CPS196.2, Chapter 5; Graphs and Matrices

1. Introduction

We study several matrices that arise naturally from graphs and digraphs. We show that some of these matrices represent (are bases for) important vector spaces associated with the graph or digraph. The resistive electrical network problem is presented as an important application, and the equations are derived.

2. Vertex Adjacency Matrix

Recall that, in Example 1 of Chapt3 we defined the vertex adjacency matrix of a graph, \( G = (X,E) \), with \( X \) ordered as \( X = \{1, 2, 3, \ldots, |X| = n\} \), as the \( n \times n \) matrix

\[
V(i,j) = \begin{cases} 
1 & \text{if } ij \in E; \\
0 & \text{otherwise.}
\end{cases}
\]

We can use the identical definition when \( D \) is a digraph. In the graph case, \( V \) is symmetric, \( V = V^T \), but not in the digraph case.

Example 1. Let \( G \), in Fig 1a, be the directed graph of Example 2, Fig 2, Chapt3 with BFS(b) spanning tree of Fig 1b, and \( D(G) \) in Fig 1c be the digraph obtained by using the spanning tree directed edges and directing non spanning tree edges from lower vertex number to higher vertex number.

![Figure 1a](image1.png)

![Figure 1b](image2.png)
We see that

\[
V_G = \begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}
\quad \text{and} \quad
V_D = \begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
\]

Notice that \(V_D\) is strictly upper triangular, \(V(i,j) = 0\) for \(i \leq j\). There are two reasons for this. First \(D\) is acyclic so there always is such an ordering that makes \(V_D\) strictly upper triangular. The fact that this particular order works is due to the use of BFS to get \(D\) as an orientation of an undirected graph. The BFS ordering makes the matrix strictly upper triangular by directing the nontree edges from lower number to higher number (note the tree edges are already have such a direction), as shown above.

**Exercise 1.** Verify the statements made about the use of BFS to give an acyclic orientation of a connected graph \(G\) and, hence, an \(A_D\) that is strictly upper triangular.

For an vertex \(x\) of a directed graph, let \(\text{indeg}(x) = |\{y \mid yx \in E\}|\) and \(\text{outdeg}(x) = |\{y \mid xy \in E\}|\).

The following proposition is relevant when \(D\) is given a` priori (not as an orientation).

**Proposition 1.** Let \(D\) be an acyclic digraph. Then there are vertices \(x\) and \(y\) such that \(\text{indeg}(x) = \text{outdeg}(y) = 0\).

**proof:**
Let \(x\) and \(y\) be the end vertices of the longest (directed) path in \(D\). Then \(\text{indeg}(x) = \text{outdeg}(y) = 0\); otherwise there is either a longer path or a cycle.

**endproof;**
To find the strictly upper triangular ordering of an acyclic D, we first order a vertex x with indeg(x) = 0, and then remove x and its incident edges. Then we find a new vertex with indeg = 0 and order it and continue the process. Such an ordering of an acyclic digraph is often called a topological ordering.

We now present a little result about the matrix powers of either $V^g$ or $V^D$. If we write $V^n$ we will use ordinary integer arithmetic; $\text{bool}(V^n)$ means the arithmetic is Boolean ($1 + 1 = 1$; $1 + 0 = 1$; $1 \cdot 1 = 1$, $1 \cdot 0 = 0$).

**Proposition 2.** Let $V$ be the v-adjacency matrix of $G$ or $D$.

1. $\text{bool}(V^n(i,j)) = 1$ iff there is a path of length $1 \leq n$ from i to j;
   $\text{bool}(V^n(i,j)) = 0$, otherwise.
2. $V^n(i,j)$ is the number of walks of length $n$ from i to j.

**Exercise 2.** Prove both parts of proposition 2 by induction on $n$.

### 3. Spanning Trees and the Cycle and Cut Subspaces

In this section we will consider our graphs to be connected; this convenience results in no loss of generality since we could consider the connected components separately. A spanning tree of a connected graph $G = (X, E)$ with $|X| = n$ and $|E| = m$ is an (edge) subgraph $T = (X, F)$ with $F \subseteq E$ such that $T$ is a tree. The edges in $E$ are called branches of $T$ while those in $F-E$ are called chords of $T$. We have seen how to construct and use both the breadth-first and depth-first spanning trees.

Let $T = (X, F)$ be a spanning tree of $G$ and recall that $T - g$ has exactly two components, $C_1g$ and $C_2g$, for any $g$ in $F$. We will call the set of all edges in $E$ between $C_1g$ and $C_2g$ the fundamental cut or f-cut, $K_g$, associated with the branch $g$. Notice that an f-cut is analogous to the unique cycle, $K_e$, in $T + e$ (and hence $G$) associated with the chord $e$; we shall called such cycles fundamental cycles or f-cycles.

Recall that the notion of a cycle is independent of any particular spanning tree. The analogous general notion is that of a cut-set: a cut set $K$ is a minimal set of edges that disconnects the graph. It is immediate from the minimal nature of $K$ that $H = (X, E-K)$ has two components; also any f-cut is a cut-set.

We will now make matrices (over $\mathbb{F}_2$) to represent these cuts and cycles. Recall there are $n' = n - 1$ branches, hence $m' = |E| - |F| = m - n + 1 = m - n'$ chords nontree edges (note $m' + n' = m$).
We construct $K_T$, the \textbf{f-cut matrix} of size $n' \times m$ and $C_T$, the \textbf{f-cycle matrix} of size $m' \times m$ as follows. First we order the vertices and edges; we order the tree edges first followed by the chords as in Fig1b. Given any branch, $g$, there is a unique f-cut, $k_g$ associated with it; this is represented by a row, say $j$, of $K_T$ by defining $K_T(j,k) = 1$ if $j$ is in $k_g$ and $K_T(j,k) = 0$, otherwise. Similarly, given any chord, $e$, there is a unique cycle, $c_e$ associated with it; this is represented by a row, say $j$, of $C_T$ by defining $C_T(j,k) = 1$ if $j$ is in $c_e$ and $C_T(j,k) = 0$, otherwise.

\textbf{Example 2.} For the graph of Fig1b, we find

$$K_T = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0
\end{bmatrix} \quad \text{and} \quad C_T = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}.$$ 

Suppose we write $K_T = [I_6, K]$ and $C_T = [C, I_3]$ and consider the product $Z = C_T(K_T)^T$ over the field $F_2$. We see that $Z = C + K_T$ and that $Z = 0$. Over $F_2$ this must mean $C = K_T$, and this is no accident. It is clear that, in general, any cut-set must share an even number of edges with any cycle; this implies the product $Z = 0$ and $C = K_T$. We summarize in (and have proved)

\textbf{Theorem 1.} Any cut-set shares an even number of edges with any cycle. For f-cuts represented by $K_T = [I_n', K]$ and f-cycles represented by $C_T = [C, I_m']$, this is equivalent to the identity $C = K_T$.

Notice that, as in Example1, there is always an identity matrix of order $n'$ as part of $K_T$ and one of order $m'$ as part $C_T$. \textit{Hence the rows of both $K_T$ and $C_T$ are independent.} Since $Z$ (defined above) has $Z = 0$, we might think that the rows in both such a $K_T$ and $C_T$ span $(F_2)^m$. This, however is not true as seen in

\textbf{Example 3.} Consider the graph in Fig2 with the heavy lines indicating the spanning tree.
The cycle vector \( \mathbf{c} = [1 \ 1 \ 1 \ 1] \) is the only cycle, and the f-cut matrix is
\[
\mathbf{K}_T = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1
\end{bmatrix}.
\]
Over \( \mathbb{F}_2 \), we see that \( \mathbf{c} \) is the sum of the rows of \( \mathbf{K}_T \). The missing row vector to span \( (\mathbb{F}_2)^4 \) could be \([0 \ 0 \ 0 \ 1]\) which is neither a cut nor a cycle. However, we do find that any m-vector \( \mathbf{v} \) with an even number of edges in common with the f-cuts of \( \mathbf{K}_T \) must be in the cycle subspace which are all linear combinations of the rows of \( \mathbf{C}_T \), and similarly any m-vector \( \mathbf{w} \) with an even number of edges in common with the f-cycles of \( \mathbf{C}_T \) must be in the cut subspace which are all linear combinations of the rows of \( \mathbf{C}_T \).

**Theorem 2.** Any m-vector \( \mathbf{v} \) such that \( \mathbf{K}_T \mathbf{v} = 0 \) has \( \mathbf{v}^\top = \mathbf{c}^\top \mathbf{C}_T \) for some \( m' \)-vector \( \mathbf{c} \); and any m vector \( \mathbf{w} \) such that \( \mathbf{C}_T \mathbf{w} = 0 \) has \( \mathbf{w}^\top = \mathbf{k}^\top \mathbf{C}_T \) for some \( n' \)-vector \( \mathbf{k} \).

**Proof.** Let \( \mathbf{v}_1 \) and \( \mathbf{v}_2 \) be \( n' \)- and m'-vectors, respectively and \( \mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2] \). Then with \( \mathbf{K}_T \) and \( \mathbf{C}_T \) written as in Thm1, we obtain \( \mathbf{v}_1 + \mathbf{Kv}_2 = 0 \) so \( \mathbf{v}_1 = \mathbf{Kv}_2 = \mathbf{C}^\top \mathbf{v}_2 \). Thus, \( \mathbf{v}^\top = [\mathbf{v}_1^\top, \mathbf{v}_2^\top] = [\mathbf{v}_2^\top \mathbf{C}, \mathbf{v}_2^\top \mathbf{I}] = \mathbf{v}_2^\top \mathbf{C}_T \) and \( \mathbf{c} = \mathbf{v}_2 \). The w case is exactly analogous. The graph interpretation of \( \mathbf{v} \) is that it is the linear combination (\( \mathbb{F}_2 \) sum) of the f-cycles of the chords in \( \mathbf{v} \).

**Endproof.**

**Theorem 3.** Any cycle is a linear combination of the rows of \( \mathbf{C}_T \) and cut-set is a linear combination of the rows of \( \mathbf{K}_T \). Furthermore \( \mathbf{C} = \text{span}(\mathbf{C}_T) = \text{span}(\mathbf{C}_T') \) and \( \mathbf{K} = \text{span}(\mathbf{K}_T) = \text{span}(\mathbf{K}_T') \) for any two distinct spanning trees, \( \mathbf{T} \) and \( \mathbf{T}' \).

**Proof.** The first statement is a direct corollary of Thms 1 and 2 as are \( \mathbf{C}_T \subseteq \text{span}(\mathbf{C}_T') \) and \( \mathbf{C}_T \subseteq \text{span}(\mathbf{C}_T') \). Hence \( \text{span}(\mathbf{C}_T) \subseteq \text{span}(\mathbf{C}_T') \) and \( \text{span}(\mathbf{C}_T) \subseteq \text{span}(\mathbf{C}_T') \) and similarly for the cut-set matrices, completing the proof.

**Endproof.**

Thm3 and the independence of the rows of \( \mathbf{K}_T \) and \( \mathbf{C}_T \), respectively immediately imply

**Corollary 1.** The rows of \( \mathbf{K}_T \) and \( \mathbf{C}_T \), for any spanning tree, \( \mathbf{T} \), are bases for \( \mathbf{C} = \text{span}(\mathbf{C}_T) \) and \( \mathbf{K} = \text{span}(\mathbf{K}_T) \), respectively.

The well-defined (independent of the particular spanning tree) subspaces \( \mathbf{K} \) and \( \mathbf{C} \) are often called the cut subspace and cycle subspace, respectively.

There is another graph related matrix that is both interesting and useful, the **vertex-edge incidence matrix**, \( \mathbf{A}_0 \). \( \mathbf{A}_0 \) is an \( n \times m \) matrix defined row-wise (row \( j \)) by
(VEIM1) \[ A_0(j,k) = 1 \text{ if } j \text{ is in } \text{adj}(i); \ A_0(j,k) = 0 \text{ otherwise,} \]

or equivalently, column-wise (column k) by

(VEIM2) \[ A_0(i,k) = A_0(j,k) = 1 \text{ for edge } k = ij, \text{ and } A_0(p,k) = 0 \text{ for } p \neq i \text{ or } j. \]

From (VEIM1) it is clear that the sum (in \( F_2 \)) of all rows is zero, hence the rows are not independent. In fact \( A_0 \) has rank \( n' \), since we will show that any \( n' \) rows of \( A_0 \) are independent. Notice that the edge sets \( \text{adj}(i) \) are cut-like objects. They are not cut-sets if \( i \) is a separator of \( G \) (such single vertex separators are called 1-separators or cut-vertices). We will call these incident edge sets \textit{v-cuts}.

Let \( a \) be a row of \( A_0 \) corresponding to a v-cut. Since any vertex \( x \) in a cycle must have distinct \( y \) and \( z \) in the cycle different from \( x \) with \( y \) and \( z \) in \( \text{adj}(x) \), any cycle has an even number of edges in common with any v-cut. Using Thms 2 and 3, we see immediately that any all rows of \( A_0 \) are in \( K \); we put any \( n' \) rows in a submatrix and generically call it \( A \). We will show \( K = \text{span}(A) \) by showing that the rows of \( A \) are independent and, hence \( \text{dim(span}(A)\text{)} = \text{dim}(K) \). First, we consider

**Example 4.** Suppose we order the vertices and edges of the graph in Fig1a as follows.

![Figure 3](image)

Here we have done a BFS from vertex b to get the vertex ordering and we have ordered the tree edges in Fig1b as they were encounted in the BSF. Hence edge \( k \) is always incident upon vertex \( k + 1 \) in the tree.

The matrix \( A_0 \) corresponding to the graph in Fig3 is then (omitting zeros)

\[
A_0 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{bmatrix}
\]
Notice that if we delete the row corresponding to the root $b$ of the BFS we obtain a submatrix $A$ of $A_0$ of the form $A = [T, S]$ where the tree edge matrix $T$ is upper triangular with ones on the diagonal. $A$ is called a reduced incidence matrix. Hence the rows of $n' = n - 1 = 6$ rows of $A$ are independent.

**Exercise 3.** By ordering the vertices and edges appropriately show that any submatrix $A$ of the vertex-edge incidence matrix $A_0$ with $n'$ rows can be written as $A = [T, S]$ where $T$ is upper triangular with ones on the diagonal. Also show that if $C_T = [C, I]$ (order consistent with $A$) then $C_TA^T = 0$ which implies that $S = TC^T$. Notice that $T^{-1}$ exists.

We then have shown

**Theorem 4.** For $A_0$ as above with submatrix $A$, $K = \text{span}(A)$.

We consider the directed graph version of Thms 2-4 as

**Exercise 4.** Let $D = (X, E)$ with $n$, $n'$, $m$ and $m'$ defined as in the graph case. The major difference here is that we will be over the field $\mathbb{R}$, rather than $\mathbb{F}_2$ so $1 \neq -1$. Our matrices will have as entries numbers from the set \{-1, 0, 1\}. Another difference is that $xy$ and $yx$ can both be (directed) edges in $E$. An oriented cycle (or-cycle), $c$, is a cycle in the undirected graph $G(D)$ or a cycle of the form $[x, y, x]$, and an oriented cut-set (or-cut-set), $k$, is a cut-set of $G(D)$ and includes both $xy$ and $yx$ if they are in $E$ and $xy$ is in the cut-set of $G$. We will drop the adjective “oriented” if the context is clear. The orientation is arbitrary; a direction is specified by taking a directed edge from $c$ or $k$, and the remaining edges are consistent or opposed to the orientation. When making a cycle or cut-set vector, $v$, $v$ has one orientation and $-v$ has the opposite orientation.

A spanning tree, $T$, of $D$ is a spanning tree of $G(D)$, but, if $xy$ is a branch and $yx$ is in $F$, then $yx$ is a chord. As before branches give rise to $f$-cuts and chords give rise to $f$-cycles; the orientation of these $f$-cuts and $f$-cycles is taken from the orientation of the branches and chords.

[4.1] Define the matrices $K_T$ and $C_T$ in analogy with the graph case. Use $-1$ when the $f$-cycle edges or $f$-cut edges are opposed to the branch and chord orientation. Show that $K_T = [I_{n'}, K]$ and $C_T = [C, I_{m'}]$; the $-1$ entries are only in $K$ and $C$. Derive $K_T$ and $C_T$ for the digraph of Fig1c (find a spanning tree first). Show that the rows of both $K_T$ and $C_T$ are independent.

[4.2] Prove that $Z = C_T(K_T)^T = 0$ still holds. State (and prove) a Theorem analogous to Thm1.

[4.3] State (and prove) a Theorem analogous to Thm2.

[4.4] State (and prove) a Theorem analogous to Thm3 and the analogue of Cor1.
Prove that the rows in $\begin{bmatrix} I & K \\
C & I \end{bmatrix}$ are independent (hence the situation in Example 2 does not happen). Start by making the graph of Fig 2 into a digraph and work through the details on this example. Hint: if $Mx = 0$ for some nonzero $x = [a; b]$ partitioned as in $M$, then $x^TMx = 0$. Is this possible for nonzero $x$?

Define $A_0$ and $A$ in the digraph case and state and prove a Theorem analogous to Thm 4. Make Example 3 into a digraph and find $A_0$ and $A$ analogous to the graph case.

4. The Resistive Electrical Network Problem

The resistive electrical network problem is defined on a connected graph, $G$, with orientation, $D(G)$. Suppose there are $(n + n_b)$ vertices and $m$ edges. Associated with each edge, $e$ are a conductivity, $c_e > 0$ (or resistance, $r = 1/c$), an edge voltage, $V_e$, and a current, $I_e$. The edge voltage for an oriented edge $e = xy$ is defined as $V_e = u_x - u_y$, and this is why we need to orient $G$; $u_x$ and $u_y$ are vertex voltages. We will see that the orientation is arbitrary (we are essentially establishing a “sign” convention). We partition the vertices as $X = [U, B]$. The vertices in $U$ are called unknown (or internal) since we must solve for the corresponding vertex voltages, while the nodes in $B$ have known boundary values. There must be a boundary edge, $e = ub$, $u \in U$, $b \in B$ for each boundary vertex. Often (in EE), $B$ contains only one vertex called “ground” and other set voltages are handled by voltage source elements, but our formulation has some advantages. We allow each vertex $u$ in $U$ to have specified current source (or current influx), $s_u$.

There are two physical laws that specify the problem as a set of equations to be solved. We will specify $V$, $I$, and $c$ to be vectors of length $m$, $u$ will be the (unknown) vector of length $n$, and $b$ will be a vector of length $m$, but will be nonzero only for the boundary edges. We first have Ohm’s Law:

(OL) $I_k = c_k V_k = c_k(u_i - u_j)$ for ordered directed edge $ij$ and ordered nodes $i$ and $j$.

We also enforce Kirchoff’s current conservation law:

(KL) $\sum_j I_{k,ij} = s_i$ for all vertices $i \in U$ where $j \in \text{adj}(i)$.

We now derive the matrix of (OL) and (KL). We will let $I$, $V$, and $b$ be $m$-vectors associated with the currents, edges voltages, and boundary edges, respectively. The vertex voltages, $u$, and sources, $s$, will be $n$-vectors. $C$ will be an $m \times m$ diagonal matrix of the conductivities.

Consider the matrix $A_0$ of Exercise 4.6. Recall that each column has a 1 and a –1, so the rows add to zero and, hence, are not independent in the matrix (linear) algebra sense. This
reflects the physical necessity to specify at least one boundary vertex. We let \( A \) be the \( n \times m \) submatrix of the \( (n + n_b) \times m \) matrix of the directed graph, \( D \), defining the resistive network (in the first paragraph of this section) obtained by deleting the rows corresponding to boundary nodes. This means, that by the \( e = u \) convention for the directed boundary edges, these columns of \( A \) have a single 1 in the row \( j = u \). As before \( A \) is called the reduced incidence matrix. We use a BFS spanning sub-tree, as in Example 4, from the first boundary vertex, say \( b_1 \), to order the vertices and edges; but this spanning tree includes only one boundary edge as a branch of the tree, and the other boundary edges are taken to be chords. By orienting the tree edges consistently \( A \), can again be written as \( A = [T,S] \) where \( T \) has the tree edges and is upper triangular with ones on the diagonal.

**Example 5.** Consider a resistive network on the graph of Figure 3 where we add two boundary vertices, \( b_1 \) and \( b_2 \). An orientation is given for the edges and this orientation is chosen so the oriented branch \( j \) is of the form \( jk \) (\( k < j \) or \( k \) is \( b_1 \)) in the BFS spanning tree from \( b_1 \)

![Figure 4](image_url)

Tree edges are bold and contain only one boundary edge

Let’s form the reduced incidence matrix \( A \) for this digraph. We see (leaving out the zeros)

\[
A = \begin{bmatrix}
1 & -1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}.
\]

Notice the first 7 x 7 submatrix is a upper triangular matrix with ones on the diagonal

**Exercise 5.** Verify that an ordering an orientation such as given in Example 5 is always possible.
This means that the rows of A are independent, so our ultimate matrix equation will have a unique solution.

Now, referring to example5 for specification, let’s reformulate (OL) and (KL) using A and the \( m \times m \) diagonal matrix \( C = \text{diag}(c) \) (\( C(j,j) = c(j); \) zero, otherwise). The matrix form of Ohm’s Law becomes

\[
(\text{OLM}) \quad I = C \cdot V = C(A^T u - b).
\]

This follows since the multiplication of \( u \) by \( A^T \) takes the appropriate difference in the \( u \) vector, and we need to fix things up for boundary edges in the vector \( b \). The matrix form is Kirchoff’s law is equally simple:

\[
(\text{KLM}) \quad A \cdot I = s,
\]

since multiplication by \( A \) just sums the node currents in the appropriate way. Combining (OLM) and (KLM) gives an \( n \times n \) matrix (equation)

\[
(\text{RN1}) \quad ACA^T u = s + AC \cdot b.
\]

The \( AC \cdot b \) term on the right gives the equivalent currents induced by our boundary values for the internal nodes connected to the boundary. The matrix

\[
(\text{RN2}) \quad M = ACA^T
\]

is the **vertex (or nodal) admittance matrix**. Since the rows of \( A \) are independent and all \( c_i > 0 \), \( M \) is symmetric (obvious), positive definite and, hence, nonsingular. This means that there is a unique solution \( u \).

**Exercise 6.** Show that \( M \) is nonsingular by showing that: if \( Mx = 0 \) for nonzero \( x \), then \( x^T M x = 0 \), and this means \( y^T y = 0 \) for nonzero \( y = C^{1/2} A^T x \) since \( Ax \neq 0 \).

**Exercise 7.** Show that the linear system for Example5, assuming all \( c_i = 1 \), all \( s_i = 1 \), and \( b = 0 \) is:

\[
\begin{bmatrix}
4 & -1 & -1 & -1 \\
-1 & 2 & -1 & \\
-1 & -1 & 3 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 3 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 3
\end{bmatrix}
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7
\end{bmatrix}
= \begin{bmatrix}1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1\end{bmatrix}.
\]
One could use the sparse matrix features of Matlab to assemble (build) the matrix $M$ from $A$. However, we must then decide how to enter $A$ as data as well as the quantities in $C$ and $b$. The code resistorsim in the appendix builds $M$ resistor by resistor and also assembles the total equivalent current vector. Enjoy!

Note to djr: define KVL (Kirchoff voltage law) and show the relation in an extended exercise like Exercise4.

Appendix: Matlab Code for resistorsim

(save as resistorsim.m in a matlab path to run)

```matlab
function [nodev, currents, M] = resistorsim(rcirc, bdary, source, printsys)
    
    % Solves resistive electrical network equations to give a vector
    % node voltages (nodev) and an array of edge currents. There are
    % m directed edges $(i,j)$, nb boundary edges, and n nonboundary
    % (internal)
    % nodes. The undirected graph of the network must be connected with at
    % least one edge from a nonboundary node to a boundary node. rcirc is
    % an mx3 array with each row of the form $(i,j, conductivity)$ for
    % nodes $i,j$ and conductivity of the edge $(i,j)$ resistor; or, when n=1,
    % rcirc=[]. bdary is a nb x 3 array of internal nodes with
    % their boundary edge to a boundary node with fixed node value;
    % each row has the form $(i, bvalue, conductivity)$. source is an
    % ns x 2 array of current sources with each row of the form
    % $(i, sourcevalue)$ unless source = []. The sign convention is:
    % $(i,j)$ current from $i$ (positive) into $j$ (negative); $(i, boundarynode)
    % from $i$ (positive) to boundarynode (negative); source currents go
    % into internal nodes (negative). If printsys is the string 'yes',
    % resistorsim prints the linear system.
    
    % Get dimensions.
    sm = size(rcirc);
    m = sm(1);
    if m ~= 0
        n = max(max(rcirc(:,1)), max(rcirc(:,2)));
    else
        n=1;
    end
    snb = size(bdary(:,1));
    nb = snb(1);
    szs = size(source);
    ns = szs(1);
    
    % Initialize matrix M and right hand side b to zero.
    M = zeros(n,n);
    M = sparse(n,n);
```
b = zeros(n,1);

% Assemble internal node matrix elements into M.
%
for k = [1: m]
i = rcirc(k,1);
j = rcirc(k,2);
cvalue = rcirc(k,3);
M(i,i) = M(i,i) + cvalue;
M(j,j) = M(j,j) + cvalue;
M(i,j) = M(i,j) - cvalue;
M(j,i) = M(j,i) - cvalue;
end
%
% Assemble boundary node (diagonal) matrix elements into M.
%
for k = [1: nb]
i = bdary(k,1);
cvalue = bdary(k,3);
M(i,i) = M(i,i) + cvalue;
end
%
% Assemble source into b.
%
for k = [1: ns]
i = source(k,1);
svalue = source(k,2);
b(i) = b(i) + svalue;
end
%
% Assemble boundary value * boundary conductivity into b.
%
for k = [1: nb]
i = bdary(k,1);
b(i) = b(i) + bdary(k,2) * bdary(k,3);
end
%
% Print matrix M and rhs b if printsys is 'yes'.
%
if printsys == 'yes'
    disp(' System Matrix M');
    M
    disp(' System right hand side b');
    b
end
%
% Compute node voltages.
%
nodev = M\b;
%
% Create and compute (m + nb) x 3 array , currents, whose rows
% have the form (i, j, ijcurrent) or the form (i, bvalue, bcurrent).
%
currents = [];
for k = [1:m]
i = rcirc(k,1);
j = rcirc(k,2);
ijcurrent = rcirc(k,3) * (nodev(i) - nodev(j));
currents(k,:) = [i , j, ijcurrent];
end
for k = [1: nb];
    i = bdary(k,1);
    bval = bdary(k,2);
    ibcurrent = bdary(k,3) * (nodev(i)-bval);
currents = [currents ; i, bval, ibcurrent];
end
\% v 1/16/01