CHAPTER 1

APPLIED LINEAR ALGEBRA

1.1 FOUR SPECIAL MATRICES

An m by n matrix has m rows and n columns and mn entries. We operate on those rows and columns to solve linear systems $Ax = b$ and eigenvalue problems $Ax = \lambda x$. From inputs A and b (and from software like MATLAB) we get outputs x and λ. A fast stable algorithm is extremely important, and this book includes fast algorithms.

One purpose of matrices is to store information, but another viewpoint is more important for applied mathematics. Often we see the matrix as an "operator." ***A acts on vectors x to produce Ax***. The components of x have a meaning—displacements or pressures or voltages or prices or concentrations. The operator A also has a meaning—in this chapter A takes differences. Then Ax represents pressure differences or voltage drops or price differentials.

Before we turn the problem over to the machine—and also after, when we interpret `A\b` or `eig(A)`—it is the meaning we want, as well as the numbers.

This book begins with **four special families of matrices**—simple and useful, absolutely basic. We look first at the properties of these particular matrices K_n, C_n, T_n, and B_n. (Some properties are obvious, others are hidden.) It is terrific to practice linear algebra by working with genuinely important matrices. Here are K_2, K_3, K_4 in the first family, with -1 and 2 and -1 down the diagonals:

$$K_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad K_3 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad K_4 = \begin{bmatrix} \mathbf{2} & \mathbf{-1} & 0 & 0 \\ \mathbf{-1} & \mathbf{2} & \mathbf{-1} & 0 \\ 0 & \mathbf{-1} & \mathbf{2} & \mathbf{-1} \\ 0 & 0 & \mathbf{-1} & \mathbf{2} \end{bmatrix}$$

What is significant about K_2 and K_3 and K_4, and eventually the n by n matrix K_n? I will give six answers in the same order that my class gave them—starting with four properties of the K's that you can see immediately.

1. These matrices are **symmetric**. The entry in row i, column j also appears in row j, column i. Thus $K_{ij} = K_{ji}$, on opposite sides of the main diagonal. Symmetry can be expressed by transposing the whole matrix at once: $\boldsymbol{K = K^{\mathrm{T}}}$.

2. The matrices K_n are **sparse**. Most of their entries are zero when n gets large. K_{1000} has a million entries, but only $1000 + 999 + 999$ are nonzero.

3. The nonzeros lie in a "band" around the main diagonal, so each K_n is *banded*. The band has only three diagonals, so these matrices are **tridiagonal**.

Because K is a tridiagonal matrix, $Ku = f$ can be quickly solved. If the unknown vector u has a thousand components, we can find them in a few thousand steps (which take a small fraction of a second). For a full matrix of order $n = 1000$, solving $Ku = f$ would take hundreds of millions of steps. Of course we have to ask if the linear equations have a solution in the first place. That question is coming soon.

4. The matrices have **constant diagonals**. Right away that property wakes up Fourier. It signifies that something is not changing when we move in space or time. The problem is shift-invariant or time-invariant. Coefficients are constant. The tridiagonal matrix is entirely determined by the three numbers $-1, 2, -1$. These are actually "second difference matrices" but my class never says that.

The whole world of Fourier transforms is linked to constant-diagonal matrices. In signal processing, the matrix $D = K/4$ is a "highpass filter." Du picks out the rapidly varying (high frequency) part of a vector u. It gives a *convolution* with $\frac{1}{4}(-1, 2, -1)$. We use these words to call attention to the Fourier part (Chapter 4) of this book.

Mathematicians call K a ***Toeplitz matrix***, and MATLAB uses that name:

The command $K = \mathsf{toeplitz}([\,2\ \ -1\ \ \mathsf{zeros}(1,2)\,])$ **constructs** K_4 **from row 1.**

Actually, Fourier will be happier if we make two small changes in K_n. Insert -1 in the southwest and northeast corners. This completes two diagonals (which circle around). All four diagonals of C_4 wrap around in this "*periodic matrix*" or "*cyclic convolution*" or **circulant matrix**:

Circulant matrix
$$C_4 = \begin{bmatrix} 2 & -1 & 0 & \mathbf{-1} \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ \mathbf{-1} & 0 & -1 & 2 \end{bmatrix} = \mathsf{toeplitz}([\,2\ \ -1\ \ 0\ \ -1\,])\,.$$

This matrix is *singular*. It is *not invertible*. Its determinant is zero. Rather than computing that determinant, it is much better to identify a nonzero vector u that solves $C_4u = 0$. (**If** C_4 **had an inverse, the only solution to** $C_4u = 0$ **would be the zero vector**. We could multiply by C_4^{-1} to find $u = 0$.) For this matrix, the column vector u of all ones (printed as $u = (1, 1, 1, 1)$ with commas) solves $C_4u = 0$.

The columns of C add to the zero column. This vector $u = \mathsf{ones}(4, 1)$ is in the *nullspace* of C_4. The nullspace contains all solutions to $Cu = 0$.

Whenever the entries along every row of a matrix add to zero, the matrix is certainly singular. The same all-ones vector u is responsible. Matrix multiplication Cu adds the column vectors and produces zero. The constant vector $u = (1, 1, 1, 1)$ or $u = (c, c, c, c)$ in the nullspace is like the constant C when we integrate a function. In calculus, this "arbitrary constant" is not knowable from the derivative. In linear algebra, the constant in $u = (c, c, c, c)$ is not knowable from $Cu = 0$.

5. All the matrices $K = K_n$ are **invertible**. They are not singular, like C_n. There is a square matrix K^{-1} such that $K^{-1}K = I = \textit{identity matrix}$. And if a square matrix has an inverse on the left, then also $KK^{-1} = I$. This "*inverse matrix*" is also symmetric when K is symmetric. *But K^{-1} is not sparse.*

Invertibility is not easy to decide from a quick look at a matrix. Theoretically, one test is to compute the determinant. There is an inverse except when $\det K = 0$, because the formula for K^{-1} includes a division by $\det K$. But computing the determinant is almost never done in practice! It is a poor way to find $u = K^{-1}f$.

What we actually do is to go ahead with the elimination steps that solve $Ku = f$. Those steps simplify the matrix, to make it triangular. The nonzero pivots on the main diagonal of the triangular matrix show that the original K is invertible. (Important: We don't want or need K^{-1} to find $u = K^{-1}f$. The inverse would be a full matrix, with all positive entries. All we compute is the solution vector u.)

6. The symmetric matrices K_n are **positive definite**. Those words might be new. One goal of Chapter 1 is to explain what this crucial property means (K_4 has it, C_4 doesn't). Allow me to start by contrasting positive definiteness with invertibility, using the words "pivots" and "eigenvalues" that will soon be familiar. *Please notice the Appendix that summarizes linear algebra.*

(***Pivots***) An invertible matrix has n *nonzero* pivots.
A *positive definite* symmetric matrix has n ***positive*** **pivots**.

(***Eigenvalues***) An invertible matrix has n *nonzero* eigenvalues.
A *positive definite* symmetric matrix has n ***positive*** **eigenvalues**.

Positive pivots and eigenvalues are tests for positive definiteness, and C_4 fails those tests because it is singular. Actually C_4 has three positive pivots and eigenvalues, so it almost passes. But its fourth eigenvalue is zero (the matrix is singular). Since no eigenvalue is negative ($\lambda \geq 0$), C_4 is **positive** *semi***definite**.

The pivots appear on the main diagonal in Section 1.3, when solving $Ku = f$ by elimination. The eigenvalues arise in $Kx = \lambda x$. There is also a determinant test for positive definiteness (not just $\det K > 0$). The proper definition of a symmetric positive definite matrix (it is connected to positive energy) will come in Section 1.6.

Changing K_n to T_n

After K_n and C_n, there are two more families of matrices that you need to know. They are symmetric and tridiagonal like the family K_n. But the $(1,1)$ entry in T_n is changed from 2 to 1:

$\mathbf{T_n(1,1) = 1}$ $\qquad T_2 = \begin{bmatrix} \mathbf{1} & -1 \\ -1 & 2 \end{bmatrix}$ and $T_3 = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$. (1)

That top row (T stands for top) represents a new boundary condition, whose meaning we will soon understand. Right now we use T_3 as a perfect example of elimination. Row operations produce zeros below the diagonal, and the pivots are circled as they are found. **Two elimination steps reduce T to the upper triangular U.**

Step 1. Add row 1 to row 2, which leaves zeros below the first pivot.

Step 2. Add the new row 2 to row 3, which produces U.

$$T = \begin{bmatrix} \textcircled{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \xrightarrow{\text{Step 1}} \begin{bmatrix} \textcircled{1} & -1 & 0 \\ 0 & \textcircled{1} & -1 \\ 0 & -1 & 2 \end{bmatrix} \xrightarrow{\text{Step 2}} \begin{bmatrix} \textcircled{1} & -1 & 0 \\ 0 & \textcircled{1} & -1 \\ 0 & 0 & \textcircled{1} \end{bmatrix} = U\,.$$

All three pivots of T equal 1. We can apply the test for invertibility (three nonzero pivots). T_3 also passes the test for positive definiteness (three *positive* pivots). In fact every T_n in this family is positive definite, with all its pivots equal to 1.

That matrix U has an inverse (which is automatically upper triangular). The exceptional fact for this particular U^{-1} is that *all upper triangular entries are* 1's:

$$U^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \mathsf{triu}(\mathsf{ones}(3)). \tag{2}$$

This says that the inverse of a 3 by 3 "difference matrix" is a 3 by 3 "**sum matrix**." This neat inverse of U will lead us to the inverse of T in Problem 2. **The product $U^{-1}U$ is the identity matrix I.** U takes differences, and U^{-1} takes sums. Taking differences and then sums will recover the original vector (u_1, u_2, u_3):

Differences from U
$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ u_3 - 0 \end{bmatrix}$$

Sums from U^{-1}
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ u_3 - 0 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$

Changing T_n to B_n

The fourth family B_n has the last entry also changed from 2 to 1. The new boundary condition is being applied at both ends (B stands for both). These matrices B_n are symmetric and tridiagonal, but you will quickly see that they are *not invertible*. The B_n are positive semidefinite but *not positive definite*:

$\mathbf{B_n(n,n) = 1}$ $\qquad B_2 = \begin{bmatrix} \mathbf{1} & -1 \\ -1 & \mathbf{1} \end{bmatrix}$ and $B_3 = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix}$. (3)

Again, elimination brings out the properties of the matrix. The first $n-1$ pivots will all equal 1, because those rows are not changed from T_n. But the change from 2 to 1 in the last entry of B produces a change from 1 to 0 in the last entry of U:

$$B = \begin{bmatrix} \textcircled{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \textcircled{1} & -1 & 0 \\ 0 & \textcircled{1} & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \textcircled{1} & -1 & 0 \\ 0 & \textcircled{1} & -1 \\ 0 & 0 & \mathbf{0} \end{bmatrix} = U. \quad (4)$$

There are only two pivots. (A pivot must be nonzero.) The last matrix U is certainly not invertible. Its determinant is zero, because its third row is all zeros. The constant vector $(1,1,1)$ is in the **nullspace** of U, and therefore it is in the nullspace of B:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and also} \quad \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The whole point of elimination was to simplify a linear system like $Bu = 0$, *without changing the solutions*. In this case we could have recognized non-invertibility in the matrix B, because each row adds to zero. Then the sum of its three columns is the zero column. This is what we see when B multiplies the vector $(1,1,1)$.

Let me summarize this section in four lines (all these matrices are symmetric):

K_n and T_n are invertible and (more than that) positive definite.

C_n and B_n are singular and (more than that) positive semidefinite.

The nullspaces of C_n and B_n contain all the constant vectors $u = (c, c, \ldots, c)$. Their columns are dependent.

The nullspaces of K_n and T_n contain only the zero vector $u = (0, 0, \ldots, 0)$. Their columns are independent.

Matrices in MATLAB

It is natural to choose MATLAB for linear algebra, but the reader may select another system. (Octave is very close, and free. Mathematica and Maple are good for symbolic calculation, LAPACK provides excellent codes at no cost in netlib, and there are many other linear algebra packages.) We will construct matrices and operate on them in the convenient language that MATLAB provides.

Our first step is to construct the matrices K_n. For $n = 3$, we can enter the 3 by 3 matrix a row at a time, inside brackets. Rows are separated by a semicolon ;

$$K = [\ 2\ \ -1\ \ 0\ ;\ -1\ \ 2\ \ -1\ ;\ 0\ \ -1\ \ 2\ ]$$

For large matrices this is too slow. We can build K_8 from "eye" and "ones":

$\mathsf{eye}(8) = 8$ by 8 identity matrix $\qquad \mathsf{ones}(7,1) =$ column vector of seven 1's

The diagonal part is 2∗eye(8). The symbol $*$ means multiplication! The -1's above the diagonal of K_8 have the vector $-\mathsf{ones}(7,1)$ along diagonal 1 of the matrix E:

Superdiagonal of −1's $\qquad E = -\mathsf{diag}(\mathsf{ones}(7,1),1)$

The -1's *below* the diagonal of K_8 lie on the diagonal numbered -1. For those we could change the last argument in E from 1 to -1. Or we can simply transpose E, using the all-important symbol E' for E^{T}. Then K comes from its three diagonals:

Tridiagonal matrix K_8 $\qquad K = 2 * \mathsf{eye}(8) + E + E'$

Note: The zeroth diagonal (main diagonal) is the default with no second argument, so eye(8)= diag(ones(8,1)). And then $\mathsf{diag}(\mathsf{eye}(8)) = \mathsf{ones}(8,1)$.

The constant diagonals make K *a Toeplitz matrix.* The toeplitz command produces K, when each diagonal is determined by a single number 2 or -1 or 0. Use the zeros vector for the 6 zeros in the first row of K_8:

Symmetric Toeplitz $\quad \mathrm{row1} = [\,2\ \ -1\ \ \mathsf{zeros}(1,6)\,];\ \ K = \mathsf{toeplitz}(\mathrm{row1})$

For an unsymmetric constant-diagonal matrix, use toeplitz(col1, row1). Taking col1 = [1 −1 0 0] and row1 = [1 0 0] gives a 4 by 3 backward difference matrix. It has two nonzero diagonals, 1's and -1's.

To construct the matrices T and B and C from K, just change entries as in the last three lines of this M-file that we have named KTBC.m. Its input is the size n, its output is four matrices of that size. The semicolons suppress display of K, T, B, C:

```
function [K,T,B,C] = KTBC(n)
% Create the four special matrices assuming n>1
K = toeplitz ([2 -1 zeros(1,n-2)]);
T = K; T(1,1) = 1;
B = K; B(1,1) = 1; B(n,n) = 1;
C = K; C(1,n) = -1; C(n,1) = -1;
```

If we happened to want their determinants (we shouldn't!), then with $n = 8$

$$\begin{bmatrix} \det(K) & \det(T) & \det(B) & \det(C) \end{bmatrix} \quad \text{produces the output} \quad \begin{bmatrix} 9 & 1 & 0 & 0 \end{bmatrix}$$

One more point. MATLAB could not store K_n as a dense matrix for $n = 10{,}000$. **The 10^8 entries need about 800 megabytes unless we recognize K as sparse**. The code sparseKTBC.m on the course website avoids storing (and operating on) all the zeros. It has K, T, B, or C and n as its first two arguments. The third argument is 1 for sparse, 0 for dense (default 0 for narg = 2, no third argument).

The input to Sparse MATLAB includes the locations of all nonzero entries. The command $A =$ sparse(i, j, s, m, n) creates an m by n sparse matrix from the vectors i, j, s that list all positions i, j of nonzero entries s. Elimination by lu(A) may produce additional nonzeros (called fill-in) which the software will correctly identify. In the normal "full" option, zeros are processed like all other numbers.

It is best to create the list of triplets i, j, s and then call sparse. Insertions $A(i, j) = s$ or $A(i, j) = A(i, j) + s$ are more expensive. We return to this point in Section 3.6.

The sparse KTBC code on the website uses spdiags to enter the three diagonals. Here is the toeplitz way to form K_8, all made sparse by its sparse vector start:

```
vsp = sparse([2 -1 zeros(1, 6)]) % please look at each output
Ksp = toeplitz(vsp) % sparse format gives the nonzero positions and entries
bsp = Ksp(:, 2)     % colon keeps all rows of column 2, so bsp = column 2 of Ksp
usp = Ksp\bsp       % zeros in Ksp and bsp are not processed, solution: usp(2) = 1
uuu = full(usp)     % return from sparse format to the full uuu = [0 1 0 0 0 0 0 0]
```

Note The open source language Python is also very attractive and convenient.

The next sections will use all four matrices in these basic tasks of linear algebra:

(**1.2**) The finite difference matrices K, T, B, C include boundary conditions

(**1.3**) Elimination produces pivots in D and triangular factors in LDL^{T}

(**1.4**) Point loads produce inverse matrices K^{-1} and T^{-1}

(**1.5**) The eigenvalues and eigenvectors of K, T, B, C involve sines and cosines.

You will see K\f in **1.2**, lu(K) in **1.3**, inv(K) in **1.4**, eig(K) in **1.5**, and chol(K) in **1.6**.

I very much hope that you will come to know and like these special matrices.

■ WORKED EXAMPLES ■

1.1 A $Bu = f$ and $Cu = f$ might be solvable even though B and C are singular!

Show that every vector $f = Bu$ has $f_1 + f_2 + \cdots + f_n = 0$. Physical meaning: **the external forces balance**. Linear algebra meaning: $Bu = f$ is solvable when f is perpendicular to the all-ones column vector $e = (1, 1, 1, 1, \dots) = \mathsf{ones}(n, 1)$.

Solution Bu is a vector of "differences" of u's. Those differences always add to zero:

$$f = Bu = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_1 - u_2 \\ -u_1 + 2u_2 - u_3 \\ -u_2 + 2u_3 - u_4 \\ -u_3 + u_4 \end{bmatrix}$$

All terms cancel in $(u_1 - u_2) + (-u_1 + 2u_2 - u_3) + (-u_2 + 2u_3 - u_4) + (-u_3 + u_4) = 0$. The dot product with $e = (1, 1, 1, 1)$ is that sum $f^{\mathrm{T}}e = f_1 + f_2 + f_3 + f_4 = 0$.

Dot product $f \cdot e = f^{\mathrm{T}}e = f_1e_1 + f_2e_2 + f_3e_3 + f_4e_4$ (f ′ ∗ e in MATLAB).

A second explanation for $f^{\mathrm{T}}e = 0$ starts from the fact that $Be = 0$. The all-ones vector e is in the nullspace of B. Transposing $f = Bu$ gives $f^{\mathrm{T}} = u^{\mathrm{T}}B^{\mathrm{T}}$, since **the transpose of a product has the individual transposes in reverse order**. This matrix B is symmetric so $B^{\mathrm{T}} = B$. Then

$$f^{\mathrm{T}}e = u^{\mathrm{T}}B^{\mathrm{T}}e \quad \text{is equal to} \quad u^{\mathrm{T}}Be = u^{\mathrm{T}}0 = 0.$$

Conclusion $Bu = f$ is only solvable when f is perpendicular to the all-ones vector e. (*The same is true for* $Cu = f$. Again the differences cancel out.) **The external forces balance** when the f's add to zero. The command B\f will produce Inf because B is square and singular, but the "pseudoinverse" u = pinv(B) ∗ f will succeed. (Or add a zero row to B and f before the command B\f, to make the system rectangular.)

1.1 B The "fixed-free" matrix H changes the *last entry of* K from 2 to 1. Connect H to the "free-fixed" T (*first entry* $= 1$) by using the reverse identity matrix J:

$$H = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix} \quad \begin{matrix} \text{comes from} \\ JTJ \text{ via the} \\ \textit{reverse identity } J \end{matrix} \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Chapter 2 shows how T comes from a *tower* structure (free at the top). H comes from a *hanging* structure (free at the bottom). Two MATLAB constructions are

$H =$ toeplitz([2 −1 0]); $H(3,3) = 1$ **or** $J =$ fliplr(eye(3)); $H = J * T * J$

Solution JT reverses the rows of T. Then JTJ reverses the columns to give H:

$$T = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad \begin{matrix} JT = \\ (\text{rows}) \end{matrix} \begin{bmatrix} 0 & -1 & 2 \\ -1 & 2 & -1 \\ \mathbf{1} & -1 & 0 \end{bmatrix} \qquad \begin{matrix} (JT)J = H\,. \\ (\textit{columns too}) \end{matrix}$$

We could reverse columns first by TJ. Then $J(TJ)$ would be the same matrix H as $(JT)J$. **The parentheses never matter in** $(AB)C = A(BC)$!

Any permutation matrix like J has the rows of the identity matrix I in some order. There are six 3 by 3 permutation matrices because there are six orders for the numbers $1, 2, 3$. *The inverse of every permutation matrix is its transpose.* This particular J is symmetric, so it has $J = J^{\mathrm{T}} = J^{-1}$ as you can easily check:

$$H = JTJ \quad \text{so} \quad H^{-1} = J^{-1}T^{-1}J^{-1} \quad \text{which is} \quad H^{-1} = JT^{-1}J. \tag{5}$$

With back = 3:−1:1, reordering to JTJ is H = T(back, back) in MATLAB.

Problem Set 1.1

Problems 1–4 are about T^{-1} and Problems 5–8 are about K^{-1}.

1 The inverses of T_3 and T_4 (with $T_{11} = 1$ in the top corner) are

$$T_3^{-1} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad T_4^{-1} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Guess T_5^{-1} and multiply by T_5. Find a simple formula for the entries of T_n^{-1} on and below the diagonal ($i \geq j$), and then on and above the diagonal ($i \leq j$).

2 Compute T_3^{-1} in three steps, using U and U^{-1} in equation (2):

1. Check that $T_3 = U^{\mathrm{T}}U$, where U has 1's on the main diagonal and -1's along the diagonal above. Its transpose U^{T} is lower triangular.
2. Check that $UU^{-1} = I$ when U^{-1} has 1's on and above the main diagonal.
3. Invert $U^{\mathrm{T}}U$ to find $T_3^{-1} = (U^{-1})(U^{-1})^{\mathrm{T}}$. *Inverses come in reverse order*!

3 The difference matrix $U = U_5$ in MATLAB is eye(5)−diag(ones(4,1),1). Construct the sum matrix S from triu(ones(5)). (This keeps the upper triangular part of the 5 by 5 all-ones matrix.) Multiply $U * S$ to verify that $S = U^{-1}$.

4 For every n, $S_n = {U_n}^{-1}$ is upper triangular with ones on and above the diagonal. For $n = 4$ check that SS^{T} produces the matrix ${T_4}^{-1}$ predicted in Problem 1. Why is SS^{T} certain to be a symmetric matrix?

5 The inverses of K_3 and K_4 (please also invert K_2) have fractions $\frac{1}{\det} = \frac{1}{4}, \frac{1}{5}$:

$$K_3^{-1} = \frac{1}{4}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad \text{and} \quad K_4^{-1} = \frac{1}{5}\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

First *guess* the determinant of $K = K_5$. Then compute $\mathsf{det}(K)$ and $\mathsf{inv}(K)$ and $\mathsf{det}(K) * \mathsf{inv}(K)$—any software is allowed.

6 (Challenge problem) Find a *formula* for the i, j entry of K_4^{-1} *below the diagonal* ($i \geq j$). Those entries grow linearly along every row and up every column. (Section 1.4 will come back to these important inverses.) Problem 7 below is developed in the Worked Example of Section 1.4.

7 **A column u times a row v^{T} is a rank-one matrix** uv^{T}. All columns are multiples of u, and all rows are multiples of v^{T}. ${T_4}^{-1} - {K_4}^{-1}$ has rank 1:

$$T_4^{-1} - K_4^{-1} = \frac{1}{5}\begin{bmatrix} 16 & 12 & 8 & 4 \\ 12 & 9 & 6 & 3 \\ 8 & 6 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}$$

Write $K_3 - T_3$ in this special form uv^{T}. Predict a similar formula for $T_3^{-1} - K_3^{-1}$.

8 (a) Based on Problem 7, predict the i, j entry of $T_5^{-1} - K_5^{-1}$ below the diagonal.

(b) Subtract this from your answer to Problem 1 (the formula for T_5^{-1} when $i \geq j$). This gives the not-so-simple formula for K_5^{-1}.

9 Following Example **1.1 A** with C instead of B, show that $e = (1, 1, 1, 1)$ is perpendicular to each column of C_4. Solve $Cu = f = (1, -1, 1, -1)$ with the singular matrix C by $\mathsf{u = pinv(C) * f}$. Try $\mathsf{u = C\backslash e}$ and $\mathsf{C\backslash f}$, before and after adding a fifth equation $0 = 0$.

10 The "hanging matrix" H in Worked Example **1.1 B** changes the last entry of K_3 to $H_{33} = 1$. Find the inverse matrix from $H^{-1} = JT^{-1}J$. Find the inverse also from $H = UU^{\mathrm{T}}$ (check upper times lower triangular!) and $H^{-1} = (U^{-1})^{\mathrm{T}}U^{-1}$.

11 Suppose U is any upper triangular matrix and J is the reverse identity matrix in **1.1 B**. Then JU is a "southeast matrix". What geographies are UJ and JUJ? By experiment, a southeast matrix times a northwest matrix is ____.

12 Carry out elimination on the 4 by 4 circulant matrix C_4 to reach an upper triangular U (or try $[L, U] = \mathsf{lu}(C)$ in MATLAB). Two points to notice: The last entry of U is ____ because C is singular. The last column of U has new nonzeros. Explain why this "fill-in" happens.

13 By hand, can you factor the circulant C_4 (with three nonzero diagonals, allowing wraparound) into circulants L times U (with two nonzero diagonals, allowing wraparound so not truly triangular)?

14 Gradually reduce the diagonal $2, 2, 2$ in the matrix K_3 until you reach a singular matrix M. This happens when the diagonal entries reach ____. Check the determinant as you go, and find a nonzero vector that solves $Mu = 0$.

Questions 15–21 bring out important facts about matrix multiplication.

15 How many individual multiplications to create Ax and A^2 and AB?

$$A_{n\times n}\, x_{n\times 1} \qquad A_{n\times n}\, A_{n\times n} \qquad A_{m\times n}\, B_{n\times p} = (AB)_{m\times p}$$

16 You can multiply Ax by rows (the usual way) or **by columns** (more important). Do this multiplication both ways:

$$\textbf{By rows} \quad \begin{bmatrix}2 & 3\\ 4 & 5\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix} = \begin{bmatrix}\text{inner product using row 1}\\ \text{inner product using row 2}\end{bmatrix}$$

$$\textbf{By columns} \quad \begin{bmatrix}2 & 3\\ 4 & 5\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix} = 1\begin{bmatrix}2\\ 4\end{bmatrix} + 2\begin{bmatrix}3\\ 5\end{bmatrix} = \begin{bmatrix}\textbf{combination}\\ \textbf{of columns}\end{bmatrix}$$

17 The product Ax is a **linear combination of the columns of** A. The equations $Ax = b$ have a solution vector x exactly when b is a ____ of the columns.

Give an example in which b is *not in the column space* of A. There is no solution to $Ax = b$, because b is not a combination of the columns of A.

18 Compute $C = AB$ by multiplying the matrix A times each column of B:

$$\begin{bmatrix}2 & 3\\ 4 & 5\end{bmatrix}\begin{bmatrix}1 & 2\\ 2 & 4\end{bmatrix} = \begin{bmatrix}8 & *\\ 14 & *\end{bmatrix}.$$

Thus, $\mathsf{A} * \mathsf{B}(:,\mathsf{j}) = \mathsf{C}(:,\mathsf{j})$.

19 You can also compute AB by multiplying each row of A times B:

$$\begin{bmatrix}2 & 3\\ 4 & 5\end{bmatrix}\begin{bmatrix}1 & 2\\ 2 & 4\end{bmatrix} = \begin{bmatrix}2*\text{row } 1 + 3*\text{row } 2\\ 4*\text{row } 1 + 5*\text{row } 2\end{bmatrix} = \begin{bmatrix}8 & 16\\ * & *\end{bmatrix}.$$

A solution to $Bx = 0$ is also a solution to $(AB)x = 0$. Why? From

$$Bx = \begin{bmatrix}1 & 2\\ 2 & 4\end{bmatrix}\begin{bmatrix}-2\\ 1\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix} \quad \text{how do we know} \quad ABx = \begin{bmatrix}8 & 16\\ * & *\end{bmatrix}\begin{bmatrix}-2\\ 1\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}?$$

20 The four ways to find AB give numbers, columns, rows, and **matrices**:

1 ***(rows of A) times (columns of B)***	C(i,j) = A(i,:) * B(:,j)
2 ***A times (columns of B)***	C(:,j) = A * B(:,j)
3 ***(rows of A) times B***	C(i,:) = A(i,:) * B
4 **(columns of *A*) times (rows of *B*)**	for k = 1:n, C = C + A(:,k) * B(k,:); end

Finish these 8 multiplications for **columns times rows**. How many for n by n?

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 5 \end{bmatrix} \begin{bmatrix} 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix} + \begin{bmatrix} * & * \\ * & * \end{bmatrix} = \begin{bmatrix} 8 & * \\ * & * \end{bmatrix}.$$

21 Which *one* of these equations is true for all n by n matrices A and B?

$$AB = BA \qquad (AB)A = A(BA) \qquad (AB)B = B(BA) \qquad (AB)^2 = A^2B^2\,.$$

22 Use $n = 1000$; $e = \mathsf{ones}(n, 1)$; $K = \mathsf{spdiags}([-e, 2 * e, -e], -1\!:\!1, n, n)$; to enter K_{1000} as a sparse matrix. Solve the sparse equation $Ku = e$ by $\mathsf{u} = \mathsf{K}\backslash\mathsf{e}$. Plot the solution by $\mathsf{plot}(u)$.

23 Create 4-component vectors u, v, w and enter $A = \mathsf{spdiags}([u, v, w], -1\!:\!1, 4, 4)$. Which components of u and w are left out from the -1 and 1 diagonals of A?

24 Build the sparse identity matrix $I = \mathsf{sparse}(i, j, s, 100, 100)$ by creating vectors i, j, s of positions i, j with nonzero entries s. (You could use a for loop.) In this case $\mathsf{speye}(100)$ is quicker. Notice that $\mathsf{sparse}(\mathsf{eye}(10000))$ would be a disaster, since there isn't room to store $\mathsf{eye}(10000)$ before making it sparse.

25 The only solution to $Ku = 0$ or $Tu = 0$ is $u = 0$, so K and T are invertible. For proof, suppose u_i is the largest component of u. If $-u_{i-1} + 2u_i - u_{i+1}$ is zero, this forces $u_{i-1} = u_i = u_{i+1}$. Then the next equations force every $u_j = u_i$. At the end, when the boundary is reached, $-u_{n-1} + 2u_n$ only gives zero if $u = 0$.

Why does this "diagonally dominant" argument fail for B and C?

26 For which vectors v is $\mathsf{toeplitz}(v)$ a circulant matrix (cyclic diagonals) ?

27 (Important) Show that the 3 by 3 matrix K comes from $A_0^{\mathrm{T}} A_0$:

$$A_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \text{is a "}\textbf{difference matrix}\text{"}$$

Which column of A_0 would you remove to produce A_1 with $T = A_1^{\mathrm{T}} A_1$? Which column would you remove next to produce A_2 with $B = A_2^{\mathrm{T}} A_2$? The difference matrices A_0, A_1, A_2 have $0, 1, 2$ boundary conditions. So do the "second differences" K, T, and B.