Isomorphism Between Graphs With Distinct Eigenvalues

Tom Leighton

April, 1979

1. Introduction

Two undirected graphs are said to be *isomorphic* if there exists a one-to-one correspondence between their nodes which preserves adjacency. There is no known polynomial time algorithm which will determine whether or not two graphs are isomorphic. A well-known necessary condition for isomorphism is that the two graphs be cospectral (i.e., their adjacency matrices have the same set of eigenvalues). Since the eigenvalues of an $n \times n$ symmetric matrix may be computed in $O(n^3)$ time, cospectral pairs of nonisomorphic graphs have received much attention in the literature.

In this paper, we present an $O(n^3)$ algorithm for determining isomorphism between two cospectral n-node graphs ~~with~~ with ~~unique~~ distinct eigenvalues. In addition, we present an algorithm which completely determine the automorphism group of a graph with distinct eigenvalues in $O(n^3)$ time.

2. The ~~Basic~~ Theoretical Framework

Let A and B be any pair of $n \times n$ nonsingular symmetric cospectral matrices. Let D be the diagonal matrix consisting of the eigenvalues of A arranged on the diagonal in nondecreasing order. Let U and V be orthogonal matrices consisting of eigenvectors of A and B, respectively. Arrange the columns of U and V (the eigenvectors of A and B) so that $AU = UD$ and $BV = VD$. Since A and B are symmetric and nonsingular, ~~then~~ U, V and D are real matrices and may be computed in $O(n^3)$ time. ?

In addition, let P denote a permutation matrix. Note that A and B are isomorphic if and only if there exists a permutation matrix P such that $B = P^T A P$. In the following theorem, we assume that E is a block diagonal matrix of the form

$$E = \begin{pmatrix} E_1 & & & \\ & E_2 & & 0 \\ & 0 & \ddots & \\ & & & E_m \end{pmatrix}$$

where each E_i is orthogonal and has dimension equal to the multiplicity of the ith (when arranged in nondecreasing order) distinct eigenvalue of A.

Theorem 1: Let A, B, P, U, V and E be as described above. Then $B = P^T A P$ if and only if $UE = PV$ for some E.

Proof: By definition, we know that $A = UDU^T$ and $B = VDV^T$. Thus $B = P^T A P$

implies that $VDV^T = P^T UDU^T P$

$$\Rightarrow U^T PVD = DU^T PV.$$

Let $E = U^T PV$. Clearly $UE = PV$. Since $ED = DE$ (above), E must have the prescribed block diagonal form. E is orthogonal since U, V, and P are each orthogonal.

Conversely, assume that $UE = PV$ for some E with the prescribed block diagonal form. Then $ED = DE$ and thus $U^T PVD = DU^T PV$. Reversing the above argument then yields $B = P^T A P$ $\square$

Theorem 1 is ~~particularly~~ especially useful when all the eigenvalues of A are different. ~~[illegible]~~

In particular, we note the following corollaries.

<u>Corollary 1</u>: If A has n distinct eigenvalues, then $B = P^T A P$ if and only if there exists a diagonal matrix E with diagonal elements ± 1 such that $UE = PV$.

<u>Proof</u>: The dimension of each block E_i in Theorem 1 must be 1 and since each E_i is orthogonal, $E_i = [\pm 1]$ $\square$

<u>Corollary 2</u>: If A has distinct eigenvalues and $P \in \mathrm{Aut}(A)$, then $P^2 = I$.

<u>Proof</u>: If $P \in \mathrm{Aut}(A)$, then $A = P^T A P$ and by Corollary 1, $P = UEU^T$ for some

diagonal matrix E with ± 1's on the diagonal. Thus $P^2 = UEU^TUEU^T = UE^2U^T = UU^T = I$ since $UU^T = I$ and $E^2 = I$ □

Follows by George Strang

<u>Corollary 3</u>: If A has distinct eigenvalues ~~and $P_1, P_2 \in Aut(A)$, th~~ then $Aut(A)$ is abelian.

<u>Proof</u>: Given $P_1, P_2 \in Aut(A)$, we know from Corollary 1 that $P_1 = UE_1U^T$ and $P_2 = UE_2U^T$ where E_1 and E_2 are diagonal matrices. Thus $P_1 P_2 = UE_1U^TUE_2U^T = UE_1E_2U^T = UE_2E_1U^T = UE_2U^TUE_1U^T = P_2 P_1$ since E_1 and E_2 commute □

Corollary 4: If A has distinct eigenvalues

then $o(\mathrm{Aut}(A)) = 2^i$ for some $i \leq n$.

<u>Proof</u>: The result follows directly from Corollaries 2 and 3 $\square$

It is clear from corollaries 1-4 that the conditions imposed by Theorem 1 are fairly strong, particularly for the case when ~~the eigenvalues~~ all the eigenvalues of a matrix are distinct. In addition, the relation $UE = PV$ is much easier to work with ~~from an algorithmic~~ algorithmically than is the associated relation $B = P^{T}AP$. In particular, when all the eigenvalues of A are distinct,

$$E = \begin{pmatrix} \pm 1 & & 0 \\ & \pm 1 & \\ 0 & & \ddots \ \pm 1 \end{pmatrix}$$

and we will show to find all solutions to $UE = PV$ in $O(n^3)$ time.

how do we solve $UE = PV$ for π parameters

3. Restrictions on P
~~Outline of~~
~~3. A The Algorithm~~

Henceforth, we assume that A has distinct eigenvalues and that $E = \begin{pmatrix} e_1 & & 0 \\ & \ddots & \\ 0 & & e_n \end{pmatrix}$ where $e_i = \pm 1$ for $1 \le i \le n$. From Corollary 1, we know that $B = P^T A P$ if and only if $UE = PV$ for some such E. In this section we ~~wish~~ ~~We now concentrate~~ ~~show~~ use this relation to partition U and V into blocks ~~our efforts on characterizing these E for~~ of rows such that P must map a given block of ~~which there exists a permutation matrix P~~ rows of V into the corresponding block of rows ~~such that UE = PV. Given any such E,~~ of U, independent of the value of E. ~~we then~~ ~~P is then trivially found from the relation by reducing the others the desired result. P = UEV^T.~~

Define u_j to be the jth column (eigenvector) of U and v_j to be the jth column of V. Then $UE = PV$ if and only if $e_j u_j = Pv_j$ for every j. Since $e_j = \pm 1$, P must permute the rows of V so that

$|u_{ij}| = |v_{p(i)j}|$

~~the i-th row of U is~~

for all i, j. Thus we may partition the rows of U and V into blocks corresponding to row vectors of identical magnitudes (of absolute values). P must then map rows in a given block of V into rows of the corresponding block of rows in U.

Such a partition can be easily computed in $O(n^3)$ time. As an example, consider the ~~[illegible]~~ ~~and, in fact, does not involve any [illegible] restrictions on the c_i's. As an example refinement of the partition in Figure 1.~~ partitioning the matrix in Figure 1.

$$\begin{pmatrix} 1 & 1 & 2 & -3 & 4 & 0 \\ 1 & -1 & -2 & 3 & 4 & 0 \\ \hline 1 & 2 & 2 & 4 & 2 & 1 \\ \hline -1 & 2 & 3 & 4 & 2 & 1 \\ \hline 2 & 3 & 3 & 4 & 2 & 2 \\ \hline -2 & -3 & 3 & -3 & 2 & 3 \end{pmatrix}$$

step	partition
0	(1, 2, 3, 4, 5, 6)
1	(1, 2, 3, 4) (5, 6)
2	(1, 2) (3, 4) (5, 6)
3	(1, 2) (3) (4) (5, 6)
4	(1, 2) (3) (4) (5) (6)
5	(1, 2) (3) (4) (5) (6)
6	(1, 2) (3) (4) (5) (6)

~~Figure 1~~

Figure 1

At the ith step, we refine the previous partition to reflect information about the magnitudes of the elements in the ith column.

Once the partition according to magnitudes is complete, the magnitude of an entry need not be considered further (as all pertinent information about the magnitude is incorporated into the partition). Thus we need only know whether an entry is positive (+), negative (-) or zero (0). ~~As might be expected, the distribu~~

The partition described above may be further refined through examination of the numbers of +'s and -'s in a column of a block of rows. For example, if the number of positive elements differs from the number of negative elements in a column of a block, then that block may be divided so as to seperate the positive elements from the negative elements. This is due to the fact that $e_i u_i = P v_i$ and (if u_i' is a block of u_i, v_i' is the corresponding block of v_i and P' is the restriction of P to v_i') $e_i u_i' = P' v_i'$. Assume v_i' is non-zero and has x ~~per~~ positive elements and y negative elements. Then ~~(~~u_i' is non zero and has x positive elements ($e_i = 1$) or has x negative elements ($e_i = -1$). Otherwise, A and B are immediately shown not to be isomorphic. Thus, if $x \neq y$, then the x ~~block~~ positive elements in v_i' are mapped by P into the set of x identically ~~elem~~ signed elements of u_i' — independent of the choice of e_i (in fact, we ~~may~~ can determine e_i). Thus, if both x and y are nonzero, then we may (nontrivially) refine the partitions of U and V so that u_i' and v_i' are separated into a positive column and a negative column. As an illustration consider the refinement of the partition of the matrix ~~imposed on the matrix in~~ ~~Figure 2~~ shown in Figure 2.

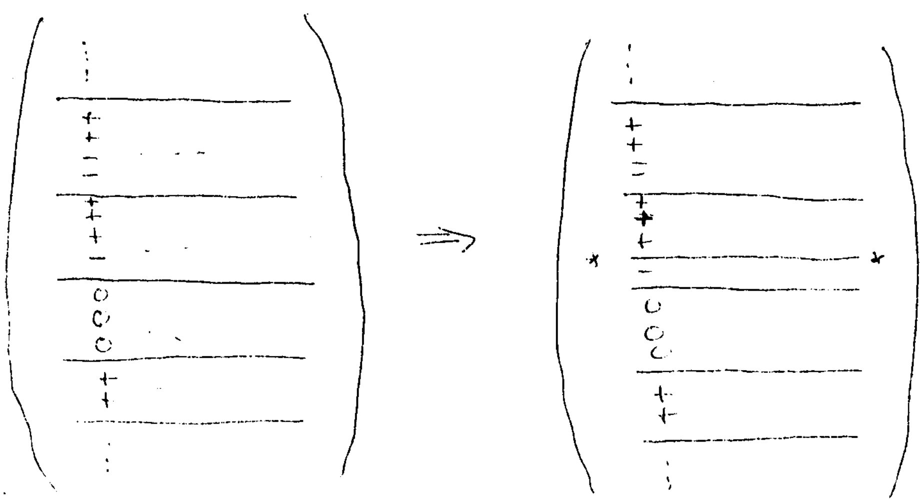

Figure 2

We may further refine the partition of U and V by inspecting the number of positive and negative elements in $v_i' \bullet v_j'$ where v_i' and v_j' are non-zero columns in a block of rows and $v_i' \bullet v_j'$ is the Hadamard (element-wise) product of v_i' and v_j'. This is due to the fact that $e_i e_j (u_i' \circ u_j') = p'(v_i' \circ v_j')$. Since $e_i e_j = \pm 1$, ~~the same argument~~ we may argue as before and conclude that if the number of positive elements differs from the number of negative elements in $v_i' \circ v_j'$, then we may ~~refine~~ split the ~~partition~~ block containing v_i' and v_j' into two blocks (if $v_i' \circ v_j'$ contain both ~~identical~~ positive and negatives elements — one otherwise) blocks so that the ~~according to the~~ ~~value~~ sign of $v_i' \circ v_j'$ is uniform in each block. As an example, consider the refinement illustrated in Figure 3.

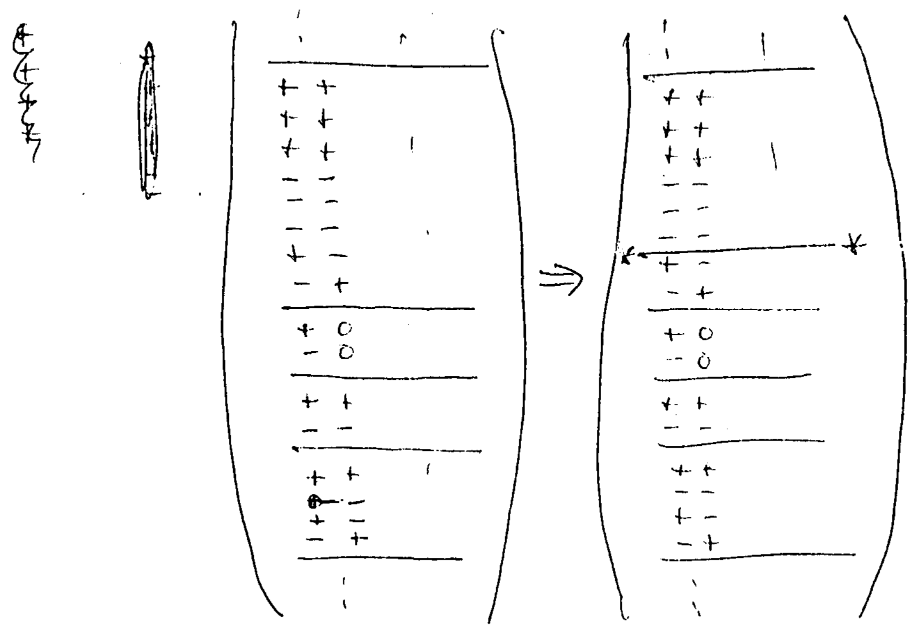

Figure 3

In general, we may refine the partition of U and V whenever there are non-zero columns $v_{i_1}', \ldots, v_{i_k}'$ in a block such that $v_{i_1}' \circ \cdots \circ v_{i_k}'$ has unequal numbers of positive and negative elements (and at least one of each). If we refine the partition as much as is possible by the repeated application of this procedure, then for every block and for any k non-zero columns $v_{i_1}, \ldots, v_{i_k}$ in ~~this~~ that block, ~~either~~ $v'_{i_1} \circ \cdots \circ v'_{i_k}$ is either uniformly signed or has an equal number of positive and negative elements. In Section 5, we show that once ~~the~~ ~~on the~~ U and V are in this refined form, it will be straightforward to find all solution to $UE = PV$. In the following section, we show how to find such a partition of U and V in $O(n^3)$ time.

4. Calculation of the Partition

Consider the following algorithm for partitioning U and V.

Algorithm 1:

Main Program

Calculate the partition according to magnitudes ~~des~~ as described in section 3. Denote the blocks $B_1, \ldots, B_r$.

Initialize $f_{B_j}(i) \leftarrow \begin{cases} 0 & \text{if the } i\text{th column of } B_j \text{ is nonzero} \\ 1 & \text{" " " " " is zero} \end{cases}$

for $1 \le i \le n$ and $1 \le j \le r$

Call Subroutine $(B_1, \ldots, B_r, f_{B_1}(*), \ldots, f_{B_r}(*))$

STOP

(Recursive) Subroutine $(B_1, \ldots, B_r, f_{B_1}(*), \ldots, f_{B_r}(*))$

Do 30 for $j = 1, r$ (i.e. for each block)

$S \leftarrow \emptyset$

Do 20 for $i = 1, n$ (i.e. for each column of the block)

~~If ($f_{B_j}(i)=0$ then Do 10 for each subset $S' \subseteq S$~~

(*) Do 10 for each subset $S' \subseteq S$ while $f_{B_j}(i)=0$

Calculate $w' \leftarrow v_i' \cdot \prod_{s \in S'} v_s'$

If w' is uniformly signed then set $f_{B_j}(i) \leftarrow 1$ and go to 10

If w' is unevenly signed then partition B_j into B_j^+ and B_j^- according to the signs of w'. Set $f_{B_j}(i)=$
~~set $f_{B_j^+}(k) \leftarrow f_{B_j^-}(k)$ set $f_{B_j}(i) \leftarrow 1$~~
initialize $f_{B_j^+}(k) \leftarrow f_{B_j^-}(k) \leftarrow f_{B_j}(k)$ for $1 \le k \le n$. Call Subroutine $(B_j^+, B_j^-, f_{B_j^+}(*), f_{B_j^-}(*))$. Go to 30

```
10        CONTINUE
20        CONTINUE
30     CONTINUE
       RETURN
```

Lemma 1: Algorithm 1 correctly and completely partitions U and V.

Proof: By induction on the number

of refinements necessary to partition the matrix.

<u>Basis</u>: Assume u and v are in the form described at the end of section 3 (i.e., for every block of ~~and~~ rows and for any k non-zero colums $v'_{i_1}, \ldots, v'_{i_k}$ of that block, $v'_{i_1} \circ \cdots \circ v'_{i_k}$ is either uniformly signed or has an equal number of positive and negative elements. Since Algorithm ~~A~~1 only ~~allows~~ splits a block when k non-zero columns ~~a refinement of the partition when~~ $v'_{i_1}, \ldots, v'_{i_k}$ of that block are found such that $v'_{i_1} \circ \cdots \circ v'_{i_k}$ is neither uniformly signed nor contains an equal number of positive and negative elements, it is clear that Algorithm 1 works in this case.

<u>Induction</u>: Assume Algorithm 1 works if fewer than $q \geq 1$ ~~partitions~~ refinements of the original partition are necessary to completely partition u and v.

Further assume that U and V require exactly q refinements (it does matter in which order we make the refinements so this notion is well-defined) to completely partition. Thus, there exist k columns $v_{i_1}', \ldots, v_{i_k}'$ of some block B_j such that $v_{i_1}' \circ \cdots \circ v_{i_k}'$ is neither uniformly signed nor equally signed. We want to show that Algorithm 1 makes at least one refinement in order to apply the reduction. Assume no refinement has been made up to the point where we have examined each of the k columns $v_{i_1}', \ldots, v_{i_k}'$ of the jth block B_j. Then we know that for $1 \le \ell \le k$ $\exists S_\ell' \subseteq S$ such that $v_{i_\ell}' \circ \prod_{s \in S_\ell'} v_s'$ is uniformly signed (this includes the case that $v_{i_\ell}' \in S$ since $v_{i_\ell}' \circ v_{i_\ell}'$ is, trivially, uniformly signed). Thus

$\prod^{1 \le \ell \le k} \left(v_{i_\ell}' \circ \prod^{s \in S_\ell'} v_{\phi_s}' \right)$ $= \left(\prod^{1 \le \ell \le k} v_{i_\ell}' \right) \circ \left(\prod^{1 \le \ell \le k} \prod^{s \in S_\ell'} v_{\phi_s}' \right)$ is uniformly signed.

The product (with repetition) of vectors in S ~~[illegible]~~ is by definition either uniformly signed or equally signed. Thus $\prod^{1 \le \ell \le k} v_{i_\ell}'$ is uniformly ~~signed or~~ or equally signed. But this contradicts the assumption that $v_{i_1}' \circ \cdots \circ v_{i_k}'$ was neither uniformly nor equally signed. Thus Algorithm 1 makes at least one refinement and we may apply the induction to the refined partition.

Had there not been any reference to the $P_{B_j}(i)$ ~~In order to complete the proof, we~~ in step (*) of the ~~preceding~~ program, our ~~[illegible]~~ proof would now be complete. ~~[illegible]~~ ~~must show that the procedure which were~~ Since reference to the $(B_j(i))$ in step (*) is ~~[illegible]~~ necessary for ~~changed before the refinement step~~ the algorithm to run quickly, we must show that such reference does not impair the algorithm's ability to find all possible refinements. To do ~~Assume~~ ~~that~~ this we observe that $f_B(i) = 1$ for some block B of a refinement (i.e. before we look at column) only if v_i' is all zeros or if there are k columns $v_{i_1}'', \ldots, v_{i_k}''$ of a super block $B' \supseteq B$ of a previous partition ~~containing~~ ~~containing~~ ~~B' containing~~ such that $v_i'' \bullet v_{i_1}'' \bullet \cdots \bullet v_{i_k}''$ is uniformly signed where v_i'' is the column of B' containing v_i' and $\{i_1 < \cdots < i_k < i\}$. Thus $v_i' \bullet v_{i_1}' \bullet \cdots v_{i_k}'$ is uniformly signed in B $(i_1 < \cdots < i_k < i)$ and, by induction, we can argue that the ith column need not be considered when looking for refinements of B. □ ~~This completes the proof of the lem.~~

Linearity

In order to show that Algorithm 1 can be executed in $O(n^3)$ time, we must show that the set S does not grow too large.

Lemma 2: $|S| \le \log m$ for a block B of m rows.

Proof: Let $S = \{v'_{i_1}, \ldots, v'_{i_k}\}$ be a collection of columns of a block B, such that for any subset $S' \subseteq S$ $\prod^{s \in S'} v'_s$ is equally signed. Let Λ the integer j $(0 \le j < 2^k)$ denote the k-bit binary word $j_1 \cdots j_k$. Define x_j to be number of rows such that the entry in that row of $\Lambda v'_{i_\ell}$ has sign $(-1)^{j_\ell}$ for $1 \le \ell \le k$. We will show that $x_0 = x_1 = \cdots = x_{2^k-1} = \frac{m}{2^k}$ which immediately implies that ~~(and thus $2^k \le m$ $\Leftrightarrow$)~~ $2^k \le m$ and ~~immediate~~ thus that $|S| \le \log m$. Note that this is equivalent to showing that we may arrange the rows of B so that $v'_{i_1} \ldots v'_{i_k}$ has the form shown in Figure 4. In this illustration

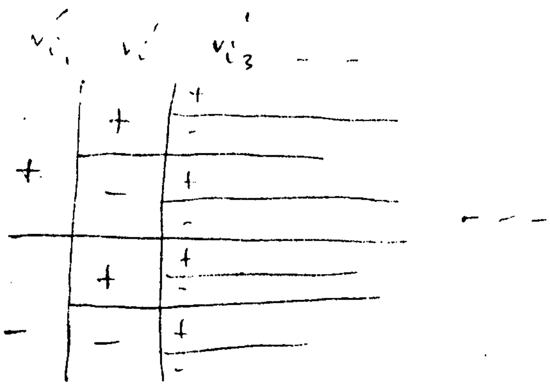

Figure 4

each subblock of signs in a given column is assumed to have the same size.

By definition, the number of rows for which $\prod_{s \in s'} V_s'$ is positive is $\frac{m}{2}$ for any $s' \subseteq S$. Thus for any $T \subseteq \{1, \ldots, K\}$, $\sum_{j \in T'} X_j = \frac{m}{2}$ where $T' = \{ j \mid \sum_{\ell \in T} j_\ell \equiv 0 \bmod 2 \}$. In the special case when $T = \emptyset$, $T' = \{0, \ldots, 2^K - 1\}$ and $\sum_{j \in T'} X_j = m$.

Since there are 2^K subsets of T, there are 2^K such restrictions on the x_j. Expressing this information in vector notation, we have that

$$A\vec{x} = \vec{b} \quad \text{where} \quad \vec{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_{2^k-1} \end{pmatrix}, \ \vec{b} = \begin{pmatrix} m \\ m/2 \\ m/2 \\ \vdots \\ m/2 \end{pmatrix}$$

and $A = [a_{tj}]$ where $a_{tj} = t_1 j_1 + \cdots + t_k j_k + 1 \pmod 2$, ~~and~~ $t = t_1 \cdots t_k$ and $j = j_1 \cdots j_k$ are the binary representations of t and j. It is easy to show that $x_0 = x_1 = \cdots = x_{2^k-1} = \frac{m}{2^k}$ is a solution of this system of equations (each row of A other than the first has 2^{k-1} 1's and the rest zeroes). All that remains to be shown is that the rank of A, $r(A)$, is 2^k. This is proved by induction on k. If $k=1$, then $A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ and the result is true. Assume that $r(A_{k-1}) = 2^{k-1}$ for $k \geq 2$. By observing that A_k may be evenly partitioned into 4 equal sized blocks as $A_k = \left(\begin{array}{c|c} A_{k-1} & A_{k-1} \\ \hline A_{k-1} & J_{k-1} - A_{k-1} \end{array}\right)$ where

J_{k-1} is the $(k-1)\times(k-1)$ matrix of ones,

it is clear that:

$$r(A_k) = r\left(\left[\begin{array}{c|c} A_{k-1} & A_{k-1} \\ \hline A_{k-1} & J_{k-1} - A_{k-1} \end{array}\right]\right)$$

$$= r\left(\left[\begin{array}{c|c} A_{k-1} & A_{k-1} \\ \hline 0 & J_{k-1} - 2A_{k-1} \end{array}\right]\right)$$

$$\cancel{r(A_{k-1}) + r} = r\left(\left[\begin{array}{c|c} A_{k-1} & A_{k-1} \\ \hline 0 & 2A_{k-1} - J_{k-1} \end{array}\right]\right)$$

$$\cancel{= 2^{k-1} + r(J_{k-1} - 2A_{k-1})}$$

$$\cancel{= 2^{k-1} + r(2A_1)} = r\left(\left[\begin{array}{c|c} A_{k-1} & A_{k-1} \\ \hline 0 & A_{k-1} \end{array}\right]\right)$$

$$= r\left(\left[\begin{array}{c|c} A_{k-1} & 0 \\ \hline 0 & A_{k-1} \end{array}\right]\right)$$

$$= 2\, r(A_{k-1})$$
$$= 2^k \quad \text{by induction.}$$

Thus $x_0 = x_1 = \cdots = x_{2^k-1} = \frac{m}{2^k}$ is the only solution to $A\vec{x} = \vec{b}$ and S has the desired form □

Lemma 3 Algorithm 1 runs in $O(n^3)$ time.

Proof: We will show by induction on the number of allowable refinements that the subroutine uses $O(m^2 g)$ time to completely partition a block B of m rows with no more than g columns i such that $f_B(i) = 0$. If B is already completely refined, then the subroutine clearly verifies this in $O(m^2 g)$ time (the do 20 loop executes completely at most g times, the do 10 loop executes at most $2^{|S|} \leq m$ times, and it takes $O(m)$ time to calculate w' and check its signs, assuming that we store previously checked values of w'). Assume the result is true if B requires r-1 refinements for $r \geq 1$. Assume that B requires r refinements. Then using the inductive hypothesis, the algorithm completely partitions B in $O(g_1 m^2 + (g-g_1)m_1^2 + (g-g_1)(m-m_1)^2)$ time ~~where~~ where g_1 is the number of columns examined before the first refinement ~~partition~~ is made and m_1 is the number of rows in the B^+ subblock of B. This is clearly $O(m^2 g)$ time and the induction is complete.

Thus the main program (which ~~par~~ completely partitions U and V) runs in $O(n^3 + \sum m_j^2 n)$ ~~where the~~ $\leq O(n^3)$ time where m_j is the number of rows in the jth block of the partition according to magnitudes[1] ~~Since~~ ~~the partition by magnitudes is trivial [illegible]~~

5. Solutions of $UE = PV$

Once U and V are completely partitioned by Algorithm 1, it is not difficult to find all solutions (if any) to $UE = PV$. In particular, we will characterize the E's for which there is a ~~a permutation~~ a permutation P mapping V to UE. The permutations are then easily recovered by using the relation $P = UEV^T$.

Given a completely refined block B of V and the corresponding refined block of U, rearrange the rows so that the columns in S have the form shown in Figure 4. The columns which are ~~present [illegible]~~ not in S are, by definition, linearly dependent on the columns in S. [illegible] for every column i of B, there is a subset [illegible]

~~Columns of S~~ = $\{v_{i_1}', \ldots, v_{i_k}'\}$ $(k \geq 0)$ of S such that $v_i' \bullet v_{i_1}' \bullet \cdots \bullet v_{i_k}'$ is uniformly signed.

For A and B to be isomorphic, the same relationship must hold for the corresponding columns in U (i.e. $u_i \bullet u_{i_1} \bullet \cdots \bullet u_{i_k}$ is uniformly signed). Thus we must have

~~the sign of $v_i' \bullet v_{i_1}'$~~

$$\left\{ \begin{array}{ll} 1 & \text{if } v_i' \bullet v_{i_1}' \cdots v_{i_k}' \cdot u_i \bullet u_{i_1} \cdots u_{i_k} \text{ is uniformly positive} \\ -1 & \text{if } v_i' \bullet v_{i_1}' \cdots v_{i_k}' \cdot u_i \cdot u_{i_1} \cdots u_{i_k} \text{ is uniformly negative} \end{array} \right\}.$$

[illegible] that these restrictions [illegible] precisely characterize those signs of columns of U which permit a permutation of B into the corresponding block of UE. Since $e_i^2 = 1$, these equations can be solved in $O(n^k m)$ time for each block of m rows.

~~The~~ In order to find the solutions to $UE = PV$, we must ~~then~~ determine the solutions to the subproblem for each block of the complete partition, and then compute the intersection of these solution spaces ~~restrictions~~. This is easily done in $O(\sum^{1\le j\le r} n^2 m_j + n^3) = O(n^3)$ time where the jth block has m_j rows. By the preceding argument, we have proved the following two ~~two~~ theorems.

<u>Theorem 2</u>: If A and B are $n\times n$ symmetric matrices with distinct eigenvalues, then we can completely determine the ~~a~~ set of isomorphisms from B to A in $O(n^3)$ time. ~~In addition,~~ ~~the set of isomorphisms may be characterized.~~

<u>Theorem 3</u>: If A is a symmetric matrix with distinct eigenvalues, then

$$\mathrm{Aut}(A) = \left\{ P \mid P = U E U^T, \quad E = \begin{pmatrix} e_1 & & 0 \\ & \ddots & \\ 0 & & e_n \end{pmatrix}, \right.$$

$e_{i_1} = \pm 1$, $e_{i_2} = \pm 1$, ..., $e_{i_k} = \pm 1$ for some subset $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$

with $k \geq 0$ and e_{i_ℓ} some fixed product of the elements

in $\left. \{-1, e_{i_1}, \dots, e_{i_k}\} \right\}$ □

Note that Corollaries 1–4 also follow trivially from Theorem 3.

6. Generalizations

Though we have not worked out the details, it appears quite likely that the preceding results can be substantially generalized to include the case when A and B have multiple eigenvalues. For example, if the maximum multiplicity of ~~key step is to characterize~~ the eigenvalues of A and B is 2, then $E = \begin{pmatrix} E_1 & & 0 \\ & \ddots & \\ 0 & & E_m \end{pmatrix}$ where each $E_i = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$ or $E_i = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ where $0 \le \theta \le 2\pi$. Thus we may view E_i as a rotation or rotation and flip matrix. It is then easy to describe UE in a manner similar to that discussed in this paper.