsyth, P. A., and Tang, W.-P., "Ordering Methods for Preconditioned Conjugate Gradient Methods Applied to Unstructured Grid Problems," *SIAM Journal on Matrix Analysis and Applications*, Vol. 13, No. 3, 1992, pp. 944–961. doi:10.1137/0613057, URL https://doi.org/10.1137/0613057.

[17] Feng, G., "An Improvement of the Gibbs-Poole-Stockmeyer Algorithm," *Journal of Algorithms & Computational Technology*, Vol. 4, No. 3, 2010, pp. 325–333. doi:10.1260/1748-3018.4.3.325, URL https://doi.org/10.1260/1748-3018.4.3.325.

[18] Liu, W.-H., and Sherman, A. H., "Comparative Analysis of the Cuthill-McKee and the Reverse Cuthill-McKee Ordering Algorithms for Sparse Matrices," *SIAM Journal on Numerical Analysis*, Vol. 13, No. 2, 1976, pp. 198–213. URL http://www.jstor.org/stable/2156087.

[19] Cuthill, E., and McKee, J., "Reducing the Bandwidth of Sparse Symmetric Matrices," *Proceedings of the 1969 24th National Conference*, ACM, New York, NY, USA, 1969, pp. 157–172. doi:10.1145/800195.805928, URL http://doi.acm.org/10.1145/800195.805928.

[20] Dutto, L. C., "The Effect of Ordering on Preconditioned GMRES Algorithm, for Solving the Compressible Navier-Stokes Equations," *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 3, 1993, pp. 457–497. doi:10.1002/nme.1620360307, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620360307.

[21] Wang, Q. S., and wei Shi, X., "An Improved Algorithm for Matrix Bandwidth and Profile Reduction in Finite-Element Analysis," *Progress in Electromagnetic Research Letters*, Vol. 9, 2009, pp. 29–38. doi:doi:10.2528/PIERL09042305.

[22] Kumfert, G., and Pothen, A., "Two Improved Algorithms for Envelope and Wavefront Reduction," *BIT Numerical Mathematics*, Vol. 37, No. 3, 1997, pp. 559–590. doi:10.1007/BF02510240, URL https://doi.org/10.1007/BF02510240.

[23] Gibbs, N. E., Poole, W. G., and Stockmeyer, P. K., "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *SIAM Journal on Numerical Analysis*, Vol. 13, No. 2, 1976, pp. 236–250. URL http://www.jstor.org/stable/2156090.

[24] Reid, J. K., and Scott, J. A., "Ordering Symmetric Sparse Matrices for Small Profile and Wavefront," *International Journal for Numerical Methods in Engineering*, Vol. 45, No. 12, 1999, pp. 1737–1755. doi:10.1002/(SICI)1097-0207(19990830)45:12<1737::AID-NME652>3.0.CO;2-T, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0207%2819990830%2945%3A12%3C1737%3A%3AAID-NME652%3E3.0.CO%3B2-T.

[25] Reid, J. K., and Scott, J. A., "Reducing the Total Bandwidth of a Sparse Unsymmetric Matrix," *SIAM Journal on Matrix Analysis and Applications*, Vol. 28, No. 3, 2006, pp. 805–821. doi:10.1137/050629938, URL https://doi.org/10.1137/050629938.

[26] Marti, R., Laguna, M., Glover, F., and Campos, V., "Reducing the Bandwidth of a Sparse Matrix with Tabu Search," *European Journal of Operational Research*, Vol. 135, No. 2, 2001, pp. 450 – 459. doi:https://doi.org/10.1016/S0377-2217(00)00325-8, URL http://www.sciencedirect.com/science/article/pii/S0377221700003258, financial Modelling.

[27] Mladenovic, N., Urosevic, D., Pérez-Brito, D., and García-González, C. G., "Variable Neighbourhood Search for Bandwidth Reduction," *European Journal of Operational Research*, Vol. 200, No. 1, 2010, pp. 14 – 27. doi:https://doi.org/10.1016/j.ejor.2008.12.015, URL http://www.sciencedirect.com/science/article/pii/S0377221708010540.

[28] Duff, I. S., and Meurant, G. A., "The Effect of Ordering on Preconditioned Conjugate Gradients," *BIT*, Vol. 29, No. 4, 1989, p. 635–657. doi:10.1007/BF01932738, URL https://doi.org/10.1007/BF01932738.

[29] Dutto, L. C., "The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations," *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 3, 1993, pp. 457–497. doi:10.1002/nme.1620360307, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620360307.

[30] Benzi, M., "Preconditioning Techniques for Large Linear Systems: A Survey," *Journal of Computational Physics*, Vol. 182, No. 2, 2002, pp. 418 – 477. doi:https://doi.org/10.1006/jcph.2002.7176, URL http://www.sciencedirect.com/science/article/pii/S0021999102971767.

[31] Benzi, M., Szyld, D. B., and van Duin, A., "Orderings for Incomplete Factorization Preconditioning of Nonsymmetric Problems," *SIAM Journal on Scientific Computing*, Vol. 20, No. 5, 1999, pp. 1652–1670. doi:10.1137/S1064827597326845, URL https://doi.org/10.1137/S1064827597326845.

[32] Chernesky, M. P., "On preconditioned Krylov subspace methods for discrete convection–diffusion problems," *Numerical Methods for Partial Differential Equations*, Vol. 13, No. 4, 1997, pp. 321–330. doi:10.1002/(SICI)1098-2426(199707)13:4<321::AID-NUM2>3.0.CO;2-N.

[33] D'Azevedo, E. F., Forsyth, P. A., and Tang, W.-P., "Ordering Methods for Preconditioned Conjugate Gradient Methods Applied to Unstructured Grid Problems," *SIAM Journal on Matrix Analysis and Applications*, Vol. 13, No. 3, 1992, pp. 944–961. doi:10.1137/0613057, URL https://doi.org/10.1137/0613057.

[34] Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

[35] Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *Journal of Aircraft*, Vol. 55, No. 2, 2018, pp. 696–714. doi:10.2514/1.C034482.

[36] Wigton, L., Yu, N., and Young, D., "GMRES Acceleration of Computational Fluid Dynamics Codes," AIAA Paper 85-1494, AIAA, 1985. doi:10.2514/6.1985-1494.

[37] Wood, S., Jones, W., Jacobson, K., and Anderson, W., "Sparse Linear Algebra Toolkit for Computational Aerodynamics," AIAA Paper 2020-0317, AIAA, 2020. doi:10.2514/6.2020-0317, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0317.

[38] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.12, Argonne National Laboratory, 2019. URL https://www.mcs.anl.gov/petsc.

[39] Poole, G., and Neal, L., "The Rook's pivoting strategy," *Journal of Computational and Applied Mathematics*, Vol. 123, No. 1, 2000, pp. 353 – 369. doi:https://doi.org/10.1016/S0377-0427(00)00406-4, URL http://www.sciencedirect.com/science/article/pii/S0377042700004064, numerical Analysis 2000. Vol. III: Linear Algebra.

[40] Larson, J. L., and Sameh, A. H., "Algorithms for Roundoff Error Analysis —A relative Error Approach," *Computing*, Vol. 24, No. 4, 1980, pp. 275–297. doi:10.1007/BF02237815, URL https://doi.org/10.1007/BF02237815.

[41] Bauer, F. N., "Computational Graphs and Rounding Errors," *SIAM J. Num. Anal.*, Vol. 11, No. 1, 1974, pp. 87–96. doi:10.1137/0711010, URL https://doi.org/10.1137/0711010.

[42] Higham, N. J., *Accuracy and Stability of Numerical Algorithms*, 2nd ed., Society for Industrial and Applied Mathematics, 2002. doi:10.1137/1.9780898718027, URL https://epubs.siam.org/doi/abs/10.1137/1.9780898718027.

[43] Doug S. Lacy, A. J. S., "Development of the High Lift Common Research Model (HL-CRM): A Representative High Lift Configuration for Transonic Transports," AIAA Paper 2016–0308, 2016.

[44] Kennedy, G. J., and Martins, J., "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, 2010. AIAA 2010-9308.

[45] Kennedy, G. J., and Martins, J., "A Parallel Finite-element Framework for Large-scale Gradient-based Design Optimization of High-performance Structures," *Finite Elements in Analysis and Design*, Vol. 87, 2014, pp. 56–73.

[46] Kenway, G., Kennedy, G., and Martins, J., "Aerostructural Optimization of the Common Research Model Configuration," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2014. doi:doi:10.2514/6.2014-3274, URL https://doi.org/10.2514/6.2014-3274.

[47] Bathe, K.-J., and Dvorkin, E. N., "A Formulation of General Shell Elements - the use of Mixed Interpolation of Tensorial Components," *International Journal for Numerical Methods in Engineering*, Vol. 22, No. 3, 1986, pp. 697–722. doi:10.1002/nme.1620220312.

[48] Rivera, J. A., E., D. B., Bennett, R. M., H., D. M., and A., S. W., "NACA 0012 Benchmark Model Experimental Flutter Results with Unsteady Pressure Distributions," *33rd AIAA Structures, Structural Dynamics and Materials Conference*, Dallas, Texas, 1992.

[49] Stanford, B. K., Jacobson, K. E., Heeg, J., Chwalowski, P., and Massey, S. J., "Transonic Aeroelastic Modeling of the NACA 0012 Benchmark Wing," *AIAA Aviation Forum*, Reno Nevada (submitted for publication), 2020.

[50] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X.

[51] Jacobson, K., Stanford, B., Wood, S., and Anderson, W. K., "Flutter Analysis with Stabilized Finite Elements based on the Linearized Frequency-domain Approach," *58th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2020.

[52] Park, M. A., Kleb, W. L., Jones, W. T., Krakos, J. A., Michal, T. R., Loseille, A., Haimes, R., and Dannenhoffer, J., "Geometry Modeling for Unstructured Mesh Adaptation," *AIAA Aviation Forum*, Dallas, Texas, 2019.

[53] Kleb, W. L., Park, M. A., Wood, W. A., Bibb, K. L., Thompson, K. B., and Gomez, R. J., "Sketch-to-Solution: An Exploration of Viscous CFD with Automatic Grids," *AIAA Aviation Forum*, Dallas, Texas, 2019.