QUANTUM THEORY PROJECT
FOR RESEARCH IN ATOMIC, MOLECULAR, AND SOLID STATE
CHEMISTRY AND PHYSICS
UNIVERSITY OF FLORIDA, GAINESVILLE, FLORIDA

A LISP FUNCTION FOR COMPUTING INDUCED REPRESENTATIONS

By

Billy S. Thomas*

PROGRAM NOTE #10

25 July 1963

*Department of Physics and Astronomy, University of Florida

ABSTRACT

Most algorithmic schemes for constructing irreducible representations of finite groups are based on the use of induced representations, induced from an appropriately chosen subgroup. In this note we present a LISP function designed for the symbolic calculation of an induced representation, given a group and a representation of its subgroup. The group may be specified either by its group table, or by a LISP function which yields the products and inverses of its elements.

ACKNOWLEDGMENTS

The author wishes to thank Harold V. McIntosh for his lucid series of lectures on the theory of induced representations, and Robert Yates for his patient explanations of the inner workings of his group theory functions.

Partial support by the Department of Physics and Astronomy of the University of Florida is acknowledged.

A LISP FUNCTION FOR COMPUTING INDUCED REPRESENTATIONS

For groups of large order, it is sometimes a difficult problem to determine the irreducible representations of the group. The theory of induced representations is useful in solving this problem, for with this theory we can show that any irreducible representation γ of any subgroup H of the group G can be used to construct a representation Γ of G. The representation Γ is irreducible provided the restriction of γ to the intersection of H with any of its conjugate subgroups does not cause it to become reducible. A representation Γ constructed according to this process is called an induced representation, and it is said to be induced by the representation γ of H.

The purposes of these notes are (1) to set forth a procedure (which is quite simple) for constructing the induced representation which is easy to program for a high speed computer, and (2) present a LISP function which will construct the induced representation from the representation γ of H. In preparation, we will first present, mostly without proof, enough of the theory of induced representations to make clear the reasoning behind the procedure.

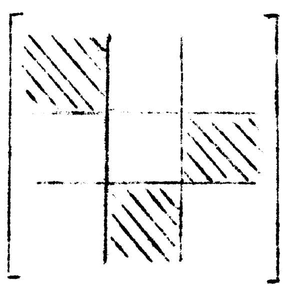
THEORY

The values of Γ at the group elements are nonsingular imprimitive transitive matrices; an imprimitive matrix is one which can be partitioned so as to have at most one non-zero submatrix in each row and column ("row" and "column" here refer to rows and columns of "blocks"into which the matrix has been partitioned); a transitive matrix is an imprimitive matrix such that any subspace V^i of the representation space V can be mapped onto any other subspace V^j of V (the relevant subspaces of V are defined by the partitioning of the imprimitive matrix to give it the "super permutation"form).

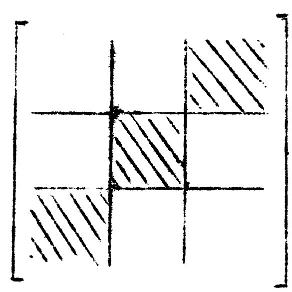
It can be proved without difficulty that the submatrices of the representing matrices are square, invertible, and all of the same dimension.

It follows from the definition that the product of two imprimitive matrices is agin an imprimitive matrix. As an example of this, let the matrices representing the group elements a and b be

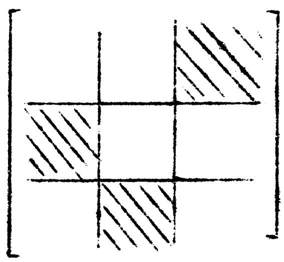
D(a) =



D(b) =



where the shaded blocks stand for the nonzero submatrices. Then by ordinary matrix multiplication, we see that the nonzero block structure of the product is

$D(a)D(b) \quad =$ 

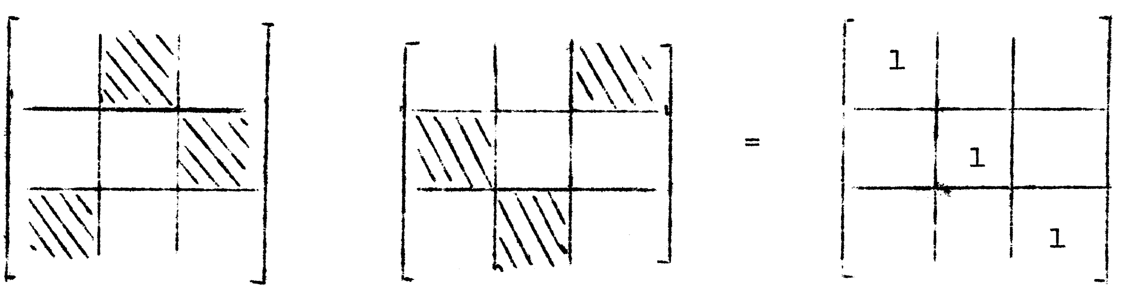
and since this is a representation, we have that

$$D(a)D(b) \quad = \quad D(ab) \quad .$$

The representing matrix of the identity element of the group is a unit matrix in any representation; thus it always has the imprimitive form.

Considering now the block structure of the equation

$$D(a)D(a^{-1}) \quad = \quad D(e) \quad ,$$



$D(a)$ $\qquad$ $D(a^{-1})$ $\qquad$ $D(e)$

we see that the nonzero block structure of $D(a^{-1})$ is the transpose of the block structure of $D(a)$. Furthermore, $D(a^{-1}) \;=\; D^{-1}(a)$ and, labeling the submatrices by (super) row and column indices, we see that $D_{ij}(a^{-1}) \;=\; D_{ji}(a))^{-1}$. In general we have

$$D_{ij}(a)D_{jk}(b) \quad = \quad D_{ik}(ab) \quad ,$$

since the rule is matrix multiplication for a representation--no sum appears because we have only one nonzero block per row and column.

Consider now the subset H of G such that $D_{ii}(h) \neq 0$, $h \in H$. This subset is a subgroup of G.

Proof:

(1) H is non-empty and contains e, for $D_{ii}(e) \neq 0$.

(2) H is closed, for

$$D_{ii}(h)D_{ii}(h') = D_{ii}(hh') \quad , \qquad h, h' \in H.$$

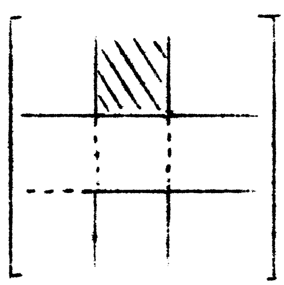
(3) Associativity is fulfilled by the nature of the matrix product.

(4) $h \in H$ implies $h^{-1} \in H$, for the block structure is transposed for inverses.

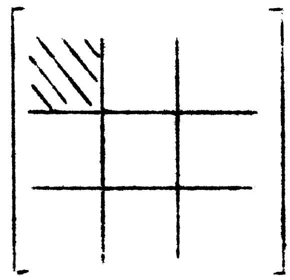
For H we introduce the notation $[V^i]$, which we may call "the <u>fixing group</u> of V^i, since it is the subgroup of G that keeps V^i fixed.

Let us now fix our attention on $[V^1]$, the subgroup that keeps the first subspace fixed. The matrix

D(a) =



(not saying where the other submatrices are, except of course they can't be in the first row or the column with the shaded block) is not in the set $[V^1]$, since it maps vectors into V^1. Upon left multiplying D(a) by $D(h) \in [V^1]$, where

$D(h) \quad =$ 

we see that the product $D(ha)$ has a nonzero submatrix in the same column of the first row as does $D(a)$. Since any element $h \in [V^1]$ could have been used, we see that all the elements of the right coset of $[V^1]$ which contains $D(a)$ have a nonzero submatrix in the same column of the first row. Thus the matrices of the different right cosets of $[V^1]$ are distinguished by having a nonzero submatrix in a different column of the first row. It should be noted that the number of cosets divides the order of G and the number of subspaces divides the order of the carrier space, so if the orders are incompatible, the group cannot act as an imprimitive transitive group on this space.

Let us now select a matrix from each right coset; each has a nonzero submatrix in a different column of the first row. Having selected these arbitrarily, we can now find a basis for V such that the nonzero submatrices in the first row of these generators are unit matrices. This is accomplished by letting the submatrix in $D(a)$, one of the selected matrices, be the image of the submatrix $D_{11}(e)$ under a similarity transformation.

$$D'(a) \quad = \quad UD(a)U^{-1} \quad ,$$

where

$$U = \left[\begin{array}{c|c|c} D_{11}(e) & & \\ \hline & D_{12}(a) & \\ \hline & & D_{13}(b) \end{array}\right]$$

for the case of only 3 right cosets. It follows that

$$U^{-1} = \left[\begin{array}{c|c|c} D_{11}(e) & & \\ \hline & D_{21}(a^{-1}) & \\ \hline & & D_{31}(b^{-1}) \end{array}\right]$$

since $(D_{ij}(a))^{-1} = D_{ji}(a^{-1})$, and under the transformation we have

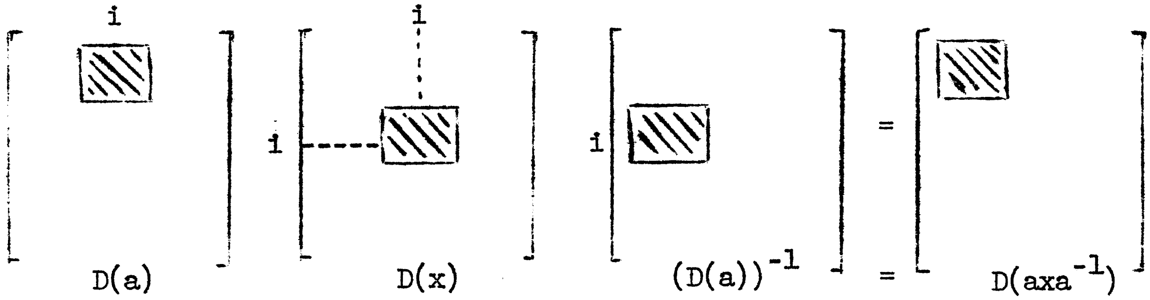
$$D'_{12}(a) = D_{11}(e)\, D_{12}(a)\, D_{21}(a^{-1}) = D_{11}(e) \quad ,$$

since $$D_{12}(a)\, D_{21}(a^{-1}) = D_{11}(aa^{-1}) = D_{11}(e) \quad .$$

In the same way $D_{13}(b)$ becomes $D_{11}(e)$, thus a basis has been found such that the representative of one element of each right coset of $[V^1]$ has a unit matrix in some column of the first row. From now on we may refer to these matrices having a unit submatrix in the first row as coset generators.

We now wish to show that selecting the first subspace as the one to be held fixed is not an essential part of the preceding argument, for we will now show that the representation of $[V^i]$, the subgroup that keeps V^i fixed, is equivalent to the representation of $[V^1]$.

Consider the equation



where $D_{1i}(a) \neq 0$, $D_{ii}(x) \neq 0$, $D_{i1}(a^{-1}) \neq 0$, $D_{11}(axa^{-1}) \neq 0$.

Doing the multiplication, we see that

$$D_{1i}(a)\, D_{ii}(x)\, D_{i1}(a^{-1}) = D_{11}(axa^{-1}) \quad ,$$

therefore the representation of $[V^i]$ is equivalent to the representation of $[V^1]$. Furthermore, we may select D(a) as the generator of the i^{th} coset; then $D_{1i}(a)\, D_{i1}(a^{-1})$ = unit matrix, and $D_{ii}(x)$ is equal to $D_{11}(axa^{-1})$, not merely equivalent to it.

The fact that

$$D_{11}(h)\, D_{11}(h') = D_{11}(hh') \quad , \qquad h, h' \in [V^1]$$

defines the D_{11} submatrices as a representation of $[V^1]$; thus if $\{D_{11}(x)\}$ means a representation of $[V^1]$, then $\{D_{ii}(a^{-1}xa)\}$ is a representation of $[V^i]$.

$\{D_{ii}(a^{-1}xa\}$ is called a conjugate representation to $\{D_{11}(x)\}$. We see that the fixing subgroups are not independent, in fact they are conjugate to one another. For our purposes, the most useful way to state this property is to note that it implies that $D_{ii}(x)$ is zero unless $axa^{-1} \in [V^1]$, in which case $D_{ii}(x) = D_{11}(axa^{-1})$. Having established this fact, we are now in a position to evaluate any submatrix of a representative of a group element. The reasoning is the same as that just given for the evaluation of a diagonal submatrix, for there is nothing special about the diagonal submatrices. Thus, we want to evaluate $D_{\eta\sigma}(a)$ where $a \in G$, and we agree that η and σ are the coset generators.

We write

$$D_{\eta\sigma}(a) \; D_{\eta\sigma}(\eta^{-1}\eta \; a \; \sigma^{-1}\sigma) \qquad ,$$

and since this is an imprimitive representation we have

$$D_{\eta\sigma}(\eta^{-1}\eta \; a \; \sigma^{-1}\sigma) \;=\; D_{\eta 1}(\eta^{-1}) \; D_{11}(\eta \; a \; \sigma^{-1}) \; D_{1\sigma}(\sigma) \quad .$$

But the first and last matrices on the right side are unit matrices, since η and σ are coset generators, therefore we have

$$D_{\eta\sigma}(a) \;=\; D_{11}(\eta \; a \; \sigma^{-1}) \quad .$$

Thus, the rule for calculating submatrices is that $D_{\eta\sigma}(a)$ is zero unless $\eta a\sigma^{-1} \in [V^{-1}]$, in which case it is equal to the submatrix $D_{11}(\eta a\sigma^{-1})$. This rule is clearly sufficient to construct the induced representation, given any representation of any subgroup of G.

PROCEDURE

Although the procedure for constructing induced representations has been given in great detail above, we now present a "cookbook" description with the idea of helping the reader to understand how the LISP functions for computing induced representations work.

To construct an induced representation of a group G we

(1) select an irreducible representation γ of a subgroup H of G

(it is necessary but not sufficient for the induced representation to be irreducible that γ be irreducible)

(2) select a cross section of the right cosets of H (the coset generators)

(3) form all products $\eta x \sigma^{-1}$ where $x \in H$ and η, σ are coset generators

(4) construct the matrix representing the element x of G by

(a) writing a zero submatrix at the intersection of the η^{th} row and σ^{th} column if $\eta x \sigma^{-1}$ is not a member of H

(b) otherwise writing the submatrix which is the value of γ at the group element $\eta x \sigma^{-1}$

(5) repeat step (4) for each element of G.

The matrices resulting from this process are the values of the representation Γ of G induced by the representation γ of H.

LISP FUNCTIONS

In describing the LISP functions, we will assume that the reader is familiar with Yates' set of group theory functions issued in program note #5, though we will at least define all functions needed, and give very brief descriptions of what they do.

The definition of the main function, INDREP, is

```
(INDREP (LAMBDA (D H) ((LAMBDA (X) (INDREP* (GELEMENTS) X
      (INVERSES X))) (XSECTION (RCOSETS H))))).
```

INDREP is a function of two variables, D and H. H is the subgroup selected to induce the representation, and D is a representation of H. The group of which we are computing an induced representation does not appear in the definition of INDREP; this is because Yates' group theory functions are arranged (for flexibility) so that the group occurs in them only as a free variable which must be bound to the desired group when the order to execute a function is given. Except for INDREP*, the other functions which appear in the definition of INDREP are from Yates' group theory package. Their definitions are given in the Appendix. Here is a brief description of what they do:

(GELEMENTS) produces a list of the group elements from the group table of G.

(INVERSES X) produces a list of the inverses, according to the group table, of the elements of the list X.

(RCOSETS H) lists the right cosets of the subgroup H of the group G.

(XSECTION L) selects one element of each right coset from the list of right cosets L, and forms a list of these elements.

The main function, INDREP, is immediately redefined in terms of an auxiliary function INDREP* which will do the actual manipulation of symbols which is required. The purpose of this redefinition is to prevent the repeated calculation of GELEMENTS, RCOSETS, etc. In other words, INDREP*, given three lists of the correct form as its arguments,will carry out a standardized operation on the list elements. At the time of execution, we will give it three lists which are the <u>values</u> of GELEMENTS, etc.

The definition of INDREP* is

```
(INDREP* (LAMBDA (G* C I) (IF (NULL G*) G* ((LAMBDA (G)
   (CONS (OPBICARTESIAN C I) (INDREP* (CDR G*) C I)))
      (FUNCTION (LAMBDA (X Y) (D MULTIGP X (CAR G*) Y)))))))).
```

Here (MULTIGP L) is again one of Yates' functions. It forms the product, according to the group table, of any number of group elements, in the order given. (OPBICARTESIAN C I) was defined with the application to induced representations in mind, but it is likely to be useful in other contexts, since its definition includes, as a free variable, a function of two variables, which function can be bound to any function desired. The definition of OPBICARTESIAN is

```
(OPBICARTESIAN (LAMBDA (U V) (IF (NULL U) U (OPBICARTESIAN* V))))
(OPBICARTESIAN* (LAMBDA (V*) (IF (NULL V*) (OPBICARTESIAN (CDR U) V)
   (CONS (G (CAR U) (CAR V*)) (OPBICARTESIAN* (CDR V*)))))).
```

We see that, by switching the program control back and forth between themselves, these two functions generate a list, the elements of which are the results of applying G to every possible pair of elements which can be formed by selecting one element from U and one from V. In the application here, G is bound to D; that is the purpose of the LAMBDA statement in INDREP*. Finally, since D is a variable, it must be bound to a function in the APPLY statement.

As a complete example, we will use the group D3, the symmetry group of the triangle. The group table is a function, defined as

```
(D3 (LAMBDA ( ) (QUOTE (E (E E A A A2 A2 R R RA RA RA2 RA2) A
   (E A A A2 A2 E R RA2 RA R RA2 RA) A2 (E A2 A E A2 A R RA RA
      RA2 RA2 R) R (E R A RA A2 RA2 R E RA A RA2 A2)
RA (E RA A RA2 A2 R R A2 RA E RA2 A) RA2 (E RA2 A R A2 RA R A RA
      A2 RA2 E)))))
```

The symbols in the group table have the following meanings:

E unit

A clockwise rotation through 120°.

A2 clockwise rotation through 240°.

R reflection about the perpendicular bisector of the horizontal side of the triangle.

RA reflection about the perpendicular bisector which has negative slope.

RA2 reflection about the perpendicular bisector which has positive slope.

For the inducing subgroup, we will use H = (E,R). This group has two irreducible representations, the identity representation, and the representation E = 1, R = -1. We will use the latter; accordingly we bind the variable D to the function D1, defined as

```
(D1 (LAMBDA (X) (IF (ELEMENT X H) (IF (EQ X (QUOTE E)) (QUOTE 1)
    (QUOTE -1)) (QUOTE 0))))
```

This function has the required "semizero" property; it gives zero if X is not in H, and one or minus one according as X is E or R. To complete the example, we write the APPLY statement, binding T (of Yates' functions) to D3, H to (E,R) and D to D1:

```
(APPLY (LAMBDA ( ) ((LAMBDA (T) (INDREP (QUOTE D1) (QUOTE (E R))))
      (D3))) ( ))
```

This causes INDREP to be executed. The result is (interpreted as)

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad A2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad RA = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad RA2 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix} .$$

The actual computer output is

((1 0 0 0 1 0 0 0 1) (0 1 0 0 0 1 1 0 0) (0 0 1 1 0 0 0 1 0)
(-1 0 0 0 0 -1 0 -1 0) (0 -1 0 -1 0 0 0 0 -1) (0 0 -1 0 -1 0 -1 0 0)).

APPENDIX

We define here all the functions needed to compute induced representations, except INDREP, INDREP*, OPBICARTESIAN and OPBICARTESIAN*, which were defined in the text.

```
(UNIT (LAMBDA ( ) (QUOTE E)))

(GP* ( LAMBDA (T) (ASSOC Y (ASSOC X T))))

(GP (LAMBDA ( X Y) (GP* T)))

(GELEMENTS* (LAMBDA (T) (IF (NULL T) T (CONS (CAR T) (GELEMENTS* (CDDR T))))))

(GELEMENTS (LAMBDA ( ) (GELEMENTS* T)))

(A-1* (LAMBDA (T) (ASSOC* (UNIT (ASSOC A T)))))

(A-1 (LAMBDA (A) (A-1* T)))

(INVERSES (LAMBDA (L) (IF (NULL L) L (CONS (A-1 (CAR L))
        (INVERSES (CDR L))))))

(XSECTION (LAMBDA (L) (IF (NULL L) L (CONS (CAAR L) (XSECTION (CDR L))))))

(RGP (LAMBDA (X L) (IF (NULL L) L (CONS (GP (CAR L) X) (RGP X (CDR L))))))

(EQRELATION (LAMBDA (R L) (IF (NULL L) L ((LAMBDA (W) (CONS W
       (EQRELATION R (ERASE W L)))) (R (CAR L))))))

(ERASE (LAMBDA (L M) (IF (NULL L) M (ERASE (CDR L) (REMOVE (CAR L) M)))))

(REMOVE (LAMBDA (X L) (COND ((NULL L) L) ((EQUAL X (CAR L)) (CDR L))
      ((AND) (CONS (CAR L) (REMOVE X (CDR L)))))))

(EQUAL (LAMBDA (X Y) (OR (EQ X Y) (AND (NULL X) (NULL Y)) (AND (NOT
    (OR (NULL X) (NULL Y) (ATOM X) (ATOM Y))) (EQUAL (CAR X) (CAR Y))
        (EQUAL (CDR X) (CDR Y))))))

(RCOSETS* (LAMBDA (G) (EQRELATION (FUNCTION (LAMBDA (X) (RGP X S))) G)))

(RCOSETS (LAMBDA (S) (RCOSETS* (GELEMENTS))))
```

```
(MONOGP (LAMBDA (L) (GP (CAR L) (IF (NULL (CDDR L)) (CADR L)
    (MONOGP (CDR L))))))

(MULTIGP (LAMBDA L (MONOGP L)))
```

This completes the list of functions which are strictly part of the induced representations package. One is fairly likely to want to use ELEMENT in defining the "semizero" function to which D must be bound (see the example D1 in the text), so we define it here:

```
(ELEMENT (LAMBDA (X L) (AND (NOT (NULL L)) (OR (EQUAL X (CAR L))
   (ELEMENT X (CDR L))))))
```

Finally, the following four functions, which are normally predefined, must be defined externally with the present processor.

```
(FUNCTION (LAMBDA* (L) (LIST (QUOTE FUNARG) L (CDDR (ALIST)))))

(FUNARG (EVAL (CONS (CADAR SYSVAR1) (EVAPPQ (CDR SYSVAR1) SYSVAR2))
    (CADDAR SYSVAR1)))

(EVAPPQ (LAMBDA (E ALIST) (IF (NULL E) E (CONS (LIST (QUOTE QUOTE)
    (EVAL (CAR E) ALIST)) (EVAPPQ (CDR E) ALIST)))))

(LIST (LAMBDA L L))
```