Report 4017

A Comparative Study of Several Core Storage Schemes for Large Sparse Positive Definite Matrices with Reference to the Cholesky Algorithm

NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20034

[redacted]

A COMPARATIVE STUDY OF SEVERAL CORE STORAGE SCHEMES FOR LARGE SPARSE POSITIVE DEFINITE MATRICES WITH REFERENCE TO THE CHOLESKY ALGORITHM

by

[redacted] Donald A. Gignac

Approved for public release; distribution unlimited

[redacted]

COMPUTATION AND MATHEMATICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

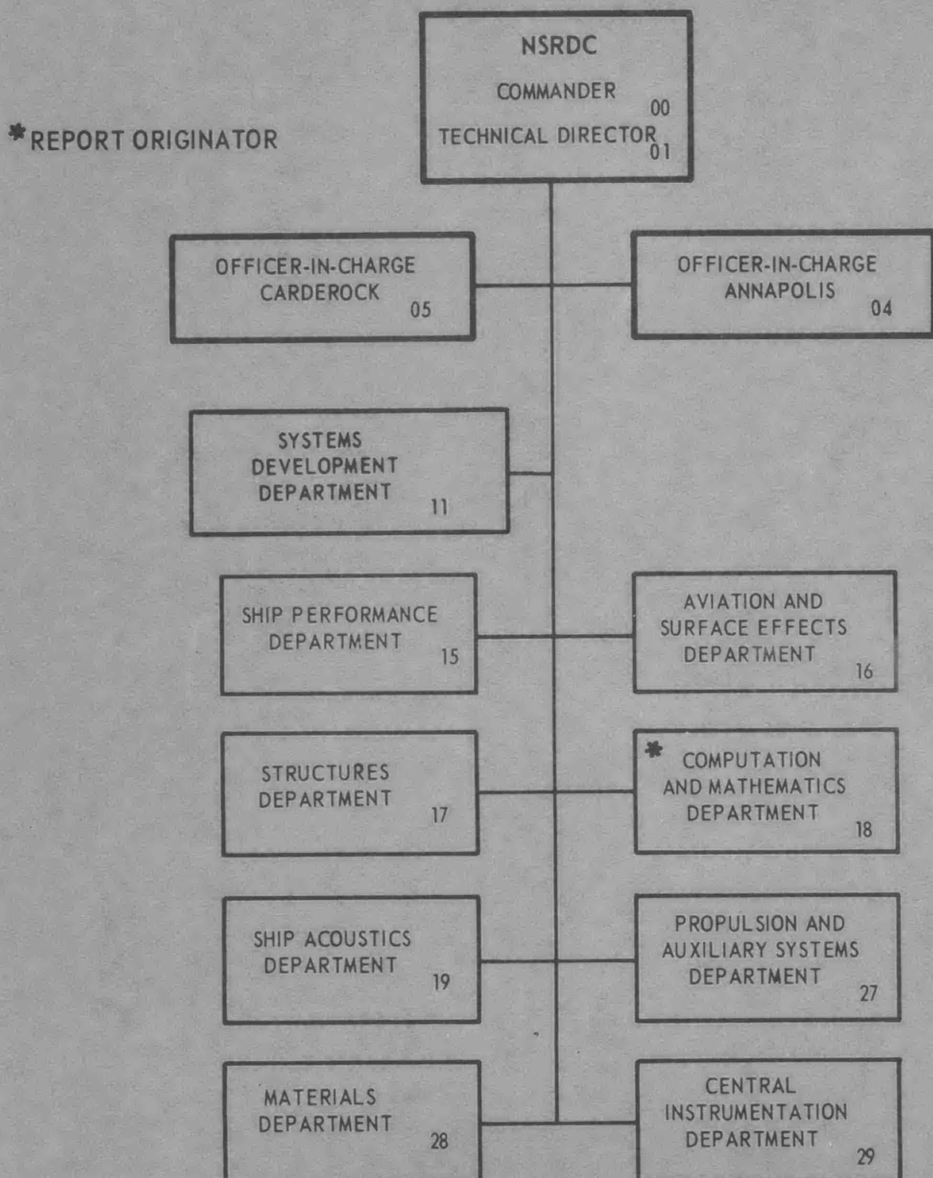
November 1972

Report 4017

The Naval Ship Research and Development Center is a U. S. Navy center for laboratory effort directed at achieving improved sea and air vehicles. It was formed in March 1967 by merging the David Taylor Model Basin at Carderock, Maryland with the Marine Engineering Laboratory at Annapolis, Maryland.

Naval Ship Research and Development Center
Bethesda, Md. 20034

MAJOR NSRDC ORGANIZATIONAL COMPONENTS



NDW-NSRDC 3960/44 (REV. 8/71)
GPO 917-872

DEPARTMENT OF THE NAVY
NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER
BETHESDA, MD. 20034

A COMPARATIVE STUDY OF SEVERAL CORE STORAGE SCHEMES FOR LARGE SPARSE POSITIVE DEFINITE MATRICES WITH REFERENCE TO THE CHOLESKY ALGORITHM

by

Donald A. Gignac

Approved for public release; distribution unlimited.

November 1972 Report 4017

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

In the finite element approach to static structural analysis, the solution of the equation

$$KU = P$$

a positive definite system of simultaneous linear equations, is basic. Considerable difficulty may be experienced when K is very large and sparse. This report documents an investigation of several FORTRAN subroutines in order to obtain an efficient Cholesky algorithm subroutine with economical core storage for an in-core solution of KU = P for large sparse K.

The following conclusions were drawn from this study:

- In general there seems to be considerable advantage to consolidating the forward pass of the back-substitution with the Cholesky row decomposition.

- For large sparse matrices, particularly where the bandwidth MB is much smaller than the order N, the modified first linear scheme subroutine CSKYLIN is undoubtedly the best subroutine of those considered. For banded matrices where the band is relatively compact and not too wide, the band scheme subroutine CSYBD1 does not fall too far behind CSKYLIN. However, for sparse positive definite matrices with "active columns" (as in the second example), CSKYVBD is definitely superior to CSKYLIN in both time and core storage requirements.

ADMINISTRATIVE INFORMATION

The Naval Ship Systems Command (0311) sponsored this study under Subproject SR 014 03 01, Task 15322. The work was performed under work unit number 1-1844-004.

INTRODUCTION

One of the long range projects of the Computation and Mathematics Department is the development of mathematical subroutines suitable for use in the computer-aided structural analysis of ships. Many individual efforts, in both government and industry, have led to computer programs for treating particular special classes of structural problems, and although similar mathematical problems arise in many structural areas, they have been treated as individual problems. The results have often been unsatisfactory. The need to coordinate these diverse efforts, to develop improved methods of more general applicability, and to produce more comprehensive programs for solving Navy structural problems became obvious. A project was therefore established to coordinate research efforts involving mathematical and computational methods in the area of structural mechanics and to integrate the work of mathematicians, computer specialists, and structural engineers in this field.

The present considerable interest in the finite element approach to structural analysis is evidenced by the widespread use of NASTRAN and other such programs. According to the NASTRAN Theoretical Manual[1], "From a theoretical viewpoint, the formulation of a static structural problem for solution by the displacement method is completely described by the matrix equation KU = P". Thus there is a need for efficient subroutines capable of solving these large sparse positive definite systems

[1] "The NASTRAN Theoretical Manual," R.H. MacNeal, editor, National Aeronautics and Space Administration, Washington, D.C., 1969.

of simultaneous linear equations. This report documents an investigation into various core storage schemes (usual, band, linear) for K to obtain an efficient Cholesky algorithm subroutine (in FORTRAN) for the in-core solution of KU = P for incorporation into such large scale structural analysis programs. Future work in this area will be directed towards the development of an out-of-core solution faculty for very large sparse K.

THE CHOLESKY ALGORITHM

The Cholesky algorithm solution of a system of simultaneous linear equations AX = B (where A is positive definite) consists of the following two steps:

- The factorization of A into the product of a lower triangular matrix S and the transpose of S, that is, finding S such that $A = SS^T$.
- The joint solution of the equivalent pair of triangular systems

$$SY = B \tag{1}$$

$$S^T X = Y \tag{2}$$

Both of these triangular systems are easily solved by back-substitution (forward for Equation (1) and backward for Equation (2)).

The solution of the Cholesky algorithm for the solution of the stiffness equation KU = P was influenced by the following factors:

(a) The Cholesky algorithm would appear to have excellent potential for minimizing core storage since only the lower (upper) triangular part of A and only the lower triangular part of S (upper triangular part of S^T) need be stored, and that part of S or S^T may be written over the stored part of A as S or S^T is calculated.

(b) The Cholesky decomposition is numerically stable and does not require pivoting.[2]

The Cholesky decomposition of A can be carried out by rows or by columns. Row decomposition is to be preferred because then the solution of system (1) can be included. The astute programmer can incorporate the dot product accumulation of the solution of the i^{th} equation of system (1) in the column loop of Cholesky row decomposition as Jennings does.[3] This combination of the decomposition and the forward pass of the back-substitution is much faster than are the two procedures performed separately, as can be seen from the test examples. The formulas for the joint decomposition and forward back-substitution solution of system (1) are:

$$s_{11} = \sqrt{a_{11}}$$

$$y_1 = b_1/s_{11}$$

(continued next page)

---

[2] Wilkinson, J. H., "The Algebraic Eigenvalue Problem," Clarendon Press, Oxford, 1965, p. 229-32.

[3] Jennings, A., "Algorithm 70 Solution of Variable Bandwidth Positive Definite Simultaneous Equations," The Computer Journal, Vol. 14, p. 446, 1971.

$$\left.\begin{aligned}
s_{i1} &= a_{i1}/s_{11} \\
s_{ij} &= (a_{ij} - \sum_{k=1}^{j-1} s_{ik}\, s_{jk})/s_{jj} \quad j = 2, \ldots, i-1 \\
s_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2} \\
y_i &= (b_i - \sum_{k=1}^{i-1} s_{ik}\, b_k)/s_{ii}
\end{aligned}\right\} \quad i = 2, \ldots, n$$

The formulas for the backward back-substitution are:

$$x_n = y_n/s_{nn}$$

$$x_i = (y_i - \sum_{k=i+1}^{n} s_{ki}\, y_k)/s_{ii} \qquad i = n-1, \ldots, 1$$

It can be seen from these equations that the feedback solution of system (2) consists of forming dot products with the rows of S^T, an awkward procedure since we must compute with the columns of S (which are difficult to reference in our linear core storage schemes). By the clever device of accumulating these dot products concurrently, Jennings[3] exchanged the problem of referencing the columns of S for the simpler problem of computing with the rows of S from top to bottom and from right to left. It was conjectured that this device would prove to be faster than referencing the columns of S directly. Unfortunately, this does not appear to be the case, as can be seen from the test examples.

FIVE LINEAR CORE STORAGE SCHEMES

We shall consider the following types of core storage schemes for sparse positive definite matrices:

(1) usual
(2) band
(3) various linear

In the usual scheme, all the elements a_{ij} of the $n \times n$ full matrix A are stored as elements A(I, J) of an NxN array A. Clearly this scheme affords the easiest access to the elements a_{ij}, but it is quite wasteful of core storage except for rather small matrices (CHLSKY subroutine).

In the band scheme the diagonal and the non-zero lines under and parallel to the diagonal are stored as rows of a rectangular array. That is, the elements a_{ij} $(i \le j)$ of these non-zero lines are stored as the elements A(I-(J-I), J) of the MB x N array A where the value of MB is the number of these non-zero lines. The value of MB is called the bandwidth of the matrix A. For example, the 3-band matrix (i.e., of bandwidth 3) of order 5

$$\begin{bmatrix} 12.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 12.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 12.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 12.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 12.0 \end{bmatrix}$$

is stored as the 3x5 array

$$\begin{matrix} 12.0 & 12.0 & 12.0 & 12.0 & 12.0 \\ 1.0 & 1.0 & 0.0 & 0.0 & * \\ 0.0 & 1.0 & 0.0 & * & * \end{matrix}$$

The *'s indicate that that particular array element need not be defined. This scheme works well for closely packed banded matrices but is inefficient for matrices with isolated non-zero elements. S is stored in the same way since S has the same bandwidth as A (CSYBD1 subroutine).

The first linear core storage scheme stores the required elements of the upper triangular portion of the matrix by stringing them out by rows in a linear array. This scheme is adapted from that of Jennings[4]. For example, the 3-band matrix (i.e., of bandwidth 3) of order 5

$$\begin{bmatrix} 12 & 1 & 0 & 0 & 0 \\ 1 & 12 & 1 & 1 & 0 \\ 0 & 1 & 12 & 0 & 0 \\ 0 & 1 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

is stored as the two linear arrays

1	3	6	8	9				
12	1	12	1	1	12	0	12	12

The first array tells where the diagonal elements of the matrix are in the second array.

Two points should be noted regarding this scheme. First, in our example the $a_{34} = a_{43}$ elements of A are zero but the s_{34} element of S will not be zero and its formula will require the value of the $a_{34} = a_{43}$ element of A. Accordingly, we must store a zero in the seventh slot of the second array. In general, whenever an element a_{ij} $(j > i)$ is zero and there is at least one non-zero element a_{kj} $(k < i)$ above it in the same

[4] Jennings, A., "A Compact Storage Scheme for the Solution of Symmetric Linear Simultaneous Equations," The Computer Journal, Vol. 9, pp. 281-285, 1966.

column, that zero element must be stored because the formula for s_{ij} requires the value of a_{ij}. If all such zeroes in the upper triangular portion of A are stored in this way, then clearly S^T can be stored in a linear array of the same length as that of A with its diagonal elements in the same locations as those of A (CMPCT1 subroutine).

Secondly, in the computation of S^T it is necessary to compute dot products of the columns of S^T an increment at a time. This procedure requires repeated row scannings of the previously computed elements of S^T. If the matrix in question can be assigned a bandwidth MB significantly smaller than N, then a considerable amount of unnecessary work in the Cholesky decomposition alone can be avoided merely by restricting this row scanning to the banded strip (since all the non-zero elements of S^T will fall within it)(CSKYLIN subroutine).

The second linear core storage scheme stores the required elements of the lower triangular portion of the matrix by stringing them out by rows (starting with the leading non-zero element of the row) in a linear array. For example, the above matrix is stored as the two linear arrays

1	2	4	6	9				
12	1	12	1	12	1	0	12	12

where the first array gives the location of the leading non-zero elements of each row in the second array.

Two points should be noted regarding this scheme. First, the location of matrix elements in the linear array is facilitated by the introduction of a third array of integers which provides the column index of the leading non-zero element of each row. For the above matrix this array is

1 1 2 2 5

Second, the introduction of a bandwidth MB in this core storage scheme does not reduce the work in the Cholesky decomposition as it does in the first linear core storage scheme. The only place in the Cholesky algorithm to utilize the bandwidth MB (in this second core storage scheme) is in the backward pass of the back-substitution (CSKYJEN subroutine).

The third linear core storage scheme is that used by Jennings in a recent ALGOL procedure[3]. It combines features of the two preceding schemes: the required elements of the lower triangular portion of the matrix are strung out by rows in a linear array, with another array locating the diagonal elements in the first array. For example, the above matrix is stored as the two arrays

1	3	5	8	9				
12	1	12	1	12	1	0	12	12

(SYMVBAJ and CSKYVBD subroutines).

TIMING PROCEDURE

The following points should be noted with regard to the timing of computations on the CDC 6700 computer:

(a) It is necessary to specify both a central processing unit (CPU) and a FORTRAN compilation mode to get a meaningful (i.e., reproducible) time for a computation. The dual processing feature of the CDC 6700 patches together time on both the 6600 CPU (unit A) and 6400 CPU (unit B). Since unit A is in general much faster than unit B, and since unit A makes use of parallel processing while unit B does not, the total time for such a run will depend largely on how this patching is done. In general, unit A times are about a third of unit B times. The choice of CPU is indicated by the CPA or CPB option on the job card.

(b) The four modes of FORTRAN compilation on the CDC 6700 differ significantly in their compilation and execution times. The mode of FORTRAN compilation is indicated by the following options on the FTN card:

OPT = 0	(fast, sloppy compilation, slow execution)
OPT = 1	(standard, default option)
OPT = 2	(slow, efficient compilation, fast execution)
C	(uses COMPASS assembler)

In addition, it is not inconceivable that, for a given CPU and FORTRAN compilation mode, compiler modifications may affect computation times appreciably.

(c) At present the SECONDS timing subroutine does not work satisfactorily on unit A. Occasionally, several integral seconds are added to the times (e.g., 2.4918 seconds instead of 0.4918 seconds), producing some uncertainty about the unit A times.

TEST EXAMPLES

FIRST EXAMPLE

Tables 1a and 1b illustrate the time advantage of the modified over the unmodified first linear core storage scheme subroutine. The results in Table 1a were obtained by solving the system AX = B where A is the 2-banded matrix of order N (N = 5, 100, 5) with 2's on the diagonal and a line of -1's above and below the diagonal[5] and where

[5] Westlake, J.R., "A Handbook of Numerical Matrix Inversion and Solution of Linear Equations," p. 140, no. 15, John Wiley, New York, 1968.

the Nx1 vector B is chosen so that the solution vector X has all coordinates equal to 1. The times in the "Unmodified" columns were obtained using the first LINEUP subroutine and the CMPCT1 subroutine; those in the "Modified" columns by the second LINEUP subroutine and the CSKYLIN subroutine. The results in Table 1b were obtained similarly, with N = 100, 1000, 25. However, this time the LINEUP subroutines were not used since it was necessary to generate very large matrices in the appropriate condensed core storage form.

Table 2a shows comparative times for the first three core storage scheme subroutines for the solution of the above system AX = B for N = 5, 100, 5. These results were obtained using the CHLSKY subroutine (usual), the SETUP and CSYBD1 subroutines (band), the second LINEUP subroutine and the CSKYLIN subroutines (modified linear as above). Table 2b presents similar results for larger orders (N = 100, 1000, 25) of this system for the band and modified linear core storage scheme subroutines. Again, for large N, A must be generated in the appropriate condensed core storage form.

SECOND EXAMPLE

Table 3 gives comparative times for the subroutines of Table 2a for the solution of the system AX = B, where A is a full matrix with 10 on the diagonal and 1's elsewhere and B is chosen so that all the coordinates of X are 1. *

---

* See Westlake, p. 141, no. 18.

THIRD EXAMPLE

The next example illustrates the time reduction provided by the Cuthill-McKee algorithm for renumbering nodes (finite element static analysis of a structural problem). Let N be an integer ≥ 3. Let B_N be the 2-banded matrix of order N with 4's on the diagonal and a line of -1's above and below the diagonal. Let I_N be the identity matrix of order N. We construct an (N+1)-banded matrix of order N^2, C_{N^2}, by stringing N B_N submatrices along the diagonal with a line of $N-1$ $-I_N$ submatrices above and below the B_N, and by setting the remaining elements of C_{N^2} equal to 0.

A unique permutation of $1, 2, \ldots, N^2$ transforms the connection table

$$\begin{array}{cccc}
1 & 2 & \cdots & N \\
N+1 & N+2 & \cdots & 2N \\
\vdots & \vdots & & \vdots \\
 & & \cdots & N^2
\end{array}$$

into the connection table

$$\begin{array}{llll}
1 & 2 & 4 & 7\ldots \\
3 & 5 & 8\ldots & \\
6 & 9\ldots & & \\
10\ldots & & & \\
 & & & \vdots \\
 & & & \ldots N^2
\end{array}$$

Let P_{N^2} be the matrix obtained by applying this permutation to the columns of I_{N^2}. The transformation induced by the Cuthill-McKee algorithm is

$$P_{N^2} C_{N^2} P^T_{N^2} .$$

Let D_{N^3} be the transformed matrix. D_{N^3} has the same bandwidth as the original matrix but the ends of the banded area are pinched, giving the band a somewhat convex contour.[6]

Tables 4a and 4b give the Cholesky algorithm solution for both core storage schemes of $C_{N^3} X = F$ and $D_{N^3} X = F$, respectively, for $N = 3, \ldots, 32$, where in each instance F is chosen so that all the coordinates of the solution X are 1. These tables give the CPB, OPT=1 times (in seconds) for the entire solution, the Cholesky decomposition, the back-substitution, the forward pass, and the backward pass.

FOURTH EXAMPLE

For $N = 3, \ldots, 32$ let G_{N^3} be the sparse positive definite matrix defined thus: Take the matrix B_{N^3} (of order N^3) defined in the previous example. Redefine the off-diagonal elements of the last row and column and the first elements of the preceding row and the column to be -1. The last two columns of G_{N^3} are said to be active. Application of the Gerschgorin circle theorem shows G_{N^3} to be positive definite[2]. Table 5 gives times for the Cholesky algorithm solution of $G_{N^3} X = F$, where F is chosen so that all the coordinates of the solution X are 1.

SUBROUTINES USED

The above Cholesky algorithm subroutines CSKYLIN and CSKYJEN use the first and second linear core storage schemes, respectively. SYMVBAJ is a FORTRAN version of SYMVBOL, Jennings' ALGOL procedure[3]. CSKYVBD is a modification of SYMVBAJ which uses the Cholesky decomposition coding of CSKYJEN in the decomposition-back-substitution procedure.

---

[6] Cuthill, E. and McKee, J., "Reducing the Bandwidth of Sparse Symmetric Matrices," Proceedings 24th National Conference of the ACM, pp. 157-172, 1969.

OBSERVATIONS

Tables 4a and 4b show that CSKYLIN is noticeably faster than the other three subroutines in total time, decomposition time, and the time for the backward pass of the back-substitution. On the other hand, Table 5 shows that CSKYVBD has the fastest time for the same three procedures and that the CSKYLIN decomposition times are grossly larger than the other decomposition times. Some comment is in order. In general, the "bookkeeping" (i.e., locating the matrix elements in the condensed core storage form) of CSKYJEN is more cumbersome than that of the other subroutines. In particular, the subscripts of CSKYJEN are definitely more complicated than those of the other subroutines. (For example, compare the codings of the backward pass of the back-substitution.) As a result CSKYJEN is significantly slower than the other subroutines, since the COMPASS assembly of CSKYJEN is significantly longer than that of the other subroutines.

This example was included for two reasons. First, problems of this type occur regularly in structural analysis and, secondly, the form of the coefficient matrix would appear to favor the other subroutines over CSKYLIN. This proved to be the case. Their considerable advantage seems to stem chiefly from the fact that the dot-products of the Cholesky decomposition are computed directly without the row-scannings of CSKYLIN which, for this example, must cover the entire upper triangular portion of the matrix. In addition, they require far less core storage than CSKYLIN in this instance.

TABLE 1A
THE TIME ADVANTAGE OF THE MODIFIED LINEAR CORE STORAGE SUBROUTINE OVER THE UNMODIFIED LINEAR CORE STORAGE SUBROUTINE FOR SMALL ORDERS OF A 2-BANDED MATRIX

NOTE: The procedure timed below includes reducing the matrix to the appropriate core storage form.

UNIT A			UNIT B		
N	Unmodified	Modified	N	Unmodified	Modified
5	.0020	.0030	5	.0040	.0040
10	.0030	.0030	10	.0080	.0090
15	.0050	.0050	15	.0130	.0130
20	.0070	.0070	20	.0230	.0200
25	.0100	.0080	25	.0300	.0240
30	.0140	.0110	30	.0390	.0330
35	.0170	.0140	35	.0500	.0410
40	.0220	.0170	40	.0640	.0490
45	.0270	.0180	45	.0780	.0600
50	.0320	.0230	50	.0910	.0690
55	.0390	.0270	55	.1090	.0780
60	.0460	.0300	60	.1350	.0890
65	.0510	.0330	65	.1500	.1030
70	.0580	.0380	70	.1730	.1150
75	.0660	.0420	75	.1900	.1300
80	.0750	.0470	80	.2130	.1430
85	.0820	.0520	85	.2380	.1600
90	.0930	.0580	90	.2670	.1770
95	.1000	.0640	95	.2930	.1940
100	.1080	.0690	100	.3200	.2150

TABLE 1B
THE TIME ADVANTAGE OF THE MODIFIED LINEAR CORE STORAGE SUBROUTINE OVER THE UNMODIFIED LINEAR CORE STORAGE SUBROUTINE FOR LARGE ORDERS OF A 2-BANDED MATRIX

NOTE: The procedure timed below does not include generating the matrix in the appropriate core storage form.

UNIT A			UNIT B		
N	Unmodified	Modified	N	Unmodified	Modified
100	.0540	.0130	100	.1560	.0410
125	.0820	.0170	125	.2310	.0510
150	.1160	.0200	150	.3220	.0600
175	.1530	.0240	175	.4260	.0720
200	.1970	.0290	200	.5480	.0820
225	.2460	.0310	225	.6820	.0910
250	.3000	.0330	250	.8350	.1010
275	.3600	.0370	275	1.0040	.1110
300	.4270	.0400	300	1.1830	.1220
325	.5000	.0430	325	1.3780	.1310
350	.5790	.0460	350	1.5920	.1410
375	.6730	.0490	375	1.8190	.1510
400	.7610	.0530	400	2.1470	.1620
425	.8560	.0560	425	2.4570	.1710
450	.9610	.5090	450	2.7570	.1810
475	1.0570	.0630	475	3.0610	.1930
500	1.1810	.0650	500	3.3800	.2030
525	1.3000	.0690	525	3.7200	.2120
550	1.4170	.0720	550	4.0710	.2220
575	1.5380	.0740	575	4.4420	.2310
600	1.6700	.0780	600	4.8250	.2420
625	1.8130	.0830	625	5.2320	.2530
650	1.9610	.0850	650	5.6440	.2630
675	2.1160	.0900	675	6.0770	.2740
700	2.2760	.0910	700	6.5240	.2820
725	2.4390	.0970	725	6.9910	.2930
750	2.6060	.0980	750	7.4730	.3020
775	2.7780	.1010	775	7.9680	.3140
800	2.9600	.1070	800	8.4800	.3230
825	3.1390	.1090	825	9.0110	.3320
850	3.3360	.1120	850	9.5560	.3430
875	3.5410	.1140	875	10.1280	.3540
900	3.7340	.1200	900	10.6910	.3620
925	3.9460	.1230	925	11.2860	.3730
950	4.1640	.1250	950	11.8920	.3830
975	4.3840	.1270	975	12.5170	.3930
1000	4.6100	.1320	1000	13.1600	.4020

TABLE 2A
THE TIMES FOR THE SUBROUTINES USING THE FIRST THREE CORE STORAGE SCHEMES FOR SMALL ORDERS OF A 2-BANDED MATRIX

UNIT A				UNIT B			
N	Usual	Band	Modified Linear	N	Usual	Band	Modified Linear
5	.0020	.0020	.0030	5	.0040	.0050	.0040
10	.0030	.0030	.0030	10	.0100	.0100	.0090
15	.0060	.0050	.0050	15	.0220	.0170	.0130
20	.0110	.0080	.0070	20	.0420	.0230	.0200
25	.0190	.0100	.0080	25	.0700	.0310	.0240
30	.0310	.0120	.0110	30	.1080	.0390	.0330
35	.0450	.0150	.0140	35	.1581	.0480	.0410
40	.0630	.0180	.0170	40	.2210	.0580	.0490
45	.0880	.0220	.0180	45	.3020	.0680	.0600
50	.1150	.0250	.0230	50	.3980	.0810	.0690
55	.1480	.0300	.0270	55	.4850	.0900	.0780
60	.1890	.0330	.0300	60	.6200	.1040	.0890
65	.2350	.0380	.0330	65	.7710	.1170	.1030
70	.2880	.0420	.0380	70	.9420	.1320	.1150
75	.3480	.0480	.0420	75	1.1450	.1460	.1300
80	.4170	.0540	.0470	80	1.3720	.1640	.1430
85	.4930	.0590	.0520	85	1.6110	.1800	.1600
90	.5810	.0640	.0580	90	1.9160	.1980	.1770
95	.6770	.0720	.0640	95	2.2270	.2160	.1940
100	.7870	.0780	.0690	100	2.5390	.2350	.2150

TABLE 2B
THE TIMES FOR THE BAND AND MODIFIED LINEAR CORE STORAGE SUBROUTINES FOR LARGE ORDERS OF A 2-BANDED MATRIX

UNIT A			UNIT B		
N	Band	Modified Linear	N	Band	Modified Linear
100	.0210	.0130	100	.0660	.0410
125	.0260	.0170	125	.0810	.0510
150	.0310	.0200	150	.0970	.0600
175	.0360	.0240	175	.1130	.0720
200	.0420	.0290	200	.1300	.0820
225	.0470	.0310	225	.1450	.0910
250	.0520	.0330	250	.1620	.1010
275	.0570	.0370	275	.1780	.1110
300	.0620	.0400	300	.1940	.1220
325	.0680	.0430	325	.2100	.1310
350	.0730	.0460	350	.2270	.1410
375	.0780	.0490	375	.2450	.1510
400	.0830	.0530	400	.2590	.1620
425	.0880	.0560	425	.2740	.1710
450	.0930	.0590	450	.2900	.1810
475	.0980	.0630	475	.3060	.1930
500	.1040	.0650	500	.3230	.2030
525	.1080	.0690	525	.3400	.2120
550	.1130	.0720	550	.3550	.2220
575	.1190	.0740	575	.3710	.2310
600	.1240	.0780	600	.3870	.2420
625	.1280	.0830	625	.4030	.2530
650	.1340	.0850	650	.4210	.2630
675	.1400	.0900	675	.4350	.2740
700	.1440	.0910	700	.4520	.2820
725	.1530	.0970	725	.4680	.2930
750	.1550	.0980	750	.4860	.3020
775	.1630	.1010	775	.5010	.3140
800	.1660	.1070	800	.5160	.3230
825	.1730	.1090	825	.5310	.3320
850	.1790	.1120	850	.5470	.3430
875	.1820	.1140	875	.5640	.3540
900	.1880	.1200	900	.5810	.3620
925	.1930	.1230	925	.5970	.3730
950	.1980	.1250	950	.6140	.3830
975	.2020	.1270	975	.6300	.3930
1000	.2090	.1320	1000	.6450	.4020

TABLE 3
THE TIMES FOR THE SUBROUTINES USING THE FIRST THREE CORE STORAGE SCHEMES FOR SMALL ORDERS OF A FULL MATRIX

UNIT A				UNIT B			
N	Usual	Band	Modified Linear	N	Usual	Band	Modified Linear
5	.0010	.0020	.0020	5	.0030	.0090	.0060
10	.0040	.0070	.0050	10	.0110	.0240	.0140
15	.0060	.0170	.0080	15	.0260	.0580	.0320
20	.0130	.0330	.0150	20	.0490	.1190	.0590
25	.0200	.0540	.2050	25	.0830	.2100	.1000
30	.0300	.0870	.0390	30	.1300	.3390	.1550
35	.0450	.1280	.0530	35	.1950	.5130	.2300
40	.0650	.1840	.0770	40	.2750	.7360	.3270
45	.0870	.2570	.0990	45	.3760	1.0190	.4460
50	.1150	.3400	.1310	50	.4980	1.3640	.5930
55	.1510	.4430	.1670	55	.6670	1.8120	.7900
60	.1880	.5630	.2090	60	.8400	2.3220	.9860
65	.2360	.7060	.2610	65	1.0320	2.9060	1.2280
70	.2910	.8650	.3180	70	1.2700	3.5980	1.5200
75	.3490	1.0590	.3820	75	1.5390	4.3860	1.8430
80	.4200	1.2690	.4580	80	1.8580	5.2600	2.1900
85	.4980	1.5200	.5420	85	2.1860	6.2050	2.5830
90	.5850	1.7870	.6350	90	2.5530	7.3030	3.0110
95	.6810	2.0680	.7300	95	2.9370	8.4500	3.4980
100	.7840	2.3900	.8430	100	3.3960	9.7980	4.0460

TABLE 4
TIME REDUCTION STUDY FOR THE LINEAR CORE STORAGE SUBROUTINES
RESULTING FROM THE CUTHILL-McKEE TRANSFORMATION

TABLE 4A – SOLUTION OF $C_{N^2}X = F$

CSKYJEN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0080	.0050	.0030	.0010	.0020
16	5	.0160	.0100	.0060	.0030	.0030
25	6	.0280	.0170	.0110	.0050	.0060
36	7	.0500	.0340	.0160	.0070	.0090
49	8	.0810	.0570	.0240	.0110	.0130
64	9	.1260	.0900	.0360	.0150	.0210
81	10	.1830	.1350	.0480	.0200	.0280
100	11	.2640	.1990	.0650	.0270	.0380
121	12	.3650	.2800	.0850	.0370	.0480
144	13	.4900	.3810	.1090	.0470	.0620
169	14	.6470	.5110	.1360	.0570	.0790
196	15	.7900	.6290	.1610	.0690	.0920
225	16	1.0060	.8100	.1960	.0830	.1130
256	17	1.2630	1.0300	.2330	.0990	.1340
289	18	1.5650	1.2860	.2790	.1180	.1610
324	19	1.9530	1.6180	.3350	.1450	.1900
361	20	2.3390	1.9500	.3890	.1630	.2260
400	21	2.7940	2.3460	.4480	.1890	.2590
441	22	3.3350	2.8180	.5170	.2180	.2990
484	23	3.9480	3.3540	.5940	.2500	.3440
529	24	4.7200	4.0210	.6990	.2920	.4070
576	25	5.4520	4.6800	.7720	.3250	.4470
625	26	6.3310	5.4650	.8660	.3640	.5020
676	27	7.4310	6.4200	1.0110	.4240	.5870
729	28	8.3660	7.2820	1.0840	.4540	.6300
784	29	9.5880	8.3800	1.2080	.5060	.7020
841	30	10.9460	9.6050	1.3410	.5650	.7760
900	31	12.3570	10.8760	1.4810	.6210	.8600
961	32	14.1280	12.4450	1.6830	.7010	.9820
1024	33	15.7700	13.9680	1.8020	.7560	1.0460

TABLE 4A (continued)

CSKYLIN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0080	.0050	.0030	.0020	.0010
16	5	.0160	.0100	.0060	.0030	.0030
25	6	.0270	.0180	.0090	.0050	.0040
36	7	.0460	.0320	.0140	.0070	.0070
49	8	.0740	.0530	.0210	.0100	.0110
64	9	.1130	.0830	.0300	.0150	.0150
81	10	.1680	.1260	.0420	.0200	.0220
100	11	.2370	.1810	.0560	.0280	.0280
121	12	.3260	.2540	.0720	.0350	.0370
144	13	.4420	.3490	.0930	.0450	.0480
169	14	.5470	.4360	.1110	.0550	.0560
196	15	.7080	.5740	.1340	.0660	.0680
225	16	.9070	.7410	.1660	.0810	.0850
256	17	1.1560	.9510	.2050	.1000	.1050
289	18	1.4060	1.1710	.2350	.1150	.1200
324	19	1.7290	1.4510	.2780	.1360	.1420
361	20	2.0960	1.7720	.3240	.1590	.1650
400	21	2.5290	2.1550	.3740	.1830	.1910
441	22	3.0040	2.5740	.4300	.2120	.2180
484	23	3.6050	3.0990	.5060	.2490	.2570
529	24	4.2440	3.6740	.5700	.2810	.2890
576	25	4.9710	4.3280	.6430	.3170	.3260
625	26	5.7060	4.9890	.7170	.3520	.3650
676	27	6.6020	5.8000	.8020	.3950	.4070
729	28	7.6320	6.7370	.8950	.4410	.4540
784	29	8.8160	7.7870	1.0290	.5130	.5160
841	30	9.9370	8.8260	1.1110	.5490	.5620
900	31	11.2640	10.0470	1.2170	.6000	.6170
961	32	12.8050	11.4280	1.3770	.6780	.6990
1024	33	14.7100	13.2050	1.5050	.7580	.7470

TABLE 4A (continued)

SYMVBAJ TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0060	.0040	.0020	0.0000	.0020
16	5	.0130	.0100	.0030	0.0000	.0030
25	6	.0260	.0200	.0060	0.0000	.0060
36	7	.0430	.0350	.0080	0.0000	.0080
49	8	.0700	.0570	.0130	0.0000	.0130
64	9	.1080	.0900	.0180	0.0000	.0180
81	10	.1600	.1350	.0250	0.0000	.0250
100	11	.2290	.1960	.0330	0.0000	.0330
121	12	.3190	.2740	.0450	0.0000	.0450
144	13	.4290	.3730	.0560	0.0000	.0560
169	14	.5680	.4970	.0710	0.0000	.0710
196	15	.7370	.6500	.0870	0.0000	.0870
225	16	.9400	.8330	.1070	0.0000	.1070
256	17	1.1850	1.0560	.1290	0.0000	.1290
289	18	1.4730	1.3180	.1550	0.0000	.1550
324	19	1.8100	1.6270	.1830	0.0000	.1830
361	20	2.2010	1.9870	.2140	0.0000	.2140
400	21	2.6550	2.4060	.2490	0.0000	.2490
441	22	3.1740	2.8860	.2880	0.0000	.2880
484	23	3.7530	3.4250	.3280	0.0000	.3280
529	24	4.4230	4.0480	.3750	0.0000	.3750
576	25	5.1710	4.7470	.4240	0.0000	.4240
625	26	6.0100	5.5320	.4780	0.0000	.4780
676	27	7.3280	6.7570	.5710	0.0000	.5710
729	28	7.9930	7.3920	.6010	0.0000	.6010
784	29	9.1510	8.4830	.6680	0.0000	.6680
841	30	10.4860	9.7400	.7460	0.0000	.7460
900	31	11.9070	11.0860	.8210	0.0000	.8210
961	32	13.4020	12.4980	.9040	0.0000	.9040
1024	33	15.0780	14.0840	.9940	0.0000	.9940

TABLE 4A (continued)

CSKYVBD TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0060	.0050	.0010	0.0000	.0010
16	5	.0130	.0100	.0030	0.0000	.0030
25	6	.0220	.0180	.0040	0.0000	.0040
36	7	.0430	.0350	.0080	0.0000	.0080
49	8	.0690	.0580	.0110	0.0000	.0110
64	9	.1060	.0900	.0160	0.0000	.0160
81	10	.1590	.1370	.0220	0.0000	.0220
100	11	.2290	.1990	.0300	0.0000	.0300
121	12	.3180	.2780	.0400	0.0000	.0400
144	13	.4300	.3800	.0500	0.0000	.0500
169	14	.5710	.5070	.0640	0.0000	.0640
196	15	.7420	.6630	.0790	0.0000	.0790
225	16	.9490	.8520	.0970	0.0000	.0970
256	17	1.1940	1.0780	.1160	0.0000	.1160
289	18	1.4860	1.3490	.1370	0.0000	.1370
324	19	1.8270	1.6640	.1630	0.0000	.1630
361	20	2.2240	2.0340	.1900	0.0000	.1900
400	21	2.6820	2.4600	.2220	0.0000	.2220
441	22	3.2070	2.9510	.2560	0.0000	.2560
484	23	3.7940	3.5010	.2930	0.0000	.2930
529	24	4.4690	4.1350	.3340	0.0000	.3340
576	25	5.2250	4.8470	.3780	0.0000	.3780
625	26	6.1380	5.7030	.4350	0.0000	.4350
676	27	7.0240	6.5450	.4790	0.0000	.4790
729	28	8.1390	7.6050	.5340	0.0000	.5340
784	29	9.2490	8.6540	.5950	0.0000	.5950
841	30	10.5490	9.8880	.6610	0.0000	.6610
900	31	11.9720	11.2410	.7310	0.0000	.7310
961	32	13.5300	12.7240	.8060	0.0000	.8060
1024	33	15.2350	14.3500	.8850	0.0000	.8850

TABLE 4B – SOLUTION OF $D_{N^2}X = F$

CSKYJEN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0090	.0050	.0040	.0020	.0020
16	5	.0130	.0080	.0050	.0020	.0030
25	6	.0230	.0150	.0080	.0040	.0040
36	7	.0410	.0250	.0160	.0070	.0090
49	8	.0610	.0410	.0200	.0080	.0120
64	9	.0930	.0640	.0290	.0120	.0170
81	10	.1330	.0930	.0400	.0160	.0240
100	11	.1850	.1320	.0530	.0220	.0310
121	12	.2490	.1810	.0680	.0280	.0400
144	13	.3300	.2420	.0880	.0360	.0520
169	14	.4320	.3230	.1090	.0440	.0650
196	15	.5460	.4140	.1320	.0520	.0800
225	16	.6870	.5270	.1600	.0640	.0960
256	17	.8550	.6620	.1930	.0770	.1160
289	18	1.0440	.8160	.2280	.0900	.1380
324	19	1.2630	.9940	.2690	.1070	.1620
361	20	1.5900	1.2630	.3270	.1290	.1980
400	21	1.8200	1.4530	.3670	.1440	.2230
441	22	2.1520	1.7340	.4180	.1640	.2540
484	23	2.5830	2.0990	.4840	.1860	.2980
529	24	2.9370	2.3940	.5430	.2110	.3320
576	25	3.4080	2.7940	.6140	.2390	.3750
625	26	3.9430	3.2510	.6920	.2690	.4230
676	27	4.5520	3.7770	.7750	.2990	.4760
729	28	5.1830	4.3060	.8770	.3360	.5410
784	29	5.9010	4.9370	.9640	.3710	.5930
841	30	7.0700	5.9470	1.1230	.4320	.6910
900	31	7.9760	6.7430	1.2330	.4760	.7570
961	32	8.5280	7.2130	1.3150	.5080	.8070
1024	33	9.6730	8.1940	1.4790	.5720	.9070

TABLE 4B (continued)

CSKYLIN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0070	.0040	.0030	.0010	.0020
16	5	.0130	.0080	.0050	.0030	.0020
25	6	.0230	.0150	.0080	.0040	.0040
36	7	.0360	.0240	.0120	.0050	.0070
49	8	.0590	.0400	.0190	.0090	.0100
64	9	.0840	.0610	.0230	.0110	.0120
81	10	.1210	.0870	.0340	.0150	.0190
100	11	.1670	.1230	.0440	.0210	.0230
121	12	.2290	.1710	.0580	.0280	.0300
144	13	.2970	.2270	.0700	.0340	.0360
169	14	.3890	.3030	.0860	.0420	.0440
196	15	.4900	.3850	.1050	.0520	.0530
225	16	.6240	.4940	.1300	.0630	.0670
256	17	.7620	.6110	.1510	.0740	.0770
289	18	.9350	.7560	.1790	.0870	.0920
324	19	1.1390	.9300	.2090	.1030	.1060
361	20	1.3670	1.1230	.2440	.1190	.1250
400	21	1.6390	1.3570	.2820	.1370	.1450
441	22	1.9780	1.6470	.3310	.1630	.1680
484	23	2.2610	1.8950	.3660	.1800	.1860
529	24	2.6940	2.2720	.4220	.2060	.2160
576	25	3.0740	2.6050	.4690	.2300	.2390
625	26	3.5620	3.0340	.5280	.2590	.2690
676	27	4.1460	3.5530	.5930	.2890	.3040
729	28	4.6670	4.0060	.6610	.3210	.3400
784	29	5.5210	4.7880	.7330	.3610	.3720
841	30	6.0680	5.2560	.8120	.3980	.4140
900	31	6.7870	5.8940	.8930	.4370	.4560
961	32	7.9150	6.8900	1.0250	.5030	.5220
1024	33	8.5870	7.5240	1.0630	.5220	.5410

TABLE 4B (continued)

SYMVBAJ TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0060	.0040	.0020	0.0000	.0020
16	5	.0120	.0090	.0030	0.0000	.0030
25	6	.0210	.0160	.0050	0.0000	.0050
36	7	.0360	.0290	.0070	0.0000	.0070
49	8	.0560	.0460	.0100	0.0000	.0100
64	9	.0830	.0690	.0140	0.0000	.0140
81	10	.1220	.1010	.0210	0.0000	.0210
100	11	.1681	.1410	.0270	0.0000	.0270
121	12	.2300	.1940	.0360	0.0000	.0360
144	13	.3040	.2590	.0450	0.0000	.0450
169	14	.3940	.3380	.0560	0.0000	.0560
196	15	.5050	.4350	.0700	0.0000	.0700
225	16	.6360	.5520	.0840	0.0000	.0840
256	17	.7880	.6890	.0990	0.0000	.0990
289	18	.9670	.8490	.1180	0.0000	.1180
324	19	1.1760	1.0380	.1380	0.0000	.1380
361	20	1.4180	1.2560	.1620	0.0000	.1620
400	21	1.6900	1.5040	.1860	0.0000	.1860
441	22	2.0020	1.7880	.2140	0.0000	.2140
484	23	2.3530	2.1100	.2430	0.0000	.2430
529	24	2.7490	2.4730	.2760	0.0000	.2760
576	25	3.1970	2.8850	.3120	0.0000	.3120
625	26	3.6910	3.3400	.3510	0.0000	.3510
676	27	4.2440	3.8510	.3930	0.0000	.3930
729	28	4.8550	4.4160	.4390	0.0000	.4390
784	29	5.5250	5.0370	.4880	0.0000	.4880
841	30	6.2630	5.7250	.5380	0.0000	.5380
900	31	7.0770	6.4820	.5950	0.0000	.5950
961	32	7.9610	7.3080	.6530	0.0000	.6530
1024	33	8.9300	8.2140	.7160	0.0000	.7160

TABLE 4B (continued)

CSKYVBD TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	4	.0060	.0040	.0020	0.0000	.0020
16	5	.0110	.0080	.0030	0.0000	.0030
25	6	.0210	.0160	.0050	0.0000	.0050
36	7	.0350	.0280	.0070	0.0000	.0070
49	8	.0540	.0440	.0100	0.0000	.0100
64	9	.0830	.0690	.0140	0.0000	.0140
81	10	.1200	.1010	.0190	0.0000	.0190
100	11	.1680	.1430	.0250	0.0000	.0250
121	12	.2270	.1950	.0320	0.0000	.0320
144	13	.3020	.2620	.0400	0.0000	.0400
169	14	.4010	.3490	.0520	0.0000	.0520
196	15	.5050	.4430	.0620	0.0000	.0620
225	16	.6340	.5600	.0740	0.0000	.0740
256	17	.7900	.7010	.0890	0.0000	.0890
289	18	.9720	.8670	.1050	0.0000	.1050
324	19	1.1830	1.0590	.1240	0.0000	.1240
361	20	1.4270	1.2830	.1440	0.0000	.1440
400	21	1.6980	1.5320	.1660	0.0000	.1660
441	22	2.0130	1.8210	.1920	0.0000	.1920
484	23	2.3690	2.1500	.2190	0.0000	.2190
529	24	2.7720	2.5250	.2470	0.0000	.2470
576	25	3.2190	2.9400	.2790	0.0000	.2790
625	26	3.7190	3.4040	.3150	0.0000	.3150
676	27	4.2770	3.9210	.3560	0.0000	.3560
729	28	4.8920	4.5000	.3920	0.0000	.3920
784	29	5.5650	5.1300	.4350	0.0000	.4350
841	30	6.3080	5.8270	.4810	0.0000	.4810
900	31	7.1240	6.5940	.5300	0.0000	.5300
961	32	8.0170	7.4340	.5830	0.0000	.5830
1024	33	8.9850	8.3460	.6390	0.0000	.6390

TABLE 5
THE TIMES FOR THE LINEAR CORE STORAGE SUBROUTINES
FOR A MATRIX WITH ACTIVE COLUMNS

TABLE 5 – SOLUTION OF $G_{N^2}X = F$

CSKYJEN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	9	.0080	.0040	.0040	.0020	.0020
16	16	.0130	.0070	.0060	.0020	.0040
25	25	.0220	.0100	.0120	.0040	.0080
36	36	.0340	.0150	.0190	.0040	.0150
49	49	.0510	.0210	.0300	.0060	.0240
64	64	.0740	.0280	.0460	.0080	.0380
81	81	.1040	.0350	.0690	.0100	.0590
100	100	.1410	.0430	.0980	.0130	.0850
121	121	.1900	.0510	.1390	.0160	.1230
144	144	.2480	.0610	.1870	.0190	.1680
169	169	.3230	.0710	.2520	.0220	.2300
196	196	.4140	.0840	.3300	.0250	.3050
225	225	.4970	.0910	.4060	.0280	.3780
256	256	.6170	.1030	.5140	.0330	.3780
289	289	.7600	.1160	.6440	.0350	.6090
324	324	.9310	.1300	.8010	.0390	.7620
361	361	1.1310	.1450	.9860	.0440	.9420
400	400	1.3670	.1620	1.2050	.0480	1.1570
441	441	1.6320	.1780	1.4540	.0530	1.4010
484	484	1.9950	.1950	1.8000	.0610	1.7390
529	529	2.2880	.2150	2.0730	.0640	2.0090
576	576	2.7140	.2310	2.4830	.0680	2.4150
625	625	3.1150	.2500	2.8650	.0740	2.7910
676	676	3.5840	.2700	3.3140	.0800	3.2340
729	729	4.1350	.2910	3.8440	.0850	3.7590
784	784	4.7360	.3100	4.4260	.0940	4.3320
841	841	5.5520	.3340	5.2180	.0980	5.1200
900	900	6.1980	.3620	5.8360	.1060	5.7300
961	961	7.0970	.3840	6.7130	.1160	6.5970
1024	1024	8.4150	.4250	7.9900	.1270	7.8630

TABLE 5 (continued)

AN EXAMPLE WHEN CSKYJEN IS FASTER THAN CSKYLIN

CSKYJEN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	9	.0080	.0050	.0030	.0010	.0020
16	16	.0120	.0060	.0060	.0030	.0030
25	25	.0200	.0100	.0100	.0030	.0070
36	36	.0300	.0140	.0160	.0040	.0120
49	49	.0470	.0200	.0270	.0070	.0200
64	64	.0660	.0260	.0400	.0070	.0330
81	81	.0940	.0330	.0610	.0110	.0500
100	100	.1260	.0400	.0860	.0130	.0730
121	121	.1690	.0500	.1190	.0150	.1040
144	144	.2200	.0580	.1620	.0170	.1450
169	169	.2860	.0690	.2170	.0210	.1960
196	196	.3620	.0800	.2820	.0230	.2590
225	225	.4560	.0910	.3650	.0270	.3380
256	256	.5670	.1030	.4640	.0300	.4340

CSKYLIN TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	9	.0100	.0060	.0040	.0020	.0020
16	16	.0280	.0190	.0090	.0040	.0050
25	25	.0760	.0610	.0150	.0070	.0080
36	36	.1880	.1610	.0270	.0140	.0130
49	49	.4250	.3790	.0460	.0220	.0240
64	64	.8820	.8060	.0760	.0370	.0390
81	81	1.7010	1.5830	.1180	.0580	.0600
100	100	3.0920	2.9150	.1770	.0870	.0900
121	121	5.3330	5.0800	.2530	.1250	.1280
144	144	8.8160	8.4620	.3540	.1750	.1790
169	169	14.0310	13.5480	.4830	.2380	.2450
196	196	21.6170	20.9720	.6450	.3180	.3270
225	225	32.3590	31.5130	.8460	.4180	.4280
256	256	47.2900	46.1990	1.0910	.5390	.5520

TABLE 5 (continued)

SYMVBAJ TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	9	.0070	.0050	.0020	0.0000	.0020
16	16	.0120	.0080	.0040	0.0000	.0040
25	25	.0170	.0130	.0040	0.0000	.0040
36	36	.0260	.0190	.0070	0.0000	.0070
49	49	.0330	.0260	.0070	0.0000	.0070
64	64	.0440	.0340	.0100	0.0000	.0100
81	81	.0550	.0440	.0110	0.0000	.0110
100	100	.0690	.0540	.0150	0.0000	.0150
121	121	.0840	.0660	.0180	0.0000	.0180
144	144	.0980	.0770	.0210	0.0000	.0210
169	169	.1150	.0900	.0250	0.0000	.0250
196	196	.1350	.1060	.0290	0.0000	.0290
225	225	.1530	.1200	.0330	0.0000	.0330
256	256	.1760	.1370	.0390	0.0000	.0390
289	289	.1980	.1550	.0430	0.0000	.0430
324	324	.2230	.1740	.0490	0.0000	.0490
361	361	.2480	.1930	.0550	0.0000	.0550
400	400	.2750	.2140	.0610	0.0000	.0610
441	441	.3020	.2360	.0660	0.0000	.0660
484	484	.3320	.2590	.0730	0.0000	.0730
529	529	.3620	.2840	.0780	0.0000	.0780
576	576	.3940	.3080	.0860	0.0000	.0860
625	625	.4260	.3330	.0930	0.0000	.0930
676	676	.4570	.3580	.0990	0.0000	.0990
729	729	.4940	.3860	.1080	0.0000	.1080
784	784	.5300	.4150	.1150	0.0000	.1150
841	841	.5680	.4440	.1240	0.0000	.1240
900	900	.6060	.4740	.1320	0.0000	.1320
961	961	.6460	.5050	.1410	0.0000	.1410
1024	1024	.6870	.5380	.1490	0.0000	.1490

TABLE 5 (continued)

CSKYVBD TIMES

Order	Band	Total	Decomp.	Backsub.	F. Pass	B. Pass
9	9	.0080	.0060	.0020	0.0000	.0020
16	16	.0100	.0070	.0030	0.0000	.0030
25	25	.0150	.0110	.0040	0.0000	.0040
36	36	.0210	.0160	.0050	0.0000	.0050
49	49	.0300	.0230	.0070	0.0000	.0070
64	64	.0390	.0290	.0100	0.0000	.0100
81	81	.0480	.0370	.0110	0.0000	.0110
100	100	.0580	.0450	.0130	0.0000	.0130
121	121	.0710	.0550	.0160	0.0000	.0160
144	144	.0850	.0650	.0200	0.0000	.0200
169	169	.0990	.0770	.0220	0.0000	.0220
196	196	.1160	.0880	.0280	0.0000	.0280
225	225	.1330	.1020	.0310	0.0000	.0310
256	256	.1500	.1160	.0340	0.0000	.0340
289	289	.1700	.1310	.0390	0.0000	.0390
324	324	.1920	.1480	.0440	0.0000	.0440
361	361	.2130	.1640	.0490	0.0000	.0490
400	400	.2370	.1830	.0540	0.0000	.0540
441	441	.2610	.2010	.0600	0.0000	.0600
484	484	.2860	.2200	.0660	0.0000	.0660
529	529	.3110	.2400	.0710	0.0000	.0710
576	576	.3400	.2610	.0790	0.0000	.0790
625	625	.3650	.2810	.0840	0.0000	.0840
676	676	.3930	.3020	.0910	0.0000	.0910
729	729	.4240	.3260	.0980	0.0000	.0980
784	784	.4540	.3490	.1050	0.0000	.1050
841	841	.4870	.3740	.1130	0.0000	.1130
900	900	.5190	.3980	.1210	0.0000	.1210
961	961	.5690	.4370	.1320	0.0000	.1320
1024	1024	.5900	.4540	.1360	0.0000	.1360

CONCLUSIONS

The following conclusions were drawn from this study:

- In general there seems to be considerable advantage to consolidating the forward pass of the back-substitution with the Cholesky row decomposition.

- Jennings' coding device of concurrently accumulating the dot-products of the backward pass of the back-substitution seems to be slower than the direct coding of this procedure (at least with reference to the third linear core storage scheme).

- The usual scheme appears to be best for small full matrices (if the core storage can be spared). Otherwise, the modified linear scheme is almost as good.

- For large sparse matrices, particularly where the bandwidth MB is much smaller than the order N, the modified first linear scheme subroutine CSKYLIN is undoubtedly the best scheme of those considered. For banded matrices where the band is relatively compact and not too wide, the band scheme subroutine CSYBD1 does not fall too far behind CSKYLIN. However, for sparse positive definite matrices with "active columns" (as in the second example), CSKYVBD is definitely superior to CSKYLIN in both time and core storage requirements.

ACKNOWLEDGMENTS

The author wishes to acknowledge his indebtedness to Dr. Elizabeth H. Cuthill (1805) for her suggestions concerning this work. He also wishes to thank Mrs. Susan Voigt (1892), Mr. Michael Golden, and Mr. James McKee (1844) for their help.

REFERENCES

1. "The NASTRAN Theoretical Manual," R.H. MacNeal, editor, National Aeronautics and Space Administration, Washington, D.C., 1969.

2. Wilkinson, J.H., "The Algebraic Eigenvalue Problem," Clarendon Press, Oxford, 1965, p. 229-32.

3. Jennings, A., "Algorithm 70 Solution of Variable Bandwidth Positive Definite Simultaneous Equations," The Computer Journal, Vol. 14, p. 446, 1971.

4. Jennings, A., "A Compact Storage Scheme for the Solution of Symmetric Linear Simultaneous Equations," The Computer Journal, Vol. 9, pp. 281-285, 1966.

5. Westlake, J.R., "A Handbook of Numerical Matrix Inversion and Solution of Linear Equations," p. 140, no. 15, John Wiley, New York, 1968.

6. Cuthill, E. and McKee, J., "Reducing the Bandwidth of Sparse Symmetric Matrices," Proceedings 24th National Conference of the ACM, pp. 157-172, 1969.

INITIAL DISTRIBUTION

Copies	
2	ONR 1 Dr. R. Lundegard (607) 1 Dr. L. D. Bram (607)
2	USNA 1 Dept of Math 1 Tech Lib
2	NAVPGSCHOL 1 Lib, Tech Rep Sec 1 Math Dept
1	NROTC
1	NAVWARCOL
1	NAVSHIPYD BREM
1	NAVSHIPYD BSN
1	NAVSHIPYD CHASN
1	NAVSHIPYD HUNTERS PT
1	NAVSHIPYD LBEACH
1	NAVSHIPYD MARE ISLAND
1	NAVSHIPYD NORVA
1	NAVSHIPYD PEARL
1	NAVSHIPYD PHILA
1	NAVSHIPYD PTSMH
1	WPAFB, AR/LB 1 Dr. Paul Nikolai
12	DDC

Copies	
1	NBS, Inst for Basic Standards Dr. Morris Newman Code 205.01
1	NIMH Dr. R. Rawlings (LL06)
1	Lockheed Missiles & Space Co., Electronic Sci Lab, Dept 52-40 Sunnyvale, Calif 94086 Mr. Eldon Hansen

CENTER DISTRIBUTION

Copies:	
1	01
1	18-1808-1809
1	1802.1
1	1802.2
1	1802.3
1	1802.4
1	1805
1	183
2	1835 (T. Walton, M. Culpepper)
1	184
1	1842 (K. Meals)
1	1843 (J. Schot)
1	1844 (S. Dhir)
13	1844 1 (P. Matula) 1 (F. Henderson) 1 (G.C. Everstine) 10 (D. Gignac)
1	185
1	186
1	1864 (R.P. Eddy)
1	188
1	189
1	189.1 (N. Taylor)

Copies:	
2	1892.1 1 (S. Good) 1 (J. Strickland)
1	1892.3 (S. Voigt)
1	1725 (N. Gifford)
1	1966 (Dr. Y. Liu)

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Ship Research & Development Center Bethesda, Maryland 20034	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3 REPORT TITLE

A COMPARATIVE STUDY OF SEVERAL CORE STORAGE SCHEMES FOR LARGE SPARSE POSITIVE DEFINITE MATRICES WITH REFERENCE TO THE CHOLESKY ALGORITHM

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

5 AUTHOR(S) (First name, middle initial, last name)

Donald A. Gignac

6 REPORT DATE	7a. TOTAL NO OF PAGES	7b. NO. OF REFS
November 1972	38	6

8a. CONTRACT OR GRANT NO b. PROJECT NO. SR 014 03 01, Task 15322 c. 1-1844-004-05 d.	9a. ORIGINATOR'S REPORT NUMBER(S) 4017 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
	NAVSHIPS (Code 0311)

13 ABSTRACT

In the finite element approach to static structural analysis, the solution of the equation

$$KU = P$$

a positive definite system of simultaneous linear equations, is basic. Considerable difficulty may be experienced when K is very large and sparse. This report documents an investigation of several FORTRAN subroutines in order to obtain an efficient Cholesky algorithm subroutine with economical core storage for an in-core solution of KU = P for large sparse K.

The following conclusions were drawn from this study:

- In general there seems to be considerable advantage to consolidating the forward pass of the back-substitution with the Cholesky row decomposition.
- For large sparse matrices, particularly where the bandwidth MB is much smaller than the order N, the modified first linear scheme subroutine CSKYLIN is undoubtedly the best subroutine of those considered. For banded matrices where the band is relatively compact and not too wide, the band scheme subroutine CSYBD1 does not fall too far behind CSKYLIN. However, for sparse positive definite matrices with "active columns" (as in the second example), CSKYVBD is definitely superior to CSKYLIN in both time and core storage requirements.

DD FORM 1473 1 NOV 65 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Sparse Matrices Cholesky Algorithm Solution of Sparse Systems Compact Linear Core Storage of Sparse Positive Definite Matrices						