MSUCL-524

# OXBASH (Nov 1986)

## The Oxford-Buenos-Aires-MSU Shell-Model Code

| | |
|---|---|
| OXBASH82 | : A.Etchegoyen, W.D.M.Rae and N.S.Godwin |
| | : contributors: W.A.Richter and C.H.Zimmerman |
| MSU version | : B.A.Brown, W.E.Ormand and J.S.Winfield |


B. A. Brown
Cyclotron laboratory, Michigan State University,
East Lansing, Michigan 48824-1321, USA

A. Etchegoyen
Comision Nacional de Energia Atomica, Departamento de Fisica
Avenida del Libertador 8250, 1429 Buenos Aires, Argentina

W. D. M. Rae
Nuclear Physics Laboratory, Oxford University,
Keble Road, Oxford, OX1-3RH, England



MSUCL Report Number 524

Everybody is welcome to use this code. Errors and modifications should be communicated to the authors. Changes in the author list and/or in the name of the code (total or by partial addition of names) are possible only with the written confirmation of the authors.

------- Appendices ----------

A1: The LOGIN.COM file
A2: The LOGIN.SHL file
A3: The LFALL.COM file which compiles and links all OXBASH programs
A4: The LFBAB.COM file which compiles and links all special programs written by BAB (see section M9)
A5: The ALL.SIZ file obtained from the /MAP option in LINK
A6: The PARAM.HLP file explains the size parameters used in PARAM.INC
A7: The LABEL.DAT file which contains the *.SPS and *.INT codes and references
A8: The LABELJ.DAT file which contains the integer codes
A9: Listing of all files

B: Sample calculation for one-body transition density for the $^{20}$Ne $0^+$ to lowest five $2^+$ states.
B1: The input to SHELL
B2: The *.COM file generated by SHELL
B3: The *.LOG file from the batch job
B4: The *.LPE eigenvalue output from LANCZOS
B5: The *.LRD transition density output from TRAMP
B6: The *.OBD transition density output from RTRD

M0: Getting started

1) In order to read the programs and data files from the tape you will need an area on the VAX with at least 10000 blocks. If you do not need to recompile the programs (see below) then the minimum needed to do calculations are the execution files on the area "SHEXE" (about 2000 blocks) and data files on the area "SHSPS" (about 2000 blocks). If it is necessary to recompile the programs you will need to increase space allocation temporarily to about 20000 blocks. "RSH:" should be assigned to the area where the programs are to be run.

2) A backup tape has been made by the command file BACKSH.COM which contains:

```
$INITIALIZE MFA0:/DENSITY=1600 BACKUP
$MOUNT/FOR MFA0:/DENSITY=1600 BACKUP
$BACKUP/LOG [MYNAME...]*.*.* MFA0:SHELL
```

where MYNAME depends upon who made the tape. At Michigan State "MYNAME"="BROWN.SHELLGROU".

In order to read the tape type:

```
$MOUNT/FOR MFA0:/DENSITY=1600 BACKUP
$BACKUP/LOG_MFA0:SHELL/SELECT=([MYNAME...]*.*)_
[YOURNAME...]*.*.0
```

where YOURNAME is your new area name. YOURNAME itself may include another subdirectory level, for example "MAIN.SUB".

3) In the the subdirectory [YOURNAME] the LOGIN.COM file at present reads:

```
$@MYDISK[MYNAME]LOGIN.SHL MYNAME MYDISK
```

At Michigan State "MYDISK"="SYS$THEORY:". Modify this to read:

```
$@YOURDISK[YOURNAME]LOGIN.SHL YOURNAME YOURDISK
```

If you have only one disk you can leave out YOURDISK, i.e. just $@[YOURNAME]LOGIN.SHL YOURNAME.
Also, in the second line of file LFALL.COM, change MYNAME to YOURNAME and MYDISK to YOURDISK.

4) Then type "@LOGIN.COM". You are now ready to use all of the shorthand commands in the LOGIN.SHL file. (This

LOGIN.COM should eventually be added to your main LOGIN.COM file.) For example, "SD SHDIR" is the shorthand for "SET DEF [YOURNAME]" and "SD RSH" is the shorthand for "SET DEF [YOURNAME.RSH]". "SHELL" is the shorthand for "RU [YOURNAME.EXE]SHELL."

5) Next type "SHOW PROC/ALL" and look at the number under "Paging file quota" (My present quota is 32235). Then look at the table in the file ALLS.SIZ (Appendix A5) to see if any of the "virtual memory allocated" sizes are larger than your quota. If so, first try to get your system manager to increase your quota.

Part of the proceedure to increase the quota involves the following for the system manager:

```
$RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SHO VIRT (this shows the current default value)
SYSGEN> SET VIRT 40000 (some number larger than 32235)
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

then reboot the SYSTEM again

If you cannot do this you will need to decrease one of the size parameters in the file PARAM.INC (for example MAX_MSCHEME) in the main area (SHGR) (see the PARAM.HLP file). For TRAMP the parameter MAX_STORE in TRPCOM.INC on area SHTR: may also be reduced, even to zero. Generally there is a trade off in TRAMP between time and size via MAX_STORE. (With MAX_STORE=0, TRAMP takes about 3500 pages in the link.)

6) If the "paging file quota" criteria are met you should be able to run the compiled programs (*.EXE) from the tape. If so, skip (7) and go to (8).

7) To recompile the programs type @LFALL or submit this to the batch queue. This will compile and link all of the programs and put the "size" information into the file ALL.SIZ. This takes about 25 minutes on the VAX780.

If the batch-job stops with the message:

```
%LINK-E-MEMFUL, insufficient virtual address space to
complete this link -LINK-E-NOIMGFIL, image file not created
```

the failure to link was probably because the "virtual memory allocated" during the link was greater than your quota. See the file ALL.SIZ to get this number or look at

the *.MAP file. In this case you must further reduce the value of one of the parameters in PARAM.INC or in the other *COM.INC files.

Then repeat the LFALL.COM job.

8) Next look at the LABEL.DAT file on area "SHSPS" and choose a model-space (*.SPS) name and an interaction (*.INT) name. (In the example given in appendix B, the model-space name is "SD" and the interaction name is "W".) Then type:

```
SET DEF RSH
SHELL
```

and answer the questions (see the example in appendix B). Then run the batch job.

9) VAX780 time and space requirements:

Listed below are "typical" times and spaces which should be good to within a factor of 2-3 depending on the model-space and particular J-T value. Times on the VAX750 are typically a factor of 2 longer.

| J-T dimension | time hr:min:sec | space blocks |
|---|---|---|
| 10 | 0:00:30 | 100 |
| 100 | 0:10:00 | 500 |
| 500 | 1:00:00 | 5000 |
| 1000 | 3:00:00 | 20000 |
| 2500 | 10:00:00 | 50000 |
| >3000 | untested | |

10) What to keep when finished:

When the calculation is finished it is usual to save some of the final output files (e.g. the formatted *.LPE and unformatted *.EIG eigenvector files and the formatted *.L%% and unformatted *.T%% transition density files) on another subdirectory and then delete all files in the RSH area. SHELL also stores the *.OP data files on area SHOP and the *.OP2 data files on the area SHOP2. These can also be deleted when the calculation is done, but some time can be saved if these files do not have to be regenerated.

M1: Introduction

A new shell-model code has been written for VAX11 computers. It works in the occupation number representation, where the occupancy (vacancy) of a bit in any given position of the computer word symbolises the presence (absence) of a particle in a specific single particle state (i.e. in a given $|n,\ell,j,mj,t_z>$ state). As there might be, in this representation, a maximum limit to the number of single particle states allowed in each calculation given by the number of bits per longword (32 for VAX11 systems), the programs generate and work with an m-scheme basis set in which any given basis state is a vector of 1,2,... "longwords" (for 32,64,... single particle states).

The codes use an m-scheme Slater determinant basis, which is generated for a particular number of particles and a particular total $J_z$ and $T_z$ by the program 'BASIS'. The program 'PROJ' will construct basis states with good spin ($J \geq J_z$) and isospin ($T \geq T_z$). This is done by a projection technique. Given this projected basis, the program 'MATRIX' constructs the Hamiltonian matrix. The program 'LANCZOS' is then used to diaganolise the matrix and find the wave functions and eigenvalues. Finally, the wave functions can be used to calculate the following quantities by running the program 'TRAMP':

i) Matrix elements of $a^+$ and $a^+a^+$ (1- and 2- particle parentange amplitudes),

ii) Matrix elements of $a^+a$ and $a^+a^+aa$ (1- and 2- body transition densities),

iii) Direct overlaps of two wave functions,

iv) Cluster overlaps involving three wave functions.

Many of the operations performed in this code do not promote particles from one J-orbit into another. This symmetry is referred as a 'partition symmetry' and is used extensively throughout the coding. The partition (configuration) of a basis vector is a multi-precision integer, each byte of which set to the number of particles in an specific J-orbit. Thus, the partition of an m-scheme basis vector tells how the total number of particles is split in the different J-orbits. Matrix elements will be non zero only when the sum of the partition number of the vector to which the operator is applied plus the partition

number of the operator itself equals the partition number of the remaining vector (obviously, for direct overlaps of two m-scheme vectors, both vectors have to belong to the same partition). The partition symmetry greatly reduces both the CPU time and the core needed for the calculations.

## M2: Brief Review of the Programs

In this section a brief review of the programs will be presented. The package consists of seven main computer codes: 'BASIS', 'PREDICT', 'PROJ', 'MATRIX', 'LANCZOS', 'MVEC' and 'TRAMP' and some subsidiary codes: 'SHELL', 'OPER', 'HCM', 'CHANGE', 'LEVEL', 'TRANS' and 'TBTDOP'. The source programs are able to work with 1,2,3,... longwords for each m-scheme vector thus allowing 32,64,96,... single particle states $|n\ell jj_z t_z>$. It is emphasised that the programs do not have separate common blocks. The common block has been written in separate files and they are included at the beginning of each routine. The dimensions have been defined with a parameter statement which should be the only necessary change if dimensions are changed.

What the multiple precision version essentially does is to replace each basis state handling instruction by a call to a relevant MACRO subroutine which performs a loop over the number of longwords.

A calculation would generally commence by running 'SHELL'. This program generates a batch command file with all the relevant information needed for a complete calculation.

'BASIS' generates an m-scheme Slater determinant basis for a particular number of particles and a particular parity for some total $J_z$ and $T_z$. The single-particle states for the calculation are contained in a data file with the extension '.SPS', and restrictions can be placed in both individual J-orbits and major shells.

The program 'PREDICT' is then run to produce a subset out of the above mentioned basis set. This subset is predicted to generate a good J-basis when used by the program 'PROJ' (as will be explained later, 'PREDICT' can only be used for a $J_z$=J basis).

The program 'PROJ' is then run to construct basis states with good spin ($J \geqq J_z$) and isospin ($T \geqq T_z$).

This is done by a projection technique. 'POSTPROJ' projects from a given $J_zT_z$ state and then orthonormalises the wave function to the space generated by all previous good JT-states. On the other hand, 'PRIORPROJ', projects from a linear combination of $J_zT_z$ states so as to finish the projection with an already orthogonal state. As will be discussed later 'POSTPROJ' is recommended for calculations in the isospin formalism while 'PRIORPROJ' is recommended for the neutron/proton formalism. For very large calculations 'POSTPROJ' needs to write wave functions on disk when it exceeds a set memory quota and thus 'PRIORPROJ' might become a recommendable alternative. Both programs will have similar performaces for any calculation not exceeding half an hour, approximately.

The program 'MATRIX' reads the output files of 'PROJ' and constructs the Hamiltonian matrix for the case under consideration. An operator file (with extension 'OP') is also required as an input containing the single particle energies and the two-body m-scheme matrix elements. These m-scheme matrix elements can be obtained by running 'OPER' which reads an input data file with extention 'INT' which contains the two-body JT-scheme matrix elements.

After 'MATRIX', the program 'LANCZOS' is run. The Lanczos algorithm efficiently finds the lowest few eigenvalues and eigenvectors of very large matricies. It was first used in shell-model calculations by the Glasgow group for their large m-scheme matricies and was then adapted to the Michigan State version of the Oak Ridge-Rochester shell-model code. We have obtained this latter version (from B.H.Wildenthal) and adapted it to the present shell-model code. 'LANCZOS' allows a change in the single particle energies. The eigenvalues, which are output by this code, for different runs (i.e. different spin, isospin and/or parity) can be arithmethically ordered by 'LEVEL'.

The eigenvectors calculated by 'LANCZOS' are input to the program 'MVEC' which rewrites them in terms of the m-scheme basis. This output file is needed by 'TRAMP' for all further calculations, that is, to calculate parentage amplitudes and one- or two-body transition densities. In order to calculate two-body transition densities, a file containig the m-scheme operator is needed in the same fashion as above explained with the Hamiltonian in 'MATRIX'. This operator file is produced by the program 'TBTDOP' which requires no input data file.

The program 'TRANS' can be used to calculate electromagnetic moments, occupation numbers, transition rates and Gamow-Teller beta decay using the one-body transition density files with extension 'TRD' from the output of 'TRAMP' and harmonic oscillator radial wave functions.

Program 'HCM' constructs the Hamiltonian for center of mass motion and its output is in the form of an interaction file to be read by 'OPER'. The program 'SHELL' creates a command file which automatically adds the CM interaction in the appropriate model space.

## M3: Running SHELL and format of the file names

The calculation will generally start by running 'SHELL'. Generally there is a default on answers to questions so that, when uncertain of the answer required, <CR> usually works. Most of the information required by 'SHELL' is straightforward and is best illustrated by examples (see Appendix B). As should be clear from the introduction each shell-model calculation consists of a number of stages and more than one program has to be run. Most programs generate at least one disk file which is accessed by other programs in the sequence. The disk files are distinguished by their name, which have different extensions; each extension is characteristic of the program used for its generation. It is simplest if the first part of each filename is the same for a particular calculation (i.e. 'BASIS' through 'LANCZOS') and suitable names are automatically generated by 'SHELL'. These consist of six characters and can be decoded as follows:

```
first  character = model-space code
second character = 2*J value, with A,B,C,... representing
                                              J=5,5.5,6,...
third  character = 2*T value, with A,B,C,... representing
                                              T=5,5.5,6,...
fourth character = 0(1) for positive(negative) parity
fifth  character = number of valence particles
sixth  character = Hamiltonian interaction code
```

The first and sixth characters are defined in the file 'LABEL.DAT' (see appendix A) which can be readily changed to include new possibilities.

The code for 2*J and 2*T is contained in the file LABELJ.DAT (see appendix A). The fourth line consists of

the code numbers used to generate second, third and fifth characters in the filename. The '!' symbol is an end of line code. (The first three lines of this file are no longer used by SHELL. They were once used to generate filenames when either $J_z$ is different from J or $T_z$ is different from T. The first line is the character associated with the 2*J (2*T) value of the second line and with the $2*J_z$ ($2*T_z$) value of the third line.)

The output files of the different programs are:

| PROGRAM | ---------- OUTPUT FILE ---------- | |
|---|---|---|
| | UNFORMATTED | FORMATTED |
| BASIS | X.BAS,X.INF | X.LPB |
| PREDICT | X.PRD | X.LPD |
| PROJ | X.PRJ,X.RED | X.LPP |
| MATRIX | X.MAT | X.LPM |
| LANCZOS | X.EIG | X.LPE |
| MVEC | X.VEC,X.INF | |
| TRAMP | Y.TUV | Y.LUV |

'X' is the six character name as explained above.

'Y' is a nine character label which is the same as 'X' for the initial state plus three characters (second, third and fourth) from the final state.

'UV' is the extension on the 'TRAMP' output files which can be:

- 1N : for the one-nucleon spectroscopic factors
- 2N : for the two-nucleon spectroscopic factors
- CC : for the core-cluster overlaps
- OL : for the direct overlaps
- RD : for the one-body transition densities
- BD : for the iso-scalar two-body transition densities
- B1 : for the iso-vector two-body transition densities
- B2 : for the iso-tensor two-body transition densities

In 'SHELL' mistakes can be rectified by answering '-1' to some of the questions, e.g. to "MIN, MAX J (2F) :", since the program will then return to the question "Option file. 'SHELL' will indicate when instructions are written to the command file by listing to the terminal the names of the relevant programs and the names of the data files, e.g., "C0206C: BASIS PRD PROJ MATRIX LANCZOS", this list indicates that 'BASIS', 'PREDICT', 'PROJ', 'MATRIX', and

'LANCZOS' will be run to obtain J=0, T=1, positive parity, six particle wave functions.

In 'SHELL' all basis are made with $J_z$=J and $T_z$=T. In general, this is the best way to perform the calculation, but when overlaps are to be calculated with 'TRAMP', $J_z$ and $T_z$ must be equal on both sides of the overlap. Hence it is sometimes necessary to perform calculations with $J_z$ or $T_z$ different from J and T, respectively. This can be done by editing the *.COM output of SHELL.

When 'SHELL' prompts: "Filename for the ground state", a reference energy filename for a particular nucleus should be input. This ground state file with extension 'GND' consists only in one number: the reference energy which will be substracted from all other energies and will, in general, be the lowest energy for a set of calculations. An answer of <CR> will take the lowest J-T value for the ground state energy and create a *.GND file for this state. If the desired ground state is not the one with the lowest J-T value, one should run SHELL first for the desired state so as to create this ground state file which will be accessed by the following calculations.

## M4: About BASIS

This section describes BASIS as it existed in Nov. 1983. In Feb. 1984 some modifications were made by J. Winfield to make it faster. These changes will be described in a future version of this manual.

The calculation is carried out one partition at a time once all possible partitions (occupancy of each J-orbit) are worked out by WOTPTS. Both J-orbit and major shell restrictions only need be considered when working out the partitions: simply those partitions that do not comply with the restrictions are discarded.

The innermost subroutine is NEWPAT (NEW PATtern), which constructs a new basis pattern from the pattern entered to the subroutine. What NEWPAT 'essentially' does is search for the first '01' occurrence in the bit pattern and swap it over to '10'. NEWPAT is called by the main program BASIS within a do-loop over the partitions. Therefore, all new patterns with a partition are worked out first and, only on completion the partition is changed. The process continues up to the last partition. NEWPAT generates the basis states in arithmetic order. It only needs, in each call, to swap particles (bits) in one J-orbit. Bits to the left-hand side of this J-orbit remain untouched and bits to the right-hand side are zeroed and the relevant J-orbits are filled with their corresponding minimum patterns. Each minimum pattern has zeros in all J-orbits but one, in which it has all bits placed to the right-hand side.

'BASIS' finishes by reordering the basis vectors according to their partition and by writing them onto disk, partition by partition, starting with the partition with largest m-scheme dimension following in decreasing arithmetic order. Basis patterns are written for the 'FIND' subroutines, that is, subroutines that find whether a basis vector belongs to the basis set.

### General information on basis

i) The programs work in an all-particle scheme rather than in a hole-particle scheme. This clears up the FORTRAN allowing all bits to be treated in a similar fashion, it opens up a consistent method of working with more than two major shells and it avoids unnecessary masking and de-masking of hole patterns in later stages (PROJ, MATRIX,...).

ii) The test of the existence of a bit in a give position I, in a basis pattern is performed with the system library subroutine LIB FFS. This subroutine will just test bit I, rather than all the bits.

iii) If ILPT is set to 25 (IBAS=25) when running basis all the good (trial) basis states will be printed (typed) out in the lineprinter file. These facilities are there mainly to debugg the program and should not be used (unless you know what you want!). If ILPT<0, NEWPAT will be run even if there were restrictions. IBAS=25 can be used to type out the trial patterns. If IBAS=26, MAXMIN will type out the maximum and minimum patterns and some further information.

iv) The basis states are ordered and searched for in arithmetic order and the last bit in each (long)word is equivalent to the other bits and does not hold the sign.

'BASIS' generates two unformatted files with extension 'BAS' and 'INF'. 'BAS' file contains the m-scheme basis and 'INF' contains general information to be used by subsequent programs.

M5: About PREDICT

This computer code only works for $J_z$=J. By applying $J^+$ operator, it predicts a set of m-scheme vectors that if used for projection will produce a good J-basis. The treatment that follows can not be generalised for both spin and isospin to predict exactly a number of m-scheme vectors equal to the good JT-basis dimension. Nevertheless there is a $T_z$=T prediction facility which, however, does not substantially reduce the number of m-scheme vectors in the predicted set. This facility goes through the same procedure as for $J_z$=J prediction but for $T^+$ equations. Unfortunately, when we go through the elimation procedure we must eliminate firstly those states already eliminated by the $J^+$ procedure and generally we end up with a no significant reduction in the number of vectors of the predicted set. The $T^+$ procedure is therefore left optional and not applied by default.

We shall use the convention of capital letters for good J and/or good T basis vector and small letters for m-scheme vectors. Kets $|i\rangle$ will have total angular momentum z-projection equal to $J_z$, while kets $|j\rangle$ will have $J_z$+1 values. The predicted set will be refered as $|i_p\rangle$, the

number of states $|j>$ is $N(J_z+1)$, the number of states $|i>$ is $N(J_z)$ and the number of states with good J is N(J) so;

$$N(J) = N(J_z) - N(J_z+1) \qquad (1)$$

We will need to set up $N(J_z+1)$ equations for $N(J_z)$ unknowns so as to obtain the N(J) independent solutions. We search for a set $|i_p>$ so that if we apply to it a projection operator $P^J$ we obtain a set of good J vectors to form a (not necessarily orthogonal) basis. Initially we consider all possible $P^J|i>$, $i=1,\ldots,N(J_z)$ and our aim is to reduce it to a set $P^J|i_p>$ by examining linear equations of the form;

$$\Sigma\ \alpha_i\ \ P^J|i> = 0 \qquad (2)$$

To make it clearer, assume that only three $\alpha_i$ are different from zero e.g,

$$\alpha_1\ P^J|1> + \alpha_7\ P^J|7> + \alpha_{20}\ P^J|20> = \ 0 \qquad (3)$$

we can neglet $P^J|20>$ as linearly dependent on $P^J|1>$ and $P^J|7>$. By setting $N(J_z+1)$ equations we could in principle end up with a set of N(J) $|i_p>$ states. Our aim is to see which parameters $\alpha$ are different from zero in each equation rather than to obtain their values. Thus, the linear equations are logically solved rather than analytically solved.

The ladder operators are defined as;

$$J^+ = \sum_{part} j^+_i \qquad (4)$$

$$J^- = \sum_{part} j^-_i \qquad (5)$$

where small letter ladder operators act only on particle 'i'.

Consider a general vector with good J (ignoring T temporarily) and expand over the m-scheme basis:

$$|J_m> = \Sigma\ c^m_i\ |i> \qquad (6)$$

apply $J^+$

$$J^+|J_m> = \Sigma\ c^m_i\ J^+|i> = 0 \qquad (7)$$

and consider now a state $<j|$ with $J_z+1$ projection value;

$$\langle j|J^+|i\rangle = 0 \qquad (8)$$

apply $J^-$ to $\langle j|$;

$$J^-|j\rangle = \Sigma\ \alpha_i\ |i\rangle, \quad \alpha_i = \langle i|J^-|j\rangle \qquad (9)$$

replacing ( 9) in ( 8);

$$\Sigma\ c^m{}_i\ \alpha_i = 0 \qquad (10)$$

although the summation in ( 9) goes through the m-scheme dimension for a particular partition, no more than a number equal to the number of particles NPTCL of $\alpha_i$ will be non zero so the linear equations are short. This is seen as $J^-$ is a sum of NPTCL operators in ( 5) and some $j^-|j\rangle$ operations might be zero if the operation changes the partition, that is if a particle is tried to be transferred into a different $J^-$orbit. Now expand

$$P^J|i\rangle = \Sigma\ \beta_m\ |J_m\rangle \qquad (11)$$

$$\beta_m = \langle i|P^J|J_m\rangle = \langle i|J_m\rangle = c^m{}_i \qquad (12)$$

(see ( 6)) and replace (12) in (11),

$$P^J|i\rangle = \Sigma\ c^m{}_i\ |J_m\rangle, \text{ for all } i=1,2,\ldots,N(J_z) \qquad (13)$$

(see ( 6)). Consider a linear combination with the $\alpha_i$-coefficient as defined in ( 9);

$$\Sigma\ \alpha_i\ P^J\ |i\rangle \qquad (14)$$

replacing (13) in (14) and using (10);

$$\Sigma\ |J_m\rangle\ (\Sigma\ \alpha_i\ c^m{}_i) = 0 \qquad (15)$$

so the linear combination defined in (14) satifies;

$$\Sigma\ \alpha_i\ P^J|i\rangle = 0 \qquad (16)$$

with the $\alpha_i$-coefficients as defined by (8) and therefore the sum consists in a maximum of NPTCL terms. This is what we set off to do and an inspection of linear equations (15) gives the desired $|i_p\rangle$ set. We inspect the first equation and set to 'LD' (i.e. to linearly dependant) the largest basis vector and proceed with following equations in a similar fashion. If for each equation we can set a state to 'LD' we will be left with N(J) 'LI' (linearly independant)

states. But sometimes some equations do not allow any state to be set 'LD' (e.g. if all set to 'LD' already) then we finish with more than N(J) 'could-be-LI' states.

There will be in general many different sets of 'LI' states - a particular set is only defined by some arbitrary elimination procedure like, for instance, picking up the arithmetically largest basis vector.

The program 'JMJPPRED' generates $|J_z+1\rangle$ m-scheme vectors by applying $J^+$ and checks if a particular $|j\rangle$ state has already been made.For this, it uses $|J_z+1\rangle$ basis (as 'PROJ' also does) and this is a reason for running 'BASIS' twice: once to generate $|i\rangle$ basis set and a second time to generate $|j\rangle$ basis set. Once it find $|j\rangle$ it looks into a logical array to see whether the pattern has already been made. If not it proceeds to apply $J^-$ to the state and so generates one equation for each of the different $N(J_z+1)$ $|j\rangle$ states. So, 'JMJPPRED' generates the linear equations and stores the indexes (not the coefficients themselves) of those $|i\rangle$ basis vectors which have $\alpha_i$-coefficients different from zero. The logical solution (setting to 'LD' procedure) is performed in 'SOLVJT'.

M6: About PROJ

TWO METHODS OF PROJECTION ARE DESCRIBED BELOW - POSTPROJ AND PRIORPROJ. NOTE THAT THE MSU VERSION USES ONLY THE POSTPROJ VERSION. THE FORTRAN PROGRAM AND SUBDIRECTORY NAMES FOR THIS VERSION ARE CALLED SIMPLY "PROJ".

One disadvantage of an m-scheme basis is that it is much larger than the corresponding basis consisting of wave functions coupled to J and T. The neutron/proton formalism enters naturally in the m-scheme formalism, since it only needs to skip those unwanted $t_z$ values in each J-orbit in the corresponding SPS file (Single Particle State file). No useless information for other T-values in neither calculated not stored. Since the nuclear Hamiltonian commutes with J, and to a good approximation with T, the smaller basis can be used for the Hamiltonian matrix. For this reason a projection operator $P^{JT}$, where

$$P^{JT} = P^J P^T \tag{17}$$

with

$$P^J = \prod_{\substack{J'=J_z \\ J' \neq J}}^{Jmax} \frac{[J^{op}]^2 - J'(J'+1)}{J(J+1) - J'(J'+1)} \tag{18}$$

and

$$P^T = \text{as above with T replacing J,} \tag{19}$$

is applied to the m-scheme basis states in 'PROJ' in order to generate a new basis with good quantum numbers J and T. In practice, it is best, if possible, to use an m-scheme basis where $J_z$=J and $T_z$=T, since this requires the least number of projections. The operations $[J^{op}]^2$ and $[T^{op}]^2$ can be performed by using raising and lowering operators, since

$$[J^{op}]^2 = J^- J^+ + J^2_z + J_z \tag{20}$$

where

$$J^+ = \sum_{part} j^+_i \tag{21}$$

and

$$J^- = \sum_{part} j^-_i \tag{22}$$

and $j^+$, $j^-$ are the single-particle angular momentum raising and lowering operators; and similar relations hold for $T^{op}$. These operators are easily applied to the basis

wave functions by using simple computer bit-handling routines. The operation of $P^{JT}$ does not shift a particular basis state out of its partitions and therefore, only one basis partition is required in the computer core memory at a given time and bit-pattern searches can be restricted to this partition.

It is easy to calculate the dimension of the new basis, as this is simply the difference between the number of m-scheme basis states which have $J_z$=J and $T_z$=T and the number with either $J_z$=J+1 and $T_z$=T, or $J_z$=J and $T_z$=T+1. These numbers can be obtained when the m-scheme basis is being generated since all the relevant bit patterns are formed in the process and they can be numbered. Even if this dimension is known an arbitrary choice of this number of m-scheme basis states for projection would not, in general, scan the full J,T basis space. When creating the new basis, it is necessary to ensure that a linearly-independent set of states is obtained and this is now summarised. As mentioned in the above section describing 'PREDICT' it is best to start the projection with a predicted set of 'good' m-scheme vectors that when used to project from will probably generate a J,T basis. (This prescription might fail since 'PREDICT' can not predict whether these vectors would have a large linearly-independent component or whether this component is small in which case some vectors could be rejected in an effort to avoid rounding errors. Nevertheless, as will be explained later, 'PROJ' could successfully carry on with the calculation even if the predicted set finishes.)

'PREDICT' could be avoided since 'PROJ' has a further way of predicting good J,T states, although it is again emphasised that it 'PROJ' works fastest when ran with 'PREDICT'. This further method is based on a random vector: an arbitrary good J,T vector is obtained by projecting from a random linear combination of the m-scheme basis states. Random numbers (normalised to give a unit vector) are used for the coefficients in the linear combination, in the hope that this 'random vector' will be dependent on all the good J,T basis vectors obtainable by projection from the m-scheme basis states. Then, as each new good J,T basis state is calculated, only that part of the random vector which is orthogonal to the new basis state and to all the previously derived basis states is retained. Hence any subsequent good J,T basis state will have an overlap with this random vector if it is linearly-independent of the other basis vectors. Furthermore, due to a property of the projection operator, $P^{JT}$, each coefficient in the

m-scheme expansion of the random vector is equal to the overlap of the random vector with the good J,T state obtainable by projection from the corresponding m-scheme basis state. Thus it is easy to predict which m-scheme states are viable for obtaining a new good J,T basis state by inspecting the coefficients of the random vector.

In order to project we need to apply the operator $J^-J^+$ for each partition and this is done by applying $J^+$ to each m-scheme vector and then $J^-$ to this result. It is best to apply it sequentially rather than to apply $J^-J^+$ as a single operator because in the former case we need only 2*NPTCL (no. particles) applications (see eq. (21) and (22)) for each m-scheme vector whereas in the latter case we would require NPTCL*NPTCL applications that would be in general significantly larger. 'JMJP0' contructs and stores in memory the $\langle J_z+1'|j^+|J_z\rangle$ ($=\langle J_z|j^-|J_z+1'\rangle$) matrix to be used by 'JMJP1' and 'JMJP2'. Note that by doing this we require to have on disk $J_z+1$ m-scheme basis elements. This is not in general an important drawback as 'BASIS' is fast and since we usually want to calculate different spins for a single nucleus. On the other hand, this might not be the case for T-projection and since the number of T-values to be projected out will generally be much smaller than the number of J-values, (the maximum to $t_z$ is 0.5) 'TMTP' is used to construct the $\langle T_z'|T^-T^+|T_z\rangle$ matrix.

M6a: Techniques of PRIORPROJ

Consider $|O_i\rangle$ an orthonormal JT-basis vector, $|I(1)\rangle$, $|I(2)\rangle$,... a 'LI' m-scheme set (certain to produce a good JT-basis) and $\alpha(i,j)$ the expansion coefficients of $|O_i\rangle$ in terms of the non-orthogonal basis $P^{JT}|I(j)\rangle$;

$$|O_i\rangle = \Sigma\ \alpha(i,j)\ P^{JT}\ |I(j)\rangle \qquad (23)$$

The procedure in 'PRIORPROJ' for obtaining a linearly-indepenent set of states is as follows:

Built up the wave function $|M_n\rangle$;

$$|M_n\rangle = |I(n)\rangle - \Sigma\ \langle O_i|I(n)\rangle\ \Sigma\ \alpha(i,j)\ |I(j)\rangle \qquad (24)$$

then project to obtain $|N_n\rangle$;

$$|N_n\rangle = P^{JT}\ |M_n\rangle \qquad (25)$$

and normalise to obtain new basis vector $|O_n>$;

$$|O_n> = |N_n>(<N_n|N_n>)^{-1/2} \qquad (26)$$

now obtain $\alpha(n,j)$ coefficients by inverting ('INVERT.FOR') the matrix $<O_n|P^{JT}|I(j)>=<O_n|I(j)>$. These latter coefficients are obtained by scanning the coeffcients of $|O_n>$ in the m-scheme basis obtained after applying $P^{JT}$ in (25) and normalising in (26).

M6b: Techniques of 'POSTPROJ'

'PRIORPROJ' projects from a linear combination of m-scheme vectors so as obtain an already orthogonal vector. On the other hand, 'POSTPROJ' firstly projects and afterwards it orthogonalises. The procedure is as follows;

Obtain the n'th good J,T state by projection:

$$|G_n> = P^{JT} |I(n)> \qquad (27)$$

orthogonalise by substracting overlaps with previous $|O_i>$ vectors

$$|N_n> = |G_n> - \Sigma <O_i|I(n)> |O_i> \qquad (28)$$

and normalises to obtain new basis vector $|O_n>$;

$$|O_n> = |N_n>(<N_n|N_n>)^{-1/2} \qquad (29)$$

We note that 'POSTPROJ' does not require the inversion of a matrix as it goes along. Nevertheless, as 'MATRIX' needs the above ($\alpha(i,j)$) coefficients the inversion is performed at the end of each partition.

M6c: Comparison Between 'PRIORPROJ' and 'POSTPROJ'

Advantages of 'PRIORPROJ'

a) For large dimension all coefficients (previous $|O_i>$ wave functions) can not be resident in memory and therefore 'POSTPROJ' needs to write/read them on disk. This is very time consuming.

b) The number of subtractions needed to generate an orthogonal vector is far less in 'PRIORPROJ' and thus this might be significantly faster. This is, as shown above, because we need to substract overlaps with previous good

m-scheme vectors, rather than with the complete m-scheme wave function.

Advantage of 'POSTPROJ'

Proton partition (number of protons per J-level) symmetry conserved. When we apply the J-projection routine starting from a single m-scheme vector $|I(j)>$ we end up with a linear combination of m-scheme basis states all having the same number of protons and neutrons per J-orbit (obviously, all patterns have the same total number of either protons or neutrons). This is because $J^+$,$J^-$ and $J_z$ operators can change $t_z$-value. On the other hand, the symmetry is not complete in 'PRIORPROJ' as we start from a linear combination of m-scheme vectors which will not generally all have the same (neither all different) proton partitions. This symmetry is superfluous in the neutron-proton formalism for which all patterns have the same proton partition, since J-orbits are either occupied by protons or neutrons. It is clear that the labelling 'proton' or 'neutron' is a matter of convenience as the names could be swapped over.

As a final word, 'PRIORPROJ' is recommended for the neutron-proton formalism and 'POSTPROJ' for the isospin formalism, provided that it is not required to perform a large number of disk input/outputs. Outputs from both programs are totally equivalent and the calculation follows in a completely similar way.

M6d: Limits and Rounding Errors

'PROJ' has several rounding error traps that essencially reject a given J,T basis vector if it has a 'large' rounding error. Two tests are carried out by both 'PRIORPROJ' and 'POSTPROJ'. In subroutine 'PRJSBS', the random vector is tested to be an eigenstate of the total angular momentum operator $J^{op}$ with eigenvalue J(J+1). This test is performed for each random vector of each partition. 'PROJ' starts a run by asking for three integers 'IOUT, ILIM and IROU': 'IOUT' if set to larger than zero will produce more and more outputs, if 'ILIM' is set different from zero change of limits is permitted and 'IROU' works in the following fashion: if zero the above mentioned test is carried out only for the random vector, if equal to one, test carried out for all good J states and if equal or larger than two, test also performed for T-projection. The limit by which the eigenvalue is allowed

to differ from J(J+1) (or T(T+1)) is called PJLIM which is at present equal to 1.E-7.

The second rounding error test is performed in the main program and 'checks' if $P^{JT}\,P^{JT} = P^{JT}$ for each good J,T vector;

take $$|N_n\rangle = \Sigma\ \text{coef0}(i)|i\rangle, \tag{30}$$
so

$$\langle N_n|N_n\rangle = \Sigma\ \text{coef0}(i)^2 = \langle M_n|P^{JT}P^{JT}|M_n\rangle = \text{coef0}(N) \tag{31}$$
and if

$$(\ \Sigma\ \text{coef0}(i)^2 - \text{coef0}(N)\ )/\text{coef0}(N) > 0.1\% \tag{32}$$
the vector is rejected.

'PRIORPROJ' also check that the expansion of each J,T orthonormal vector ($|O_i\rangle$) in the $P^{JT}|I(j)\rangle$ basis has unit length: as seen in (23)

$$|O_i\rangle = \Sigma\ \alpha(i,j)\ P^{JT}\ |I(j)\rangle \tag{33}$$

thus,

$$\langle O_i|O_i\rangle = \langle I(i)|P^{JT}|O_i\rangle$$

$$= \Sigma\ \alpha(i,j)\ \langle I(i)|P^{JT}|I(j)\rangle \tag{34}$$

the vector is rejected unless it length differs from 1 in less than 0.1%. A further, most important limit, is NLILIM: $|N_i\rangle$ vectors are rejected if

$$\langle N_i|N_i\rangle > \text{NLILIM} \tag{35}$$

this limit is also used to predict next 'LI' m-scheme vector by the random vector. After a new $|O_i\rangle$ vector is accepted, a new random vector $|R_{i+1}\rangle$ is made orthogonal to it by substracting the overlap $\langle O_i|R_i\rangle$;

$$|R_{i+1}\rangle = |R_i\rangle - \langle O_i|R_i\rangle\ |O_i\rangle \tag{36}$$

the coefficients of the random vector are inspected, and the next m-scheme basis state which satisfies;

$$\langle m|R_{i+1}\rangle > 2*\text{NLILIM} \tag{37}$$

is used for projection.

'PROJ' creates two unformatted files: one with extension 'PRJ' containing the orthonormal wave function m-scheme expansion coefficients and a second with extension 'RED' with the α-coefficients.

M6e: What to do if PROJ fails

a) Try a different value of $2*J_z$ and/or $2*T_z$. In SHELL the default values are $2*J_z = 2*J$ and $2*T_z = 2*T$. You will need to run basis for the values if they have not already been run.

b) Try changing the limits for various tests in the program by setting ILIM from 0 to 1 in the *.COM file and then entering the values of parameters. See section M6d above and the comments in the program PROJ.FOR on area SHPR for details.

M7: About MATRIX, OPER and CHANGE

The matrix elements of the Hamiltonian operator H between the many-particle basis wave functions can be obtained by using the second-quantised operators $a^+$ and a. Thus, H can be written as

$$H = H_{core} + \sum \epsilon_i \; a^+_i \, a_i + \sum_{\substack{i>j=1 \\ k>l=1}} v_{ijkl} \, a^+_i \, a^+_j \, a_l \, a_k \qquad (38)$$

where $\epsilon_i$ is the energy of a single-particle state, and

$$v_{ijkl} = \langle ij|V|kl\rangle \qquad (39)$$

and $|ij\rangle$ is an antisymmetrised two-particle m-scheme state. The results of $a^+_i \, a_i$ and $a^+_i \, a^+_j \, a_k \, a_l$ operations can be determined by simple computer tests, and the matrix elements $\langle ij|V|kl\rangle$ can easily be derived from the conventional J and T coupled two-body matrix elements, since

$$\langle j_1 \, m_1 \, t_{z1} \; j_2 \, m_2 \, t_{z2} \; |V| \; j_3 \, m_3 \, t_{z3} \; j_4 \, m_4 \, t_{z4}\rangle$$

$$= \sum_{J \; T} \langle j_1 \; j_2 \; |V| \; j_3 \; j_4\rangle^{JT} \, [(1+\delta_{12})(1+\delta_{34})]^{1/2}$$

$$\times \langle j_1 \, m_1 \; j_2 \, m_2 \; |J \; m_1+m_2\rangle \, \langle j_3 \, m_3 \; j_4 \, m_4 \; |J \; m_3+m_4\rangle$$

$$x \langle 1/2\ t_{z1}\ 1/2\ t_{z2} |\ T\ t_{z1}+t_{z2}\rangle$$

$$x \langle 1/2\ t_{z3}\ 1/2\ t_{z4} |\ T\ t_{z3}+t_{z4}\rangle \qquad (40)$$

'OPER' executes the above transformation: it reads J,T matrix elements from disk and writes a new file, with extension 'OP', with the m-scheme matrix elements. It partitions the matrix elements v according to which j-orbit are created and destroyed by the corresponding $a^+$ and a operators. In this way, a number of operations can be made successively which involve only single basis partitions of the operand and the resultant. 'HCM' can be used to generate harmonic oscillator J,T matrix elements. 'CHANGE' allows one to add two interactions, multiplied by any number. These two programs might be useful to detect eigenvectors with a large spurious center of mass motion. 'CHANGE' can be run to generate new 'SPS' files.

'MATRIX' will first of all make up $\langle I(j)|H|O_i\rangle$ matrix elements, where the notation has been above explained. After this, a subroutine called 'ORTHOG' is called to obtain $\langle O_j|H|O_i\rangle$ matrix. This is easily performed by reading the $\alpha(i,j)$ coefficients from the 'RED' file. The result is written in a file with extension 'MAT'.

M8: About TRAMP

After running 'MVEC', 'TRAMP' could be run to obtain some of the following information;

M8a: Shell-Model Parentage Amplitudes

In order to calculate spectroscopic amplitudes from the shell-model wave functions, it is necessary to remove their dependence on the center of mass coordinates. Therefore, an intermediate result which is simply an overlap of shell model wave functions is defined. This is known as a parentage amplitude and its relation to a spectroscopic amplitude is dealt with in the next section.

Before defining the parentage amplitudes, it is useful to explain the conventions used for calculating these in the code 'TRAMP'. Initially, only two particles with angular momentum will be considered, since the following formulae are easily generalised to the many-particle case with angular momentum, isospin, and radial quantum numbers.

The antisymmetrised m-scheme wave functions for two particles can be written as;

$$|j_m j'_{m'}> = a^+_{j'm'}\ a^+_{jm}\ |> \qquad (41)$$

where |> is the vacuum state. This wave function corresponds to the bit pattern '11', where the rightmost bit represents the occupancy of the state |jm>.

The antisymmetrised wave function for two particles coupled to J and T, and normalised to unity is then;

$$\begin{aligned} |jj'JM> &= (1+\delta_{j,j'})^{-1/2} \\ &\quad x \sum <jmj'm'|JM>\ a^+_{j'm'}\ a^+_{jm}\ |> \\ &= -(1+\delta_{j,j'})^{-1/2}\ [a^+_j \otimes a^+_{j'}]^{JM}|> \qquad (42) \end{aligned}$$

Therefore, a creation operator which creates two particles in the state |jj'JM> can be defined as ;

$$X^+(jj';JM) = -(1+\delta_{j,j'})^{-1/2}\ [a^+_j \otimes a^+_{j'}\ ]^{JM} \qquad (43)$$

The form of the Wigner-Eckart theorem which will be used in subsequent formulate correspond to that of (Bo75);

$$<JM|T_{kq}|J'M'> = (2J+1)^{-1/2}\ <J'M'kq|JM>\ <J||T_k||J'> \qquad (44)$$

where $T_{kq}$ is a spherical tensor operator of rank k.

The notation will now be introduced where one Greek symbol is used to denote angular momentum and isospin quantum numbers, and in some cases, also the other quantum numbers necessary to define a particular shell-model state. Thus, $\rho$ will be used for the single-particle state quantum numbers $n,\ell,j$ and $|\Gamma>$ denotes a many-particle state (in general), with angular momentum and isospin quantum numbers J,T.

It is now possible to define the shell-model parentage amplitude for one particle in the state $\rho$ as;

$$S^{1/2}(\rho) = (2\Gamma+1)^{-1/2} <\Gamma|||a^{+}|||\Gamma'> \qquad (45)$$

which is the same as the definition given by (Br77) for their shell-model spectroscopic amplitude. Similarly, the two-particle parentage amplitude for two particles coupled to angular momentum and isospin '$\lambda$' can be defined as;

$$S^{1/2}(\rho,\rho';\lambda) = (2\Gamma+1)^{-1/2} <\Gamma|||X^{+}(\rho,\rho';\lambda)|||\Gamma'>. \qquad (46)$$

A generalisation of the two-particle creation operator $X^{+}$ to an operator which creates k particles in the state with all the relevant quantum numbers specified by '$\lambda$' now allows the k-particle parentage amplitude to be written:

$$S^{1/2}(k;\lambda) = (2\Gamma+1)^{-1/2} <\Gamma|||X^{+}(k;\lambda)|||\Gamma'> \qquad (47)$$

The computer code 'TRAMP' can generate parentage amplitudes for one particle, two particles or for a cluster of particles. This parentage amplitudes can easily be transformed to spectroscopic amplitudes by introducing two factors, as it will now be shown. The definition of a spectroscopic amplitude for the break up of a nucleus A into a nucleus A' and the particle group k is;

$$\theta(k;\lambda) = \mathrm{BINOM}(A/k)^{1/2} <\phi^{\Gamma}_{A}\ \phi^{\Gamma'}_{A'}\ |[\phi^{\Gamma}_{k}\ \phi^{NL}_{kA'}]^{\lambda}> \qquad (48)$$

BINOM() is the binomial coefficient. $\phi$ denotes an antisymmetrised wave function which depends only on internal coordinates and $\phi^{NL}_{kA'}$ describes the relative motion between the centers of mass of k and A'. It can be seen that the differences between a parentage and a spectroscopic amplitude are that the parentage amplitude is an overlap of fully antisymmetrised wave functions, whereas there is no antisymmetrisation between the particle groups

k and A' in the ket of the spectroscopic amplitude, and also that the parentage amplitude depends on internal and center of mass coordinate, whilst the spectroscopic amplitude is a function of internal coordinates only. For two shells, the amplitudes can now be related (An74, Sm61);

$$\theta(k;\lambda) = (A/(A-k))^{\lambda/2}\, G([N_i]^k)\, S^{1/2}(k;\lambda) \qquad (49)$$

where

$$G([N_i]^k) = [k!\lambda!/(k_1!k_2!k^{\lambda/4})]^{1/2} \times \prod_{i=1}^{k} (N_i!)^{-1/2} \qquad (50)$$

where $\lambda = \Sigma N_i = \Sigma (2n_i + \ell_i)$, with $n_i$ and $\ell_i$ being the principal and angular momentum quantum numbers. $k_1$ is the number of particles from the cluster that occupy orbits on the first shell and $k_2$ is the number of particles that occupy orbits in the second shell, $k = k_1+k_2$.

In deducing these two factors relating the amplitudes, harmonic oscillator wavefuntions with the same single-particle oscillator frequency are assumed, which is not necessarily a good assumption.

M8b: One-Body Transition Densities

In order to enable the matrix elements of a single particle operator to be evaluated between the wave functions obtained with 'OXBASH', the calculation of one-body transition densites has been coded in 'TRAMP'. These are defined by

$$OBTD(\rho,\rho';\lambda) = (2\lambda+1)^{-1/2} \langle\Gamma|||[a^{+}_{\rho}\otimes a^{\sim}_{\rho}]^{\lambda}|||\Gamma'\rangle \quad (51)$$

where $a^{\sim}$ is a tensor operator which creates the hole state $|\rho^{-1}\rangle$, and which is related to the usual particle annihilation operator by

$$a^{\sim}_{j,m} = (-1)^{j+m} a_{j,-m} \quad (52)$$

The many particle matrix elements of the single particle tensor operator O can be calculated, since

$$\langle\Gamma|||O^{\lambda}|||\Gamma'\rangle = \sum_{\rho,\rho'} \langle\rho|||O^{\lambda}|||\rho'\rangle \, OBTD(\rho,\rho';\lambda). \quad (53)$$

This formula is relevant to the calculation of spurious center of mass motion and to the evaluation of electromagnetic transition probabilities.

M8c: Two-Body Transition Densities

As the calculation of one-body transition densities is relevant to single particle operators, two-body transition densities will be required to generate two-body operators. The calculation of two-body transition densities coded up in 'TRAMP' are defined as below;

$$TBTD(\rho_1\rho_2;\lambda_{12}:\rho_3\rho_4;\lambda_{34}:\lambda)$$

$$= (2\lambda+1)^{-1/2} \times \langle\Gamma|||TBTD^{op}|||\Gamma'\rangle \quad (54)$$

where

$$TBTD^{op}$$

$$= -[(1+\delta_{\rho_1\rho_2})(1+\delta_{\rho_3\rho_4})]^{-1/2} \times$$

$$\times \{[a^{+}_{\rho_1}\otimes a^{+}_{\rho_2}]^{\lambda_{12}}\otimes[a^{\sim}_{\rho_3}\otimes a^{\sim}_{\rho_4}]^{\lambda_{34}}\}^{\lambda} \quad (55)$$

The wavefuntions $\Gamma$ and $\Gamma'$ are worked out by 'OXBASH' in the standard fashion and the operator as defined in the last equation is read in from disk. Thus, we need to create

before hand which is easily done by running 'TBTDOP'. This is a procedure similar to that performed with 'OPER' and 'MATRIX'.

M8d: Direct and Core-Cluster Overlaps

The direct overlap option in TRAMP will give the overlap probability between two eigenvectors with the same number of particles. The Core-Cluster option takes the overlap of the form <Core|Cluster|Final> where the number of particles in the Core+Cluster must add up to the number of particles in the Final state.

It is important that the $J_z$ and $T_z$ values of the wave functions add up properly and one must remember that the "default" values are $J_z$=J and $T_z$=T. Often one must edit the *.COM files produced by SHELL to change the $J_z$ and $T_z$ value so that these overlaps can be calculated.

M9: TRANS, ISOV and PNCME

These programs take the OBTD and TBTD output of TRAMP and calculate various observables. For the most part they are self-documenting. This means that in the first pass the defaults (<CR>) can be used for all questions except the one pertaining to the name of the OBTD of TBTD file. More details of the equations used and the parameters can then be obtained from the output and these can then be changed on the second pass. Program summaries:

TRANS purpose: Calculate moments B(EL) B(ML) and B(GT)
input: The *.TRD files on the present area
output: TRANS.DAO contains the full output
TRANS.DAS contains a shortened version of the same output
references:
moments: B. A. Brown et al. Phys. Rev. C22, 774 (1980)
B(E2): B. A. Brown et al. Phys. Rev. C26, 2247 (1982)
B(E4): B. A. Brown et al. Phys. Rev. C21, 2600 (1980)
B(GT): B. H. Wildenthal et al. Phys. Rev. C28, 1343 (1983)

ISOV purpose: Calculate b and c coefficients of isobaric mass mutiplet equation and isospin mixing matrix elements
input: For b-coefficients and isovector mixing
The *.TRD and *.TB1 files on the present area plus the *.ISO file on area SHISOH
input: for c-coefficients and isotensor mixing
The *.TB2 file on the present area
plus the *.ISO file on area SHISOH
output: ISOV.DAS

PNCME purpose: Calculate parity nonconserving (PNC) matrix elements
input: For isoscalar PNC
The *.TRD and *.TBD files on the present area plus the isoscalar *.PNC file on area SHPNCH
input: For isovector PNC
The *.TRD and *.TB1 files on the present area plus the isovector *.PNC file on area SHPNCH
input: For isotensor PNC
The *.TB2 file on the present area
plus the isotensor *.PNC file on area SHPNCH
output: PNCME.DAS
references: B. A. Brown et al. Phys. Rev. Lett. 21, 1681 (1980)

M10: Spurious Center of Mass Motion

At present two different approaches can be readily used to estimate the center of mass spuriosity in shell model wave functions. In the first approach, 'HCM' and 'CHANGE' are run in order to obtain a new interaction, Hnew. This new interaction is the sum of the Hamiltonian for the center of mass motion, HCM, multiplied by a factor at wish (typically 10) plus the Hamiltonian used to generate the above mentioned shell-model wave functions, H;

$$H_{new} = 100 * H_{cm} + H \qquad (56)$$

New shell-model wave functions are now generated with $H_{new}$ in the hope that only those with large spuriosity will significantly increase their eigenenergy.

The intrinsic motion for a many-particle system, such as a nucleus, should be described by a wave function which involves only relative coordinates. However, in practice, it is unfeasible to construct a many-particle wave function which observes the antisymmetrization requirements of the Pauli principle, unless the set of spatial coordinates treats all the particles in a symmetric manner. This can be done by referring all the coordinates to an arbitrary origin, as in the shell-model of the nucleus. In the shell model description the particles are bound to the origin by the single-particle potential U, which has the consequence that the center of mass of the system undergoes unphysical oscillations about this point. In principle, any shell model wave function can be separated into a wave function describing the internal motion of the system and a wave function for the center of mass motion. However, it is not necessary to make this separation if the center of mass is in its lowest state of motion for all the states of interest, since the relative energies of the states are unaffected by the center of mass energy and the wave function overlaps required for calculating spectroscopic amplitudes can be transformed from overlaps of shell-model wave functions by using the factor given in equations (49) and (50). In order to ensure that the center of mass will always be in its lowest state of motion, all the wave functions which describe exicted center of mass states must be completely excluded from the shell-model basis. This is possible, if the shell-model potential U is a harmonic oscillator potential. Then, the shell-model Hamiltonian H can be separated into a function of the relative coordinates, which is the Hamiltonian for the internal motion $H_{rel}$, and a function of the center of

mass coordinates, which is the Hamiltonian for the center of mass motion $H_{CM}$. Hence, by selecting the eigenstates of $H_{CM}$ with the lowest center of mass energy, a shell model basis will be obtained which is free from any spurious effects due to excited center of mass motion.

If $p_i$ and $r_i$ are the momentum and position vectors for a single particle and k is the oscillator constant, the separation of the shell-model Hamiltonian H for a system of A particles follows, since

$$\begin{aligned} H &= \sum_{i=1,A} \{ (1/2m)\, p_i^2 + (Ak/2)\, r_i^2 \} \\ &= \sum_{i<j} \{ (1/2Am)\, (p_i - p_j)^2 + (k/2)\, (r_i - r_j)^2 \} \\ &\qquad + \{ (1/2Am)\, P^2 + (A^2k/2)\, R^2 \} \\ &= H_{rel} + H_{CM}, \end{aligned} \tag{57}$$

where P and R are the momentum and position vectors for the center of mass, defined by

$$P = \Sigma_i\, p_i \tag{58}$$

and

$$R = (1/A)\, \Sigma_i\, r_i \tag{59}$$

A convenient formula for calculating the matrix elements of $H_{CM}$ for use in shell-model calculations may be obtained by introducing the vector creation and annihilation operators of the single-particle oscillator quanta:

$$a^+_i = (2m\hbar\omega)^{-1/2}\, (p_i + im\omega\, r_i) \tag{60}$$

and

$$a_i = (2m\hbar\omega)^{-1/2}\, (p_i - im\omega\, r_i), \tag{61}$$

where $\omega = (Ak/m)^{-1/2}$.

Thus,

$$\begin{aligned} H_{CM} &= (1/2Am)\, P^2 + (A^2k/2)\, R^2 \\ &= (1/A)\, \Sigma_i \{(1/2m)\, p_i^2 + (Ak/2)\, r_i^2\} \\ &\qquad + (1/2Am) \sum_{i \neq j} \{ p_i.p_j + Akm\, r_i.r_j \} \\ &= (1/A)\, H + (\hbar\omega/2A) \sum_{i \neq j} \{a^+_i.a_j + a_i.a^+_j\} \end{aligned} \tag{62}$$

The expression in equation (62) shows the division of the operator $H_{cm}$ into a single-body operator and the two-body operator $V_{cm}$. The single-particle energies will, therefore, be given by

$$\epsilon_i = (1/A)(2n_i + \ell_i + 3/2)\,\hbar\omega + (2\rho_i+1)^{-1} \Sigma_{\rho,\Gamma}\,(2\Gamma+1) \times \langle \rho_i \rho_c\, \Gamma | V_{cm} | \rho_i \rho_c\, \Gamma \rangle \qquad (63)$$

where $n_i$ and $\ell_i$ are the radial and angular momentum quantum numbers of single-particle state, and the term involving $V_{cm}$ gives the interaction energy of the single particle with the core. The two-body matrix elements of $V_{cm}$ can be calculated from the matrix elements of the single-particle operators $a^+$ and a, which are given in reference (Ga57), since for nonantisymmetrized two-body states:

$$\begin{aligned} &\langle \ell_1 \ell_2 | a^+_1 . a_2 | \ell_3 \ell_4 \rangle_L \\ &= -3^{1/2} \langle \ell_1 \ell_2 | [a^+_1 \times \tilde{a}_2]_{00} | \ell_3 \ell_4 \rangle_L \\ &= (-1)^{L+\ell_1+\ell_4}\, W(\ell_1 \ell_2 \ell_3 \ell_4; L1) \\ &\quad \times \langle \ell_1 || a^+ || \ell_3 \rangle \langle \ell_2 || \tilde{a} || \ell_4 \rangle \end{aligned} \qquad (64)$$

where W is a Racah coefficient. Then, the result for the two-body matrix elements of $V_{cm}$ calculated between antisymmetrized states is

$$\begin{aligned} &\langle \rho_1 \rho_2 | V_{cm} | \rho_3 \rho_4 \rangle \\ &= -(\hbar\omega/A)\,[(2j_1+1)(2j_2+1)(2j_3+1)(2j_4+1)]^{1/2} \\ &\times [(1+\delta_{\rho_1\rho_2})(1+\delta_{\rho_3\rho_4}]^{-1/2} \\ &\times \Sigma_{L,S}\,(2L+1)\,(2S+1) \\ &\times 9j(\ell_1\ s_1\ j_1\ \ell_2\ s_2\ j_2\ L\ S\ J) \\ &\times 9j(\ell_3\ s_3\ j_3\ \ell_4\ s_4\ j_4\ L\ S\ J) \\ &\times \{\ (-1)^{L+\ell_1+\ell_4}\, W(\ell_1 \ell_2 \ell_3 \ell_4; L1) \\ &\quad \times \langle \ell_1 || a^+ || \ell_3 \rangle \langle \ell_2 || a^+ || \ell_4 \rangle \end{aligned}$$

$$+ (-1)^{S+T+1+l_1+l_4} W(\ell_1\ell_2\ell_4\ell_3;L1)$$

$$x \langle \ell_1 || a^+ || \ell_4 \rangle \langle \ell_3 || a^+ || \ell_2 \rangle \quad \} \qquad (65)$$

In certain calculations it is important to know the amount of spurious center of mass motion in a particular shell-model wave function. This can sometimes be determined by calculating the expectation value of $H_{new}$ as defined in Eq. (56)) and comparing these new wave functions with those obtained from the Hamiltonian H (see Eq. (56)).

A second method has been described by Baranger and Lee (Ba61) which uses the operator for creating $1\hbar\omega$ excitations of the center of mass:

$$A^+ = (2Am\hbar\omega)^{-1/2} (P + iAm\omega R) = A(-1/2) \Sigma_i a^+_i \qquad (66)$$

It is known (El55) that states which have the lowest energy allowed by the Pauli principle are also the states with the center of mass in its lowest state of motion. Therefore, by applying the operator $A^+$ to these states, all the spurious states with a particluar center of mass energy can be generated and overlaps can be taken in order to obtain the weight of spurious states in any other shell-model state. In fact, it is not necessary to generate the spurious states explicitly and for a shell-model state involving $1\hbar\omega$ excitations, the weight of spurious states W can be obtained by using the one-body transition densities (OBTD's) calculated between all the $0\hbar\omega$ states and the $1\hbar\omega$ state of interest:

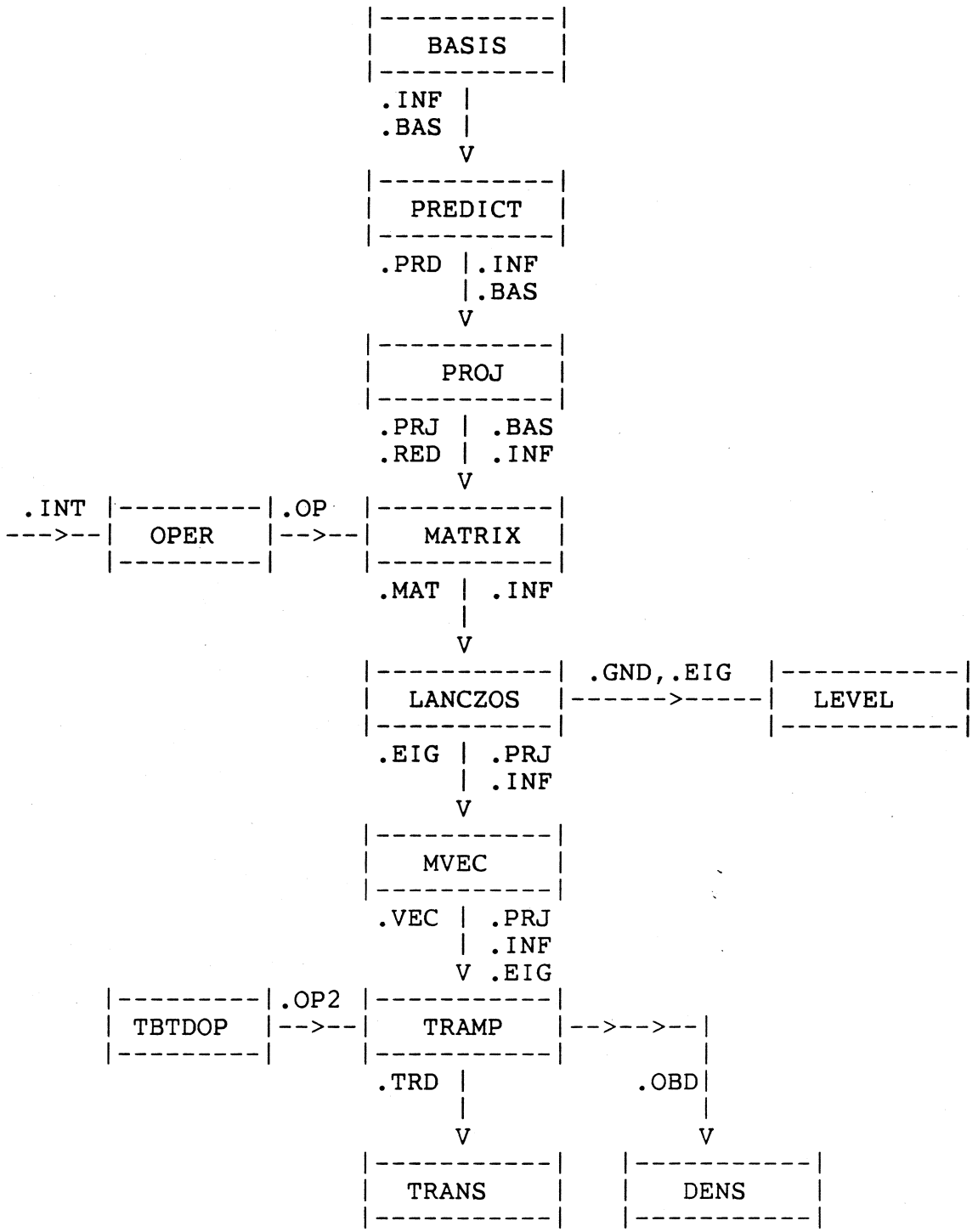
$$W = [A(2\Gamma+1)]^{-1} \Sigma_\beta \mid \Sigma_{\rho,\rho'}$$

$$x \langle \rho ||| a^+ ||| \rho' \rangle \quad OBTD(\rho,\rho';\lambda=(1,0);\beta) \mid^2 \qquad (67)$$

where the sum involving $\beta$ runs over all the shell-model wave functions $0\hbar\omega$ center of mass states which can couple with the $1\hbar\omega$ state to make $\lambda=(1,0)$. 'OBOP' calculates 'W' as defined in Eq. (67).

## M11: Diagram of Typical Calculations

The diagram shows the sequence in which programs should be run and the extensions of the input and generated disk files by each executable code. These extensions, when written on the left hand (or upper) side, correspond to files just created, and when written on the right hand (or lower) side, correspond to file already on disk, that is created by another program rather than the one just above. In either case, these files are used by the computer code which is run immediately after. The '.SPS' file is needed by all programs. (DENS is not part of OXBASH.)


BASIS
.INF
.BAS
PREDICT
.PRD
.INF
.BAS
PROJ
.PRJ
.BAS
.RED
.INF
.INT
OPER
.OP
MATRIX
.MAT
.INF
LANCZOS
.GND,.EIG
LEVEL
.EIG
.PRJ
.INF
MVEC
.VEC
.PRJ
.INF
.EIG
TBTDOP
.OP2
TRAMP
.TRD
.OBD
TRANS
DENS

## M12: The multiple-precision integer macro routines

There are four sets of macro subroutines to be used if multiple precision integers calculations are to be performed: XARITH, XLOGIC, XSHIFT and XTEST. For these subroutines (or functions) A, B and RESULT are multi-precision integer operands and LENGTH is an integer telling the number of longwords of the operands. The programs were written at Oxford by Mr. T.W.Thacker.

------ XARITH.MAR - Extended precision arithmetic operations

```
Multiple precision ADD
Purpose  :  Calculate multiple precision sum
Call :      CALL LIB_ADDX(A,B,RESULT,LENGTH)
Multiple precision SUB
Purpose :  Calculate multiple precision difference
Call :           CALL LIB_SUBX(A,B,RESULT,LENGTH)
Return : RESULT set to A - B

Multiple precision assignment
Purpose :  Multiple precision integer to integer assignment
Call :     CALL LIB_ASSIGNX(A,B,LENGTH)
Return :   B  set to A

Multiple precision zero
Purpose :  Assignment of zero to multiple precision integer
Call :     CALL LIB_ZEROX(A,LENGTH)
```

------ XLOGIC.MAR  Extended precision logical operations

```
Multiple precision logical AND
Purpose :  Calculate multiple precision logical AND
Call :     CALL LIB_ANDX(A,B,RESULT,LENGTH)

Multiple precision logical OR
Purpose :  Calculate multiple precision logical OR
Call :     CALL LIB_ORX(A,B,RESULT,LENGTH)

Multiple precision logical XOR
Purpose :  Calculate multiple precision logical XOR
Call :     CALL LIB_XORX(A,B,RESULT,LENGTH)
```

------ Extended precision bits count

```
Purpose :  Count the number of set bits in a multiple precision
           integer
```

```
Call :      N = LIB_BITSX(A,LENGTH)


------ XSHIFT.MAR  Extended precision logical shifts

Purpose :  Calculate multiple precision logical shift
Call :     CALL LIB_SHIFTX(INPUT,KOUNT,RESULT,LENGTH)
           KOUNT integer*4   number of bits by which to shift INPUT


------ XTEST.MAR  Extended precision arithmetic tests

     Last bit in each word is equivalent to the other bits and does
not hold the sign (unsigned tests).

Multiple precision unsigned LESS OR EQUAL
Purpose :  Determines multiple precision unsigned LESS OR EQUAL
Call :     IF (LIB_LEX(A,B,LENGTH)) ...
Return :   LIB_LEX    = (A.LEU.B)

Multiple precision unsigned LESS
Purpose :  Determines multiple precision unsigned LESS
Call :     IF (LIB_LTX(A,B,LENGTH)) ...
Return :   LIB_LTX    = (A.LTU.B)

Multiple precision unsigned GREATER OR EQUAL
Purpose :  Determines multiple precision unsigned GREATER OR EQUAL
Call :     IF (LIB_GEX(A,B,LENGTH)) ...
Return :   LIB_GEX    = (A.GEU.B)

Multiple precision unsigned GREATER
Purpose :  Determines multiple precision unsigned GREATER
Call :     IF (LIB_GTX(A,B,LENGTH)) ...
Return :   LIB_GTX    = (A.GTU.B)

Multiple precision arithmetic NOT EQUAL
Purpose :  Determines multiple precision arithmetic NOT EQUAL
Call :     IF (LIB_NEX(A,B,LENGTH)) ...
Return :   LIB_NEX    = (A.NE.B)

Multiple precision arithmetic EQUAL
Purpose :  Determines multiple precision arithmetic EQUAL
Call :     IF (LIB_EQX(A,B,LENGTH)) ...
Return :   LIB_EQX    = (A.EQ.B)

Multiple precision BIT TEST and SET
Purpose :  Test a specified bit of a multiple precision integer,
           then set it.
Call :     IF (LIB_BTSX(A,N,LENGTH)) ...
```

```
Return :   LIB_BTSX = (A.AND.2**N) .NE. 0
         A = A .OR. 2**N

Multiple precision BIT TEST and CLEAR
Purpose :  Test a specified bit of a multiple precision integer,
           then clear  it.
Call :     IF (LIB_BTCX(A,N,LENGTH)) ...
Return :   LIB_BTCX = (A.AND.2**N) .NE. 0
         A = A .AND. .NOT.(2**N)

Multiple precision BIT TEST
Purpose :  Test a specified bit of a multiple precision integer.
Call :     IF (LIB_BTX(A,N,LENGTH)) ...
Return :   LIB_BTX    = (A.AND.2**N) .NE. 0
```

M13: References

AN74 N. Anyais-Weiss, J.C. Cornell, P.S. Fisher, P.N. Hudson, A. Menchaca-Rocha, D.J. Millener, A.D. Panagiotou, D.K. Scott, D. Strottman, D.M. Brink, B. Buck, P.J. Ellis and T. Engeland; Phys. Rep. C12 (1974) 201.

Ba61 E. Baranger and C.W. Lee, Nucl. Phys. 22 (1961) 157.

Bo75 A. Bohr and B.R. Mottelson, Nuclear Structure, Vol I, (W.A. Benjamin Inc. 1975).

Br77 P.J. Brussaard and P.W.M. Glaudemans, Shell Model Applications in Nuclear Spectroscopy, (North-Holland 1977).

El55 J.P. Elliot and T.H.R. Skurme, Proc. Roy. Soc., London A232 (1955) 561.

Ga57 S. Gartenhaus and C. Schwartz, Phys. Rev. 108 (1957) 482.

Sm61 Ya. F. Smirnov, Nucl. Phys. 27 (1961) 177.

M14: History of OXBASH

1976 Conception and embryo version: W. D. M. Rae and C. H. Zimmerman

1978 Working PDP10 version : N. S. Godwin and W. D. M. Rae

1980 First VAX version with MACRO multiple-integer words, new BASIS, added LANCZOS and added tables in PROJ and MATRIX : A. Etchegoyen, B. A. Brown, W. A. Richter, N. S. Godwin and J. S. Winfield

1982 Added PREDICT, changes in PROJ and MATRIX, new ORTHOG and tables incorporated to TRAMP : W. D. M. Rae

1982 General overall upgrade, changes in BASIS, OPER and PROJ, added PRIORPROJ, ORTHOG incoporated into MATRIX, multiple-integer words used for partitions: A. Etchegoyen and W. D. M. Rae. Added TRANS, ISOV, and PNCME: B. A. Brown.

1983 BASIS speeded up: J. S. Winfield. Documentation and general organization added: B. A. Brown.

1984-1986 Continued upgrades to MSU version: B. A. Brown

1986 Upgrade to standard FORTRAN and preparation for use on the FPS-164 array processor: W. E. Ormand (as of Nov 1986 there are still some bugs in the FPS version).

Dates for documented MSU versions:
Nov 1983
Dec 1984
Aug 1985
Nov 1985
Nov 1986

M15: OXBASH mailing list

| | |
|---|---|
| Address: | Prof. B. Alex Brown<br>National Superconducting Cyclotron Laboratory<br>Michigan State University<br>E. Lansing MI, 48824-1321 |
| Phone: | 517-355-7588 or FTS 517-355-7588 |

| | |
|---|---|
| Address: | Dr. A. Etchegoyen<br>Comision Nacional de Energai Atomica<br>Departamento de Fisica<br>Avenida del Libertador 8250<br>1429 Buenos Aires, Argentina |
| Date: | Nov 1983 |

| | |
|---|---|
| Address: | Dr. W. M. D. Rae<br>Nuclear Physics Laboratory<br>Keble Road,<br>Oxford, OX1-3RH England |
| Date: | Nov 1983 |

| | |
|---|---|
| Address: | Prof. E. G. Adelberger<br>Nuclear Physics Laboratory<br>University of Washington<br>Seattle, Washington 98195 |
| Phone: | 206-543-4080 or FTS 206-543-4080 |
| Date: | Dec 1984 |

| | |
|---|---|
| Address: | Dr. C. L. Woods<br>Dept. of Nuclear Physics<br>Institute for Advanced Study<br>The Australian National University<br>P. O. Box 4, Canberra ACT 2600, Australia |
| Date: | Dec 1984, Nov 1986 |

| | |
|---|---|
| Address: | Dr. W. A. Richter<br>Physics department<br>The Univerity of Stellenbosch<br>Stellenbosch 7600, South Africa |
| Date: | Dec 1984, Nov 1985 |

| | |
|---|---|
| Address: | Dr. S. J. Seestrom-Morris<br>MP4 MS-846<br>Los Alamos National Labortory<br>Los Alamos N.M. 87545 |
| Phone: | 505-663-5652 or FTS 843-5652 |
| Date: | Jan 1984 |

| | |
|---|---|
| Address: | Dr. C. M. Vincent<br>Physics department<br>University of Pittsburgh<br>Pittsburgh PA 15260 |
| Phone: | 412-624-2488 |
| Date: | Feb 14 1984, Nov 1986 |

| | |
|---|---|
| Address: | Dr. E. K. Warburton<br>Physics department<br>Brookhaven National Laboratory<br>Upton, NY 11973 |
| Phone: | 516-282-4560 |
| Date: | March 1985, Jan 1986 |

| | |
|---|---|
| Address: | Prof. A. Pilt<br>Tandem, Accelerator Labortory<br>McMaster University<br>Hamilton Ontario, L8S 4M1 Canada |
| Phone: | 416-525-9140 ext 4046 |
| Date: | Dec 1984 |

| | |
|---|---|
| Address: | Prof. B. H. Wildenthal<br>Department of Physics<br>Drexel University<br>Philadelphia, PA 19104 |
| Phone: | 215-895-2707 |
| Date: | Dec 1984 |

| | |
|---|---|
| Address: | Prof. H. Nann<br>Department of Physics<br>Indiana University<br>Bloomington, IN 47405 |
| Phone: | 812-335-3427 |
| Date: | Dec 1984, Jan 1986 |

| | |
|---|---|
| Address: | Dr. Sun Tsu hsun<br>Institute of Atomic Energy<br>Academica Sinica<br>P.O. Bax 275-48<br>Beijing, P. R. China |
| Date: | Dec 1984 |

| | |
|---|---|
| Address: | Dr. John A. Becker<br>L-280 Lawrence Livermore Laboratory<br>P. O. Box 808<br>Livermore CA 94550 |
| Date: | March 1985, August 1985 |

| | |
|---|---|
| Address: | Dr. Claude Pruneau<br>Physics department<br>Levelle University<br>Quebec, Canada G1K 7P4 |
| Date: | April 1985 |

| | |
|---|---|
| Address: | Dr. John Nelson<br>Department of Physics<br>University of Birmingham<br>Birmingham B15 2TT<br>England |
| Date: | July 1985 |

| | |
|---|---|
| Address: | Dr. H. Sagawa<br>Department of Physics<br>Univ. of Tokyo<br>Hongo Bunkyo-ku<br>Tokyo Japan |
| Date: | August 1985 |

| | |
|---|---|
| Address: | Dr. Y. Takahashi<br>Cyclotron Center<br>Tohoku Univ.<br>Sendai 980, Japan |
| Date: | August 1985 |

| | |
|---|---|
| Address: | Dr. E. Roeckle<br>GSI<br>Planckstrasse 1,<br>Postfach 110541<br>6100 Darmstadt 11, W. Germany |
| Date: | August 1985 |

| | |
|---|---|
| Address: | Dr. A. Couch<br>Dept. of Natural Philosophy<br>Glasgow Univ.<br>Glasgow 8QQ, U. K. |
| Date: | November 1985 |

| | |
|---|---|
| Address: | Dr. C. C. Yang<br>National Tsing Hau Univ.<br>Dept. of Physics<br>855 Kuang Fu Road,<br>Hsinchu, Taiwan |
| Date: | November 1985 |

| | |
|---|---|
| Address: | Dr. J. F. Dawson<br>Univ. of New Hampshire<br>Dept. of Physics<br>DeMerritt Hall<br>Durham, NH 03824 |
| Date: | November 1985 |

| | |
|---|---|
| Address: | Prof. H. T. Fortune<br>Physics Dept. 2E5 DRL E1,<br>Univ. of Penn.<br>Philadelphia, PA 19104 |
| Date: | Jan 1986 |

| | |
|---|---|
| Address: | Dr. R. Gilman<br>Phyiscs Dept.<br>Argonne National Laboratory<br>Argonne IL, 60439 |
| Date: | Jan 1986 |

| | |
|---|---|
| Address: | Dr. M. Zirnbauer<br>Kellogg Radiation Lab. 106-38,<br>Caltech,<br>Pasadena, CA 91125 |
| Date: | Feb 1986 |

| | |
|---|---|
| Address: | Prof. L. Zamick<br>Serin Phys. Lab.,<br>Busch Campus, Rutgers Univ.,<br>Piscataway, NJ 08854 |
| Date: | May 1986 |

| | |
|---|---|
| Address: | Prof. O. Hausser<br>TRIUMF<br>4004 Wesbrook Mall,<br>Vancouver, B. C. V6T 2A3, Canada |
| Date: | May 1986 |

| | |
|---|---|
| Address: | Dr. E. Ormand<br>NORDITA<br>Blegdamsvej 17<br>DK-2100 Kobenhavn 0<br>Denmark |
| Date: | Aug 1986 |

Address: Dr. T. Lonnroth
NORDITA
Blegdamsvej 17
DK-2100 Kobenhavn 0
Denmark
Date: Nov 1986

Address: Dr. P. Halse
Physics Dept. Blgd. 81
Univ. of Arizona,
Tucson, AZ 85721
Date: Nov 1986

E1: Sample *.SPS file in isospin formalism
The 'P.SPS' looks like:

```
  12    2    4    8    1    2     4  2
1P1/2     1P3/2
   1    1    1    1   -1   -1
   2    1    1    1   -1    1
   3    1    1    1    1   -1
   4    1    1    1    1    1
   5    1    1    3   -3   -1
   6    1    1    3   -3    1
   7    1    1    3   -1   -1
   8    1    1    3   -1    1
   9    1    1    3    1   -1
  10    1    1    3    1    1
  11    1    1    3    3   -1
  12    1    1    3    3    1
```

first line
- 12 = number of m states
- 2 = number of j orbitals
- 4 = number of m states in first orbit
- 8 = number of m states in second orbit etc
- 1 = number of major shells
- 2 = number of j orbits in first major shell etc
- 4 = A of core
- 2 = Z of core

second line = J-orbits names in A9 format

following lines = dummy and n,l,2*J,2*$J_z$,2*$T_z$ values.

E2: Sample *.SPS file in proton/neutron formalism
The 'PPN.SPS' looks like:

```
  12   4    2  4    2  4     2  2  2  4 2
P1P1/2    P1P3/2    N1P1/2    N1P3/2
     1     1     1     1    -1    -1
     2     1     1     1     1    -1
     3     1     1     3    -3    -1
     4     1     1     3    -1    -1
     5     1     1     3     1    -1
     6     1     1     3     3    -1
     7     1     1     1    -1     1
     8     1     1     1     1     1
     9     1     1     3    -3     1
    10     1     1     3    -1     1
    11     1     1     3     1     1
    12     1     1     3     3     1
```

The first line is different from that of 'P.SPS'. The J-orbits are divided into two groups: one for protons and one for neutrons. This grouping is totally equivalent to the major shell grouping in the isospin formalism: either grouping links J-orbits together. Nevertheless, the number of particles in each major shell in the isospin formalism can be varied by setting the relevant major shell restrictions whereas in the N/P formalism the number of particles in each group has to remain, obviously, fixed.

first line
- 12 = number of m states
- 2 = number of j orbitals
- 4 = number of m states in first orbit
- 8 = number of m states in second orbit etc
- 2 = number of major shells (always 2 in p/n)
- 1 = number of j orbits in proton shell
- 1 = number of j orbits in neutron
- 4 = A of core
- 2 = Z of core

second line = J-orbits names in A9 format

following lines = dummy and n,l,2*J,2*$J_z$,2*$T_z$ values.

E3: Sample *.INT file in proton/neutron formalism
The 'CKPOT.INT' looks like:

```
! ORDER IS:  1 = 1P1/2    2 = 1P3/2
    15       2.420,1.130
   1    1    1    1         0    1      0.24
   1    1    1    1         1    0     -4.29
   2    1    1    1         1    0      1.20
   2    1    2    1         1    0     -6.56
   2    1    2    1         1    1      0.73
   2    1    2    1         2    0     -4.06
   2    1    2    1         2    1     -1.14
   2    2    1    1         0    1     -5.05
   2    2    1    1         1    0      1.77
   2    2    2    1         1    0      3.21
   2    2    2    1         2    1     -1.74
   2    2    2    2         0    1     -3.33
   2    2    2    2         1    0     -3.44
   2    2    2    2         2    1      0.09
   2    2    2    2         3    0     -7.27
```

The first line and any subsequent line with a "!" in front is header information not used by the programs. NMAT in the second line is the number of matrix elements. This number is not used but OPER will give a warning message if the number of inputs differs from NMAT. The remaining entries in the second line are the single-particle energies. Then comes a list of the matrix elements in the form I1,I2,I3,I4,J,T,V where I is the orbit index number. The input is aways read in "free" format.

E4: Sample *.INT file in proton/neutron formalism
The 'CKPOTPN.INT' looks like:

```
 ! 1=P1P1/2 2=P1P3/2 3=N1P1/2 4=N1P3/2
    44   2.42000   1.13000   2.42000   1.13000
    1    1    1    1    0    1     0.24000
    2    1    2    1    1    1     0.73000
    2    1    2    1    2    1    -1.14000
    2    2    1    1    0    1    -5.05000
    2    2    2    1    2    1    -1.74000
    2    2    2    2    0    1    -3.33000
    2    2    2    2    2    1     0.09000
    3    3    3    3    0    1     0.24000
    4    3    4    3    1    1     0.73000
    4    3    4    3    2    1    -1.14000
    4    4    3    3    0    1    -5.05000
    4    4    4    3    2    1    -1.74000
    4    4    4    4    0    1    -3.33000
    4    4    4    4    2    1     0.09000
    1    3    1    3    0    1     0.24000
    1    3    1    3    1    0    -4.29000
    2    3    1    3    1    0     1.20000
    1    4    1    3    1    0    -1.20000
    2    3    2    3    1    0    -6.56000
    1    4    2    3    1    0     6.56000
    2    3    1    4    1    0     6.56000
    1    4    1    4    1    0    -6.56000
    2    3    2    3    1    1     0.73000
    1    4    2    3    1    1     0.73000
    2    3    1    4    1    1     0.73000
    1    4    1    4    1    1     0.73000
    2    3    2    3    2    0    -4.06000
    1    4    2    3    2    0    -4.06000
    2    3    1    4    2    0    -4.06000
    1    4    1    4    2    0    -4.06000
    2    3    2    3    2    1    -1.14000
    1    4    2    3    2    1     1.14000
    2    3    1    4    2    1     1.14000
    1    4    1    4    2    1    -1.14000
    2    4    1    3    0    1    -5.05000
    2    4    1    3    1    0     1.77000
    2    4    2    3    1    0     3.21000
    2    4    1    4    1    0    -3.21000
    2    4    2    3    2    1    -1.74000
    2    4    1    4    2    1     1.74000
    2    4    2    4    0    1    -3.33000
    2    4    2    4    1    0    -3.44000
    2    4    2    4    2    1     0.09000
    2    4    2    4    3    0    -7.27000
```

```
! file: LOGIN.COM
$@SYS$THEORY:[BROWN.SHELLGROU]LOGIN.SHL BROWN.SHELLGROU SYS$THEORY:
```

```
! file: LOGIN.SHL
$ AS :== ASSIGN
$ DI :== DIR/SIZE=ALL/PROT/DATE=(MODIFIED)
$!------------ Main OXBASH areas -------------------------------
$ AS 'P2'['P1']           SHDIR: ! *.DIR directory files
$ AS 'P2'['P1'.TXT]       SHTXT: ! OXBASH manaual SHELL.TXT
$ AS 'P2'['P1'.RSH]         RSH: ! (#) Area where OXBASH is run
$                                ! RSH can be reassigned to
$                                ! individual user areas
$ AS 'P2'['P1'.MAIN]       SHGR: ! Programs common to all such as
$                                ! MAINCOM.FOR, *.MAR, TITLE.FOR
$                                ! OPENX.FOR, OPENAREA.FOR
$ AS 'P2'['P1'.EXE]       SHEXE: ! *.EXE files
!$ AS 'P2'['P1'.MAIN.FOR] SHGRLIB: ! *.FOR for extended integer
$ AS 'P2'['P1'.MAIN.MAC]  SHGRLIB: ! *.MAR for extended integer
$ AS 'P2'['P1'.SHELL]      SHSH: ! SHELL
$ AS 'P2'['P1'.BASIS]      SHBA: ! BASIS
$ AS 'P2'['P1'.PROJ]       SHPR: ! PROJ
$ AS 'P2'['P1'.PRIOR]     SHPPR: ! PRIORPROJ (NOT FPS CONVERTED)
$ AS 'P2'['P1'.MATRIX]     SHMA: ! MATRIX,OPER
$ AS 'P2'['P1'.LANCZOS]    SHLA: ! LANCZOS
$ AS 'P2'['P1'.TRAMP]      SHTR: ! TRAMP,MVEC,TBTDOP
$ AS 'P2'['P1'.TRANS]     SHTRA: ! RTRD,TRANS
$ AS 'P2'['P1'.MISC]       SHMI: ! CHANGE,PN,HCM,SU3INT
$ AS 'P2'['P1'.LEV]       SHLEV: ! LEVEL
$
$ AS 'P2'['P1'.SPS]       SHSPS: ! *.SPS,*.INT inputs
$ AS 'P2'['P1'.INT]       SHINT: ! Additional *.INT inputs
$ AS 'P2'['P1'.OP]         SHOP: ! *.OP files generated by OPER
$ AS 'P2'['P1'.OP2]       SHOP2: ! *.OP2 generated by TBTDOP
$!---------------------------------------------------------------
$
$!------------ Other areas ---------------------------------------
$ AS 'P2'['P1'.MOSH]     SHMOSH: ! Moshinsky braket program for input to
$                                ! SHISOH:PRCO and SHPNCH:PRG
$ AS 'P2'['P1'.ISOH]     SHISOH: ! PRCO program and *.ISO files
$ AS 'P2'['P1'.ISOP]     SHISOP: ! ISOV program
$ AS 'P2'['P1'.PNCH]     SHPNCH: ! PRG program and *.PNC files
$ AS 'P2'['P1'.PNCP]     SHPNCP: ! PNCME programs
$                                ! double beta decay programs
$ AS 'P2'['P1'.TRD]       SHTRD: ! (#) *.TRD storage
$ AS 'P2'['P1'.OBD]       SHOBD: ! (#) *.OBD storage
$ AS 'P2'['P1'.INTH]     SHINTH: ! Programs to create *.INT files
$                                ! for delta function etc
$ AS 'P2'['P1'.LIBR]     SHLIBR: ! BAB library used in TRANS, ISOV etc
$!---------------------------------------------------------------

$!----------- Useful shorthand assignments ----------------------
$ SD      :== SET DEF              !
$ SHELL   :== RU SHEXE:SHELL       ! Runs OXBASH and creates *.COM files
$ PN      :== RU SHEXE:PN          ! Changes isospin *.INT and *.SPS
$                                  ! files into proton-neutron format
$ CHANGE  :== RU SHEXE:CHANGE      ! Combines *.INT files
$ HCM     :== RU SHEXE:HCM         ! Creates center of mass *.INT file
$ OPER    :== RU SHEXE:OPER        ! Creates *.OP from *.INT files
$ TBTDOP  :== RU SHEXE:TBTDOP      ! Creates *.OP2 files
$ RTRD    :== RU SHEXE:RTRD        ! Converts the *.TRD into *.OBD files
$ LEVEL   :== RU SHEXE:LEVEL       ! Creates level scheme from *.EIG files
$ MVEC    :== RU SHEXE:MVEC        ! Makes m-scheme vector file
$ TRANS   :== RU SHEXE:TRANS       ! Calculates moments, B(EL), B(ML)
```

```
$                                       ! and B(GT) from *.TRD files
$ CMOSH    :== RU SHEXE:CMOSH           ! See comment above after SHMOSH
$ PRDV     :== RU SHEXE:PRDV            ! Makes *.INT files delta function
$ PRCO     :== RU SHEXE:PRCO            ! Makes *.ISO files for ISOV
$ ISOV     :== RU SHEXE:ISOV            ! Isospin mixing from *.TRD and *.TBX
$ PRG      :== RU SHEXE:PRG             ! Makes *.PNC files for PNCME
$ PNCME    :== RU SHEXE:PNCME           ! Parity mixing from *.TRD and *.TBX
$ INTER    :== RU SHEXE:INTER           ! Converts *.INO to *.INT
$                                       ! (*.INO are the interaction files
$                                       ! in the "old" format)
$ PUT      :== @SHGR:PUT                ! Renames *.EXE file to area SHEXE:
$ HESH     :== @SHTXT:HESH              ! HELP for OXBASH (not up to date)
$ TECO     :== $SYS$SYSTEM:TECO TECO
$ MAKE     :== $SYS$SYSTEM:TECO MAKE
$!-------------------------------------------------------------------
$
$!---------- Special areas for FPS ------------------------------
!$ APF :== $SYS$SYSROOT:[SYSMGR.FPSCYC]APFTN64/DEBUG/LIS
!                                         SHORTEN AP COMPILER COMMAND
!$ AS 'P2'['P1'.MAIN.FPS]    SHGR: ! AREA FOR FPS LIBRARY ROUTINES
!$ AS 'P2'['P1'.MIAN.FPS] SHGRLIB: ! FPS VERSION OF MULTI-INTEGER FUNCTIONS
!$ AS 'P2'['P1'.OPFPS]   SHOPFPS: ! *.OP files generated by OPER (FPS)
!$ AS 'P2'['P1'.OP2FPS] SHOP2FPS: ! *.OP2 generated by TBTDOP (FPS)
!$ AS 'P2'['P1'.IMG]       SHIMG: ! (for storing OPT=3 *.EXE files)
!$ CHANGEOX :== RUN SHGR:INCLUDE
!$ CONVERTOX :== RUN SHGR:CONVERT
```

```
! file : LFALL.COM
$SET VERIFY
$SET DEF SYS$THEORY:[BROWN.SHELLGROU]
$@LOGIN
$SET DEF SHGR !-----------------------------------------------------
$@MAKEALL.SIZ                  ! Initiallizes the ALL.SIZ file
$@OPENFOR                      ! OPEN,TITLE Fortran library
$SET DEF SHGRLIB:
$@INTCOMP                      ! extended-integer Fortran library
$SET DEF SHSH !-----------------------------------------------------
$@SHELL                        ! SHELL      runs AUTOSHELL and creates the
$                              !            command (*.COM) files
$@SHGR:MAP SHELL               ! This puts the virtual memory allocation
$                              ! size into the file ALL.SIZ on area SHGR
$REN SHELL.EXE SHEXE:*.*       ! This puts *.EXE file on the SHEXE area
$SET DEF SHBA !-----------------------------------------------------
$@BASIS                        ! BASIS      Sets up m-scheme basis and
$@SHGR:MAP BASIS               !            creates *.BAS and *.INF files
$REN BASIS.EXE SHEXE:*.*
$SET DEF SHPR !-----------------------------------------------------
$                              ! Final version of PROJ (based on POSTPROJ)
$                              ! If PROJ fails try using PRIORPRPOJ
$@PREDICT                      ! PREDICT    creates *.PRD for PROJ input
$@SHGR:MAP PREDICT             !
$REN PREDICT.EXE SHEXE:*.*
$@PROJ                         ! PROJ       Projects out good J,T and
$@SHGR:MAP PROJ                !            creates *.PRJ files
$REN PROJ.EXE SHEXE:*.*
$SET DEF SHPPR !----------------------------------------------------
$@PRIORPROJ                    ! PRIORPROJ  Projects out good J,T and
$@SHGR:MAP PRIORPROJ           !            creates *.PRJ files
$REN PRIORPROJ.EXE SHEXE:*.*
$SET DEF SHMA !-----------------------------------------------------
$@MATRIX                       ! MATRIX     Calculates matrix and creates
$@SHGR:MAP MATRIX              !            *.MAT files
$REN MATRIX.EXE SHEXE:*.*
$@OPER                         ! OPER       Creates *.OP files
$@SHGR:MAP OPER
$REN OPER.EXE SHEXE:*.*
$SET DEF SHLA !-----------------------------------------------------
$@LANCZOS                      ! LANCZOS    Finds eigenvalue and creates
$@SHGR:MAP LANCZOS             !            *.EIG and *.LPE files
$REN LANCZOS.EXE SHEXE:*.*
$SET DEF SHTR !-----------------------------------------------------
$@MVEC                         ! MVEC       Creates *.VEC files
$@SHGR:MAP MVEC
$REN MVEC.EXE SHEXE:*.*
$@TRAMP                        ! TRAMP      Creates *.TRD, *.TBD files
$@SHGR:MAP TRAMP               !            etc
$REN TRAMP.EXE SHEXE:*.*
$@TBTDOP                       ! TBTDOP     Creates *.OP2 files
$@SHGR:MAP TBTDOP
$REN TBTDOP.EXE SHEXE:*.*
$SET DEF SHTRA !----------------------------------------------------
$@LFTRANS                      ! TRANS      Reads *.TRD files to obtain ME
$@SHGR:MAP TRANS
$REN TRANS.EXE SHEXE:*.*
$@LFRTRD                       ! RTRD       Converts *.TRD (binary) into
$@SHGR:MAP RTRD                !            *.OBD (formatted )files
$REN RTRD.EXE SHEXE:*.*
```

```
$SET DEF SHMI !-------------------------------------------------------
$@CHANGE                        ! CHANGE     Manipulates *.INT files
$@SHGR:MAP CHANGE
$REN CHANGE.EXE SHEXE:*.*
$@HCM                           ! HCM        Makes center of mass *.INT
$@SHGR:MAP HCM                  !            file
$REN HCM.EXE SHEXE:*.*
$@PN                            ! PN         Converts *.INT and *.SPS files
$@SHGR:MAP PN                   !            in isospin formalism into files
$REN PN.EXE SHEXE:*.*           !            in proton-neutron formalism
$@SU3INT                        ! SU3INT     Makes an interaction which
$@SHGR:MAP SU3INT               !            has eigenvectors diagonal
$REN SU3INT.EXE SHEXE:*.*       !            in the SU3 quantum numbers
$SET DEF SHLEV !-----------------------------------------------------
$@LEVEL                         ! LEVEL      Makes energy level scheme
$@SHGR:MAP LEVEL
$REN LEVEL.EXE SHEXE:*.*
$@SHDIR:LFBAB ! Special programs by BAB for TBME, isospin mixing etc
$SET DEF SHGR  !-----------------------------------------------------
$COPY SHGR:ALL.SIZ,SHGRLIB:MACFOR.INC,SHGR:PARAM.INC ALLS.SIZ
$                               ! Finish construction of size file
$SET DEF SHDIR !-----------------------------------------------------
$@CLEANUP                       ! clean up - see CLEANUP.COM
$SET NOVERIFY
```

```
! file: LFBAB.COM
$SET DEF SHLIBR !-----------------------------------------------------
$@LFLIBR                        ! BAB Library
$SET DEF SHMOSH !-----------------------------------------------------
$@LFCMOSH                       ! BAB Moshinsky bracket program
$REN CMOSH.EXE SHEXE:*.*
$SET DEF SHINTH !-----------------------------------------------------
$@LFPRDV                        ! BAB Delta function TBME
$REN PRDV.EXE SHEXE:*.*
$SET DEF SHISOH !-----------------------------------------------------
$@LFPRCO                        ! BAB Coulomb two-body matrix elements
$REN PRCO.EXE SHEXE:*.*
$SET DEF SHISOP !-----------------------------------------------------
$@LFISOV                        ! BAB isospin mixing matrix element program
$REN ISOV.EXE SHEXE:*.*
$SET DEF SHPNCH !-----------------------------------------------------
$@LFPRG                         ! BAB parity mixing two-body matrix element
$REN PRG.EXE SHEXE:*.*
$SET DEF SHPNCP !-----------------------------------------------------
$@LFPNCME                       ! BAB parity mixing matrix element program
$REN PNCME.EXE SHEXE:*.*
```

file: ALL.SIZ

16-NOV-1986 23:56  VAX-11 Linker V04-00
[BROWN.SHELLGROU.SHELL]SHELL.OBJ;1  16-Nov-1986 23:53  VAX FORTRAN V4.4-17
Virtual memory allocated:  (85504. bytes, 167. pages)

16-NOV-1986 23:58  VAX-11 Linker V04-00
[BROWN.SHELLGROU.BASIS]BASIS.OBJ;1  16-Nov-1986 23:56  VAX FORTRAN V4.4-17
Virtual memory allocated:  (1507328. bytes, 2944. pages)

16-NOV-1986 23:59  VAX-11 Linker V04-00
[BROWN.SHELLGROU.PROJ]PREDICT.OBJ;1  16-Nov-1986 23:58  VAX FORTRAN V4.4-17
Virtual memory allocated:  (5723136. bytes, 11178. pages)

17-NOV-1986 00:01  VAX-11 Linker V04-00
[BROWN.SHELLGROU.PROJ]PROJ.OBJ;1  16-Nov-1986 23:59  VAX FORTRAN V4.4-17
Virtual memory allocated:  (6995456. bytes, 13663. pages)

17-NOV-1986 00:03  VAX-11 Linker V04-00
[BROWN.SHELLGROU.PRIOR]FEDUP.OBJ;1  17-Nov-1986 00:02  VAX FORTRAN V4.4-17
Virtual memory allocated:  (7450112. bytes, 14551. pages)

17-NOV-1986 00:04  VAX-11 Linker V04-00
[BROWN.SHELLGROU.MATRIX]MATRIX.OBJ;1 17-Nov-1986 00:03  VAX FORTRAN V4.4-17
Virtual memory allocated:  (6100992. bytes, 11916. pages)

17-NOV-1986 00:05  VAX-11 Linker V04-00
[BROWN.SHELLGROU.MATRIX]OPER.OBJ;1  17-Nov-1986 00:04  VAX FORTRAN V4.4-17
Virtual memory allocated:  (2112000. bytes, 4125. pages)

17-NOV-1986 00:07  VAX-11 Linker V04-00
[BROWN.SHELLGROU.LANCZOS]LANC.OBJ;1  17-Nov-1986 00:06  VAX FORTRAN V4.4-17
Virtual memory allocated:  (2034688. bytes, 3974. pages)

17-NOV-1986 00:07  VAX-11 Linker V04-00
[BROWN.SHELLGROU.TRAMP]MVEC.OBJ;1  17-Nov-1986 00:07  VAX FORTRAN V4.4-17
Virtual memory allocated:  (293376. bytes, 573. pages)

17-NOV-1986 00:10  VAX-11 Linker V04-00
[BROWN.SHELLGROU.TRAMP]TRAMP.OBJ;1  17-Nov-1986 00:08  VAX FORTRAN V4.4-17
Virtual memory allocated:  (2210816. bytes, 4318. pages)

17-NOV-1986 00:11  VAX-11 Linker V04-00
[BROWN.SHELLGROU.TRAMP]TBTDOP.OBJ;1  17-Nov-1986 00:11  VAX FORTRAN V4.4-17
Virtual memory allocated:  (207360. bytes, 405. pages)

17-NOV-1986 00:14  VAX-11 Linker V04-00
[BROWN.SHELLGROU.TRANS]TRANS.OBJ;1  17-Nov-1986 00:12  VAX FORTRAN V4.4-17
Virtual memory allocated:  (71680. bytes, 140. pages)

17-NOV-1986 00:15  VAX-11 Linker V04-00
[BROWN.SHELLGROU.TRANS]RTRD.OBJ;1  17-Nov-1986 00:14  VAX FORTRAN V4.4-17
Virtual memory allocated:  (375296. bytes, 733. pages)

17-NOV-1986 00:16  VAX-11 Linker V04-00
[BROWN.SHELLGROU.MISC]CHANGE.OBJ;1  17-Nov-1986 00:15  VAX FORTRAN V4.4-17
Virtual memory allocated:  (26112. bytes, 51. pages)

17-NOV-1986 00:17  VAX-11 Linker V04-00
[BROWN.SHELLGROU.MISC]HCM.OBJ;1  17-Nov-1986 00:16  VAX FORTRAN V4.4-17

```
                Virtual memory allocated:  (10752. bytes, 21. pages)

 17-NOV-1986 00:18   VAX-11 Linker V04-00
[BROWN.SHELLGROU.MISC]PN.OBJ;1       17-Nov-1986 00:17  VAX FORTRAN V4.4-17
                Virtual memory allocated:  (161280. bytes, 315. pages)

 17-NOV-1986 00:18   VAX-11 Linker V04-00
[BROWN.SHELLGROU.MISC]SU3INT.OBJ;1   17-Nov-1986 00:18  VAX FORTRAN V4.4-17
                Virtual memory allocated:  (12288. bytes, 24. pages)

 17-NOV-1986 00:19   VAX-11 Linker V04-00
[BROWN.SHELLGROU.LEV]LEVEL.OBJ;1     17-Nov-1986 00:18  VAX FORTRAN V4.4-17
                Virtual memory allocated:  (99840. bytes, 195. pages)

CWEO PARAMETERS USED WITH MACRO ROUTINES
CWEO IFOR=0 -> MACRO ROUTINES
      PARAMETER (MAX_BIT_NLWRDS=32,IMAC_NSPS=32,IFOR=0,IFPS=0)
      PARAMETER (MAX_NOSC=4)
      PARAMETER (N_WORDS=12)
      PARAMETER (MAX_NJL=N_WORDS)
      PARAMETER (MAX_NLWRDS=3)
      PARAMETER (MAX_NSPS=MAX_NLWRDS*MAX_BIT_NLWRDS)
      PARAMETER (NLWRDS_BLOCK=64)
      PARAMETER (MAX_NPART=600)
      PARAMETER (MAX_MSCHEME=25000)
      PARAMETER (NO_JVAL=10)
      PARAMETER (NO_TVAL=3)
      INTEGER TWELVE
      PARAMETER (TWELVE=3+NO_TVAL*NO_JVAL)
      PARAMETER (MAX_NJT=700)
      PARAMETER (MAX_PRED=600)
      PARAMETER (MAX_IPROT=24)
      PARAMETER (MAX_NPTCLE=30)
      PARAMETER (MAX_MAT=3000)
```

```
C file: PARAM.HLP
CBAB COMMENTS FOR PARAM.INC

CA  THESE PARAMETER STATEMENTS ARE THE ONLY THING TO CHANGE IN ORDER TO
CA  CHANGE THE DIMENSIONS. SEE IF THERE IS ANY OTHER IN THE SUBROUTINES
CA  MEANING:
CB  IMAC_NSPS       PROGRAMS WILL USE MACRO MULTI-INTERGER SUBROUTINES
CB                  WHEN NSPS IS GREATER THAN IMAC_NSPS, OTHERWISE
CB                  SOME FORTRAN (WHICH IS FASTER) WILL BE USED
CA  MAX_NJL         MAXIMUM NUMBER OF J-LEVELS, one J-level per byte
CA  MAX_NLWRDS      MAXIMUM NUMBER OF LONG WORDS
CA                  (ONE LONG WORD IS MAX_BIT_NLWRDS)
CE  MAX_BIT_NLWRDS  NUMBER OF BITS TO USED IN ONE LONG WORD
CA  MAX_NSPS        MAXIMUN NUMBER OF SINGLE PARTICLE STATES
CA                  (one s.p.state per bit)
CA  MAX_NPART       MAXIMUM NUMBER OF PARTITIONS
CA  NLWRDS_BLOCK    NUMBER OF LONG WORDS PER EACH DISK BLOCK. IN THE AP,
CA                  THIS NUMBER IS 64. IF A RECORD WITH LESS THAN 64
CA                  LONG WORDS IS WRITTEN ONTO DISK, THE REMAING IS FILLED
CA                  WITH ZER0S THEREFORE THE DISK SPACE WOULD NOT BE
CA                  CORRECTLY USED. THIS IS A COMPROMISE WITH THE BASIS
CA                  DIMENSION
CA  MAX_NOSC        MAXIMUM NUMBER OF MAJOR SHELLS. USED IN MAXMIM.FOR
CA                  AND NEWPATR
CA  MAX_MSCHEME     MAX_MSCHEME DIMENSION FOR A GIVEN PARTITION
CA  MAX_IPROT       MAXIMUM NO. OF PROTON PARTITIONS FOR A GIVEN PARTITION

CBAB                                        1 2 3 4 5 6 7 8 9 A B C D
CBAB                                        -----------------------
CBAB  PROGRAMS WHERE USED                   S B P P O M L M T C T I H
CBAB                                        H A R R P A A V R H B N C
CBAB                                        E S E O E T N E A A T T M
CBAB                                        L I D J R R C C M N D E
CBAB                                        L S I     I Z   P G O R
CBAB                                          C     X O     E P
CBAB                                          T       S

CWEO *** THESE PARAMETER STATEMENTS ARE NOW CONTAINED IN 'SHGRLIB:MACFOR.IN(
CWEO      PARAMETER (MAX_BIT_NLWRDS=31)       4         9
CWEO      PARAMETER (IMAC_NSPS=31)
CWEO      PARAMETER (IFOR=0)
CWEO ***************************************************************

      INCLUDE 'SHGRLIB:MACFOR.INC'
      PARAMETER (MAX_NOSC=4)            !    1 2 3 4 5 6 7 8 9   B   D
      PARAMETER (N_WORDS=12)            !    1 2 3 4 5 6 7 8 9 A B C D
      PARAMETER (MAX_NJL=N_WORDS)       !    1 2 3 4 5 6 7 8 9 A B C D
      PARAMETER (MAX_NLWRDS=3)          !      2 3 4 5 6 7 8 9   B
      PARAMETER (MAX_NSPS=MAX_NLWRDS*MAX_BIT_NLWRDS)
                                        !      2 3 4 5 6     9
      PARAMETER (NLWRDS_BLOCK=64)       !      2 3 4   6   8 9
      PARAMETER (MAX_NPART=600)         !      2 3 4   6   8 9   B
CBAB      PARAMETER MAX_MSCHEME=25000   !      2 3 4   6   8 9
CBAB      PARAMETER MAX_MSCHEME=100000  !      2 3 4   6   8 9
      PARAMETER (MAX_MSCHEME=25000)     !      2 3 4   6   8 9
      PARAMETER (NO_JVAL=10)
      PARAMETER (NO_TVAL=3)
      INTEGER TWELVE                    !        2 3 4   6     9
      PARAMETER (TWELVE=3+NO_TVAL*NO_JVAL)
CBAB (Historically NO_JVAL=NO_TVAL=3 and hence TWELVE=12)
```

```
CBAB       PARAMETER MAX_NJT=600        !          3 4    6
      PARAMETER (MAX_NJT=700)           !          3 4    6
CEO       PARAMETER MAX_PRED=800        !          3 4
CEO CHANGED 17 SEPT. 1985. NEEDED MAX_PRED = 855 FOR 7/2-
CEO STATE IN A=41 (D3/2-F7/2) MAX_PRED=860
CEO IS APPORXIMATELY AS BIG AS MAX_PRED CAN BE
CEO FOR PROJ TO COMPILE!!!!
CBAB       PARAMETER MAX_PRED=860       !          3 4
      PARAMETER (MAX_PRED=600)          !          3 4
      PARAMETER (MAX_IPROT=24)          !          3 4
CWEO       PARAMETER (MAX_IPROT=48)
      PARAMETER (MAX_NPTCLE=30)         !          3
      PARAMETER (MAX_MAT=3000)          !        2
      LOGICAL LIB_LEX,LIB_LTX,LIB_GEX,LIB_GTX,LIB_NEX,LIB_EQX,
     1LIB_BTX,LIB_BTSX,LIB_BTCX,LIB_WHERE_BIT

CBAB COMMENTS FOR DATA.INC
C AREA_RUN NOT USED IN WEO VERSION
      CHARACTER*13 AREA_RUN             !      1 2 3    5 6 7 8 9
      CHARACTER*13 AREA_EXE             !      1
      CHARACTER*13 AREA_SPS             !      1 2 3 4 5 6 7 8 9 A B C D
      CHARACTER*13 AREA_INT             !              5
      CHARACTER*13 AREA_OP              !              5 6
      CHARACTER*13 AREA_OP2             !      1                9   B
      DATA AREA_EXE/'OXEXE:'/    ! SUBDIRECTORY FOR *.EXE FILES
      DATA AREA_SPS/'OXSPS:'/    ! SUBDIRECTORY FOR *.SPS AND *.INT FILES
      DATA AREA_INT/'OXINT:'/    ! SUBDIRECTORY FOR ADDITIONAL *.INT FILES
      DATA AREA_OP/'OXOP:'/      ! SUBDIRECTORY FOR *.OP  FILES
      DATA AREA_OP2/'OXOP2:'/    ! SUBDIRECTORY FOR *.OP2 FILES
      DATA IVERS/2/
```

```
! file: LABEL.DAT
!LADEL.DAT is an input to SHELL and read in the subroutine RLABEL.
!(A row with a "!" in the first column (like this one) will be
!skipped over in the program RLABEL.FOR and can thus be used for
!information.) When SHELL is run, only the following combinations of
!model-space and interaction can be specified. A new combination
!must be entered in this list before running SHELL.
!
!IMPORTANT: If you change a *.SPS file or one of the two-body
!           matrix elements in a *.INT file be sure to rerun OPER
!           or delete the old *.OP files so that OPER will run
!           automatically in SHELL. LANCZOS uses the single-particle
!           energy (SPE) from the *.INT files, so OPER need not be
!           rerun if only a SPE is changed.
!
!(1) Model space (*.SPS file) name (single-particle state file name)
!(2) Interaction (*.INT file) name
!(3) Model space code used for the output files
!    (The labels "P" and "N" should be reserved for charge-dependent
!    interations names which are set up in the subroutine PN_INT)
!(4) Interaction code used for the output files
!(5) References and information  (* indicates an undocumented test
!    interaction used by B. A. Brown.)
!
!   Reference list for model spaces (* is reserved; ? is not yet used)
!   (3) (1)       (3) (1)       (3) (1)       (3) (1)
!    A   P         H   HO-F5P    O   F7P3      V   ?
!    B   SD        I   GL        P   *         W   ?
!    C   ZBM       J   SLG       Q   N82       X   ?
!    D   PSD       K   HASP      R   A80       Y   ?
!    E   F7        L   SDPF      S   SPSD      Z   ?
!    F   FP        M   PFG       T   SPSDPF
!    G   D3F7      N   *         U   I13
!
!    List of available model-space and interaction combinations
!(1)    (2)     (3) (4)   (5)

                              ! ------------------------------------------
                              ! P Model Space
                              ! 1p1/2,1p3/2 orbits
P       CKPOT    A   A       ! (8-16)POT  COHEN-KURATH NP A73, 1(1965)
P       CKI      A   B       ! (6-16)TBME COHEN-KURATH NP A73, 1(1965)
P       CKII     A   C       ! (8-16)TBME COHEN-KURATH NP A73, 1(1965)
P       MP       A   M       ! * D.J.MILLENER (PRIVATE COMMUNICATION NOV 1984)
P       WP       A   W       ! * B.H.WILDENTHAL AND C.CHITWOOD
P       CKIIM    A   X       ! * MODIFIED CKII
P       SU3P     A   S       ! SU3 INT (from Millener)
                              ! Put IRAN=10 in LANCZOS input to break degeneracy

                              ! ------------------------------------------
                              ! PPN Model Space
                              ! P1p1/2,P1p3/2,N1p1/2,N1p3/2 orbits
PPN     CKPOTPN  A   U        ! (8-16)POT  COHEN-KURATH NP A73, 1(1965)

                              ! ------------------------------------------
                              ! SD Model Space
                              ! 1d3/2,1d5/2,2s1/2 orbits
SD      KUOSD    B   A       ! RENORMALIZED KUO NPA103, 71 (1967)
SD      BKUOSD   B   B       ! BARE KUO NPA103, 71 (1967)
SD      PW       B   P       ! (PW) PREEDOM-WILDENTHAL PR C6, 1633 (1972)
```

```
SD      CW      B   C  ! (CW) CHUNG-WILDENTHAL A=17-28 INTERACTION
SD      SDM     B   M  ! WILDENTHAL-MCGRORY MSDI (PRC4, 1708 (1971))
SD      W       B   W  ! (W) WILDENTHAL'S A=17-39 "USD" INT (JULY 1982)
SD      HBGF    B   H  ! * BROWN-RICHTER
SD      SU3SD   B   S  ! SU3 INT
                       ! Put IRAN=10 in LANCZOS input to break degeneracy
SD      WZ      B   Z  ! W INT WITH TBME SET TO ZERO

                       ! ---------------------------------------------
                       ! SDW Model Space
                       ! 1d5/2,2s1/2,1d3/2 orbits
SDW     HBUSD   B   K  ! * BROWN-RICHTER 0.35 MASS DEPENDENCE
SDW     TEST    B   Z  ! *

                       ! ---------------------------------------------
                       ! SDPN Model Space
                       ! P0d3/2,P0d5/2,P1s1/2,N0d3/2,N0d5/2,N1s1/2 orbits
                       ! (SEE SUBROUTINE PN_INT FOR RELABELING OF *.MAT
                       ! *.EIG, *.LPE AND *.GND FILES)
SDPN    WPN     B   U  ! "USD" INT PN FORMALSIM
SDPN    WCDPN   B   U  ! "USD" PLUS COULOMB AND CHARGE DEPENDENT RHO
                       ! ORMAND AND BROWN (NPA440, 274 (1985))
SDPN    WWCDPN  B   U  ! "USD" PLUS COULOMB PLUS "USD" ISOTENSOR
SDPN    SDMPN   B   V  ! MSDI (PRC4, 1708 (1971)) IN PN FORMALISM
SDPN    SDMCDPN B   V  ! MSDI (PRC4, 1708 (1971)) PLUS INC INTERACTION
                       ! OF TOWNER & HARDY (NPA205, 33 (1973))
SDPN    WCDNP   B   U  ! SAME AS WCDPN  EXCEPT P-N LABELS INTERCHANGED
SDPN    WWCDNP  B   U  ! SAME AS WWCDPN EXCEPT P-N LABELS INTERCHANGED
SDPN    SDMCDNP B   V  ! SAME AS MCDPN  EXCEPT P-N LABELS INTERCHANGED


                       ! ---------------------------------------------
                       ! ZBM Model Space
                       ! 1p1/2,1d5/2,2s1/2 orbits
ZBM     ZBMI    C   A  ! (I)  ZUKER-BUCK-MCGRORY PRL 21, 39 (1968)
ZBM     ZBMII   C   B  ! (II) ZUKER-BUCK-MCGRORY PRL 21, 39 (1968)
ZBM     REWIL   C   R  ! REEHAL-WILDENTHAL PART AND NUCL 6, 137 (1973)
                       ! F-INT IN PRC7, 974 (1973)
ZBM     ZWM     C   Z  ! MCGRORY-WILDENTHAL PR C7, 974 (1973) (Z-INT)
ZBM     ZBMO    C   O  ! * THE OXFORD AVILA-AGUIRRE BROWN INTERACTION
ZBM     SU3ZBM  C   S  ! SU3 INT
                       ! Put IRAN=10 in LANCZOS input to break degeneracy

                       ! ---------------------------------------------
                       ! ZBMPN Model Space
                       ! P1p1/2,P1d5/2,P2s1/2,N1p1/2,N1d5/2,N2s1/2 orbits
ZBMPN   REWILPN C   P  ! REEHAL-WILDENTHAL PART AND NUCL 6, 137 (1973)
ZBMPN   REWILC  C   C  ! REWILPN + ZBMC

                       ! ---------------------------------------------
                       ! PSD Model Space
                       ! 1p1/2,1p3/2,1d3/2,1d5/2,2s1/2 orbits
                       ! SHELL will include the center of mass
                       ! Hamiltonian PSDCM in the list of inputs
                       ! to MATRIX
PSD     PSDMK   D   M  ! MK(P-SD) + PW(SD) + CKPOT(P)
                       ! Millener and Kurath, NPA255, 315 (1975)
PSD     REWILE  D   Q  ! * REWIL + DJMPW
PSD     SU3PSD  D   S  ! SU3 INT
                       ! Put IRAN=10 in LANCZOS input to break degeneracy
```

```
PSD       PSDZ       D    Z   ! * PSDZ = PSDMK WITH 1P1H ME SET TO ZERO

                                ! ---------------------------------------------
                                ! PSDPN Model Space
                                ! P 1p1/2,1p3/2,1d3/2,1d5/2,2s1/2 orbits
                                ! N 1p1/2,1p3/2,1d3/2,1d5/2,2s1/2 orbits
                                ! SHELL will include the center of mass
                                ! Hamiltonian PSDCMPN in the list of inputs
                                ! to MATRIX
PSDPN     PSDMWKPND     U     ! * MP(P) + W(SD) + MK(OTHER)

                                ! ---------------------------------------------
                                ! SPSD Model Space
                                ! 1s1/2,1p3/2,1p1/2,1d5/2,1d3/2,2s1/2 orbits
                                ! SHELL will include the center of mass
                                ! Hamiltonian SPSDCM in the list of inputs
                                ! to MATRIX
SPSD      SPSDMK   S    M     ! * DJMPW + MK INT FOR 0S ORBIT
SPSD      SPSDMWK  S    W     ! * MP(P) + W(SD) + MK(OTHER)

                                ! ---------------------------------------------
                                ! * SPSDPF Model Space
                                ! 1s1/2,1p3/2,1p1/2,1d5/2,1d3/2,2s1/2
                                ! 1f7/2,1F5/2,2P3/2,2P1/2 orbits
                                ! SHELL will include the center of mass
                                ! Hamiltonian SPSDPFCM in the list of inputs
                                ! to MATRIX
SPSDPF    SG16F    T    B     ! * SG16(Skyrme SGII)
SPSDPF    SG40F    T    H     ! * SG40(Skyrme SGII)
SPSDPF    SGF      T    S     ! * SG16-40(Skyrme SGII)

                                ! ---------------------------------------------
                                ! D3F7 Model Space
                                ! 1d3/2,1f7/2 orbits
D3F7      W0       G    S     ! WILDENTHAL AS USED IN PRL 33, 233 (1974)
                                ! BY SETH ET AL
D3F7      W4       G    W     ! WILDENTHAL'S MORE RECENT MODIFICATION TO W0
D3F7      SAS      G    A     ! SAKAKURA-ARIMA-SEBE PL 61B, 335(1976)
D3F7      FEP      G    F     ! FEDERMAN-PITTEL PR 186, 1106 (1969)
D3F7      FEPP     G    P     ! * FEDERMAN-PITTEL PR 186, 1106 (1969) 0.6XFFDD
D3F7      FEPPZ    G    Z     ! *  " " " ZERO PH INTERACTION
D3F7      FEPQ     G    Q     ! * FEDERMAN-PITTEL PR 186, 1106 (1969) 0.6XFFDD0
D3F7      HW       G    H     ! * Hsieh-Wildenthal interaction
                                ! * (private commumication)

                                ! ---------------------------------------------
                                ! D3F7PN Model Space
                                ! P1d3/2,P1f7/2,N1d3/2,N1f7/2 orbits
                                ! (SEE SUBROUTINE PN_INT FOR RELABELING OF *.MAT
                                ! *.EIG, *.LPE AND *.GND FILES)
D3F7PN    HWPN     G    U     ! * "HW" INT PN FORMALSIM
D3F7PN    HWCDPN   G    U     ! * "HW" PLUS COULOMB PLUS "HW" ISOTENSOR
D3F7PN    HWCDNP   G    U     ! * SAME AS HWCDPN EXCEPT P-N LABELS INTERCHANGED

                                ! ---------------------------------------------
                                ! HASP Model Space
                                ! 1d3/2,2s1/2,2p3/2,1f7/2 orbits
HASP      HASP     K    G     ! + parity int from HASPER, PRC19, 1482 (1979)
HASP      HASN     K    H     ! - parity int from HASPER, PRC19, 1482 (1979)
HASP      VPNP     K    P     ! MSDI int: van-der-Poel Nucl. Phys. A373, 81 (82)
```

```
HASP    VPTH     K   Q  ! MSDI int: van-der-Poel Thesis, p.50 (JAB)

                        ! ---------------------------------------------
                        ! * SDPF Model Space
                        ! * 1d5/2,1d3/2,2s1/2,1f7/2,1f5/2,2p3/2,2p1/2 orbits
SDPF    SDPFVP   L   P  ! * SDPFW + SDPFVH + SDPFVPNP
                        !   (SDPF)W = (SD)W FOR A=40
                        !   (SDPF)VH = (FP)VH
                        !   (SDPF)VPNP = MSDI int from van-der-Poel
                        !                Nucl. Phys. A373, 81 (82)
SDPF    SDPFMK   L   M  ! * SDPFW + SDPFVH + SDPFMK
                        !   (SDPF)W = (SD)W FOR A=40
                        !   (SDPF)VH = (FP)VH
                        !   (SDPF)MW  = MILLENER-KURATH
SDPF    SDPFMW   L   H  ! * SDPFW + SDPFVHW + SDPFMW  (JAB)
                        !   (SDPF)W = (SD)W FOR A=40
                        !   (SDPF)VHW = (SDPF)VH + MODIFIED BY (D3F7)HW
                        !   (SDPF)MW  = MILLENER-KURATH D3-F7 EXACT FIT
                        !               TO 1P1H STATES IN 40CA
SDPF    SDPFM40  L   I  ! Phys. Rev. C34, 1031 (1986)
                        !   SDPFW + SDPFVHB + SDPFM40  (JAB)
                        !   (SDPF)VHB = (SDPF)VHW
                        !               + (F7/2)**2 MADE TO FIT 42SC
                        !   (SDPF)M40 = MILLENER-KURATH D3-F7 EXACT FIT
                        !               TO 1P1H STATES IN 40CA EXCEPT
                        !               FOR 3- T=0

                        ! ---------------------------------------------
                        ! F7 Model Space
                        ! 1f7/2 orbits
F7      F7MBZ    E   M  ! J. D. McCullen, B. F. Bayman and L. Zamick
                        ! Phys. Rev. 134, B515 (1963).
F7      F742     E   A  ! KUTCHERA et al. "42SC" R NUOVO CIM, VOL-1 NO-12
F7      F748     E   B  ! KUTCHERA et al. "48SC" R NUOVO CIM, VOL-1 NO-12
F7      F754     E   C  ! KUTCHERA et al. "54CO" R NUOVO CIM, VOL-1 NO-12

                        ! ---------------------------------------------
                        ! F7PN Model Space
                        ! P1f7/2,N1f7/2 orbits
F7PN    F742PN   E   U  ! KUTCHERA PN INT BASED ON 42TI, 42SC AND 42CA
F7PN    F748PN   E   V  ! KUTCHERA PN INT BASED ON 50TI, 48SC AND 50TI
F7PN    F748BPN  E   W  ! KUTCHERA PN INT BASED ON 50TI, 48SC AND 46CA

                        ! ---------------------------------------------
                        ! HO Model Space
                        ! P1f7/2,N2p3/2,N1f5/2,N2p1/2 orbits
                        ! ---------------------------------------------
HO      HO       H   H  ! H.Horie and K.Ogawa, Prog. of Theoretical
                        ! Physics 46, 439 (1971), Nucl. Phys. A216,
                        ! 407 (1973).

                        ! ---------------------------------------------
                        ! F7P3 Model Space
                        ! 1f7/2,2p3/2 orbits
F7P3    FPN      O   A  ! Modified FPV interaction

                        ! ---------------------------------------------
                        ! F7P3PN Model Space
                        ! P1f7/2,2p3/2 orbits
                        ! N1f7/2,2p3/2 orbits
```

```
F7P3PN   FPNPN    O   U  ! * Modified FPV interaction
F7P3PN   FPNCDPN  O   U  ! * Modified FPV interaction
                           !  INC interaction included
F7P3PN   FPNCDNP  O   U  ! * Modified FPV interaction
                           !  INC interaction included
                           !  protons and neutrons switched

                           ! ------------------------------------------
                           ! FP Model Space
                           ! 1f7/2,2p3/2,1f5/2,2p1/2 orbits
FP       FPUSD    F   G  ! *
FP       HBUFP    F   A  ! * BROWN-RICHTER 0.35 MASS DEPENDENCE
FP       FPE      F   E  ! *
FP       FPM      F   M  ! MOOY INTERACTION
                           !  ZP A312, 59 (1983) AND PRIVATE COMMUNICATION
FP       FPY      F   Y  ! F7P3F5P1 + YOKOYAMA "D" INTERACTION
                           !  PHYS REV C31, 1012 (1985) AND PRIV COMM
                           !  PLUS MSDI(0.25,0.53,-1.09,.50)
                           !  FOR ALL OTHER MATRIX ELEMENTS FROM THE
                           !  1ST COLUMN IN TABLE 1 OF
                           !  KOOPS AND GLAUDEMANS, Z. PHYS. A280, 181 (1977)
FP       FPV      F   V  ! VAN HEES INTERACTION Z. PHYS. A303, 267 (1981)
                           !  PLUS MSDI(0.25,0.53,-1.09,.50)
                           !  FOR ALL OTHER MATRIX ELEMENTS FROM THE
                           !  1ST COLUMN IN TABLE 1 OF
                           !  KOOPS AND GLAUDEMANS, Z. PHYS. A280, 181 (1977)
FP       FPVH     F   H  ! SAME AS FPV BUT REPLACING F5P1P3 WITH
                           !  WITH VAN HIENEN - CHUNG - WILDENTHAL
                           !  INTERACTION NPA269, 159 (1976)
FP       FPV2     F   B  ! * FPV + 2 MEV ADDED ONTO F5/2 ORBIT
                           !  TO GET BETTER 1+ PH SPECTRUM IN 48CA
                           !  SEE B.D.ANDERSON ET AL, PRC31, 1161 (1985)
FP       KBFP     F   K  ! RENORMALIZED KUO-BROWN, NPA114, 241 (1986)
FP       BKBFP    F   L  ! BARE KUO-BROWN, NPA114, 241 (1986)

                           ! ------------------------------------------
                           ! FPPN Model Space
                           ! P1f7/2,2p3/2,1f5/2,2p1/2 orbits
                           ! N1f7/2,2p3/2,1f5/2,2p1/2 orbits
                           ! (SEE SUBROUTINE PN_INT FOR RELABELING OF *.MAT
                           ! *.EIG, *.LPE AND *.GND FILES)
FPPN     FPVPN    F   U  ! * "FPV" INT PN FORMALSIM
FPPN     FPVCDPN  F   U  ! * "FPV" PLUS COULOMB PLUS "FPV" ISOTENSOR
FPPN     FPVCDNP  F   U  ! * SAME AS FPVCDPN EXCEPT P-N LABELS INTERCHANGED

                           ! ------------------------------------------
                           ! FPN Model Space
                           ! N1f7/2,2p3/2,1f5/2,2p1/2 orbits
FPN      FPVN     F   X  ! VAN HEES INTERACTION Z. PHYS. A303, 267 (1981)

                           ! F5P Model Space
                           ! 1f5/2,2p3/2,2p1/2 orbits
F5P      F5PV     H   V  ! MSDI(0.25,0.53,-1.09,.50)
                           ! KOOPS AND GLAUDEMANS, Z. PHYS. A280, 181 (1977)

                           ! ------------------------------------------
                           ! GL Model Space
                           ! P2p1/2,P1g9/2,N3s1/2,N2d5/2 orbits
GL       GL       I   A  ! GLOECKNER INTERACTION (FINAL)
GL       GLB      I   B  ! GLOECKNER INTERACTION (N=51)
```

```
                                  ! ------------------------------------------
                                  ! SLG Model Space
                                  ! P2p1/2,P1g9/2,N2p1/2,N1g9/2 orbits
SLG      SLGM     J    A         ! "seniority-fit total-energy" PP and PN SLG int
                                  ! F.J.D.Serduke, R.D.Lawson and D.H.Gloeckner
                                  ! NPA256, 45 (1976)
                                  ! * NN INTERACTION FROM BAB
SLG      SLGMT0   J    Z         ! "T=0" PP and PN SLG int
                                  ! * NN INTERACTION FROM BAB
SLG      SLGML    J    L         ! "seniority-fit levels" PP and PN SLG int
                                  ! * NN INTERACTION FROM BAB
SLG      SLGO     J    B         ! PP AND PN SERDUKE-LAWSON-GLOECKNER INTERACTION
                                  ! * NN INTERACTION FROM ZP314, 205 (1983)
                                  ! (LAWSON PRIVATE COMMUNICATION)
                                  ! ------------------------------------------
                                  ! SLGT Model Space
                                  ! 2p1/2,1g9/2
SLGT     SLGT     J    T         ! SPACE SERDUKE-LAWSON-GLOECKNER INTERACTION
                                  ! ------------------------------------------
                                  ! * FPGPN Model Space
                                  ! P1f7/2,1f5/2,2p3/2,2p1/2,1g9/2,1g7/2 orbits
                                  ! N1f7/2,1f5/2,2p3/2,2p1/2,1g9/2,2g7/2 orbits
PFGPN    PFGPN    M    U         ! *
                                  ! ------------------------------------------
                                  ! * FPG Model Space
                                  ! 1f7/2,1f5/2,2p3/2,2p1/2,1g9/2,1g7/2 orbits
PFG      PFG      M    M         ! *
                                  ! ------------------------------------------
                                  ! N82 Model Space
                                  ! P - 1G7/2,2D5/2,2D3/2,3S1/2,1H11/2
N82      N82K     Q    K         ! Kruse-Wildenthal, BAPS 27, 533 (1982) p. 725.
                                  ! * A80PN Model Space
                                  ! P1f5/2,2p3/2,2p1/2,1g9/2 orbits
                                  ! N1f5/2,2p3/2,2p1/2,1g9/2 orbits
A80PN    A80PN    R    U         ! *
                                  ! ------------------------------------------
                                  ! * A80 Model Space
                                  ! 1f5/2,2p3/2,2p1/2,1g9/2 orbits
A80      A80      R    A         ! *
                                  ! ------------------------------------------
                                  ! * I13 Model Space
                                  ! 1i13/2 orbits
I13      I13      U    A         ! * Delta (-500.,.12)
```

```
! file LABELJ.DAT
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYABCDEFGHIJKLMNOPQRSTUVWXYZ!
0         1         2         3         4         5         6
01234567890123456789012345678901234567890123456789012345678901234567890
```

```
Directory SYS$THEORY:[BROWN.SHELLGROU]

BACKSH.COM;26                1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
BASIS.DIR;1                  7/10      30-SEP-1983 10:24  (RE,RWE,RE,RE)
CHANGEALL.COM;5              1/2       11-NOV-1986 14:47  (RE,RWED,RE,RE)
CLEANUP.COM;3                1/2       15-NOV-1986 17:51  (RE,RWED,RE,RE)
DELLIS.COM;5                 1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
DELLOG.COM;5                 1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
DELMAP.COM;5                 1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
DELOBJ.COM;5                 1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
DELOBJ.FPS;1                 1/4        9-JUN-1986 10:29  (RE,RWED,RE,RE)
EDTINI.EDT;6                 1/2        9-JUN-1982 10:58  (RE,RWED,RE,RE)
EXE.DIR;1                    3/6       14-APR-1984 12:38  (RWE,RWE,RWE,RWE)
IMG.DIR;1                    1/2       11-NOV-1986 10:44  (RWE,RWE,RE,RE)
INT.DIR;1                    1/2       12-NOV-1986 21:24  (RWE,RWE,RWE,RWE)
INTH.DIR;1                   2/4        6-NOV-1983 14:53  (RE,RWE,RE,RE)
ISOH.DIR;1                   3/4        6-NOV-1983 14:53  (RE,RWE,RE,RE)
ISOP.DIR;1                   1/2       14-NOV-1986 17:58  (RWE,RWE,RWE,RWE)
LANCZOS.DIR;1                5/10      30-SEP-1983 10:24  (RE,RWE,RE,RE)
LEV.DIR;1                    3/4        7-NOV-1983 18:45  (RE,RWE,RE,RE)
LFALL.COM;41                 9/10      16-NOV-1986 18:26  (RE,RWED,RE,RE)
LFALL.FPS;1                  8/8       10-JUN-1986 14:42  (RE,RWED,RE,RE)
LFALL.LOG;2                120/128     16-NOV-1986 23:52  (RWED,RWED,RE,)
LFBAB.COM;7                  3/4       16-NOV-1986 18:16  (RE,RWED,RE,RE)
LIBR.DIR;1                   1/2        9-NOV-1983 14:05  (RE,RWE,RE,RE)
LISTING.COM;4                1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
LISTING.DAT;1                0/32      17-NOV-1986 00:27  (RE,RWED,RE,RE)
LNKALL.COM;3                 8/8       12-NOV-1986 11:40  (RE,RWED,RE,RE)
LOGIN.COM;12                 1/2       16-NOV-1986 18:15  (RE,RWED,RE,RE)
LOGIN.FPS;1                  4/4       10-JUN-1986 14:05  (RE,RWED,RE,RE)
LOGIN.SHL;116               10/10      16-NOV-1986 18:15  (RE,RWED,RE,RE)
MAIN.DIR;1                  10/16      13-DEC-1984 09:23  (RWE,RWE,RWE,RWE)
MATRIX.DIR;1                 6/16      30-SEP-1983 10:25  (RE,RWE,RE,RE)
ME.DIR;1                     1/2       11-NOV-1986 10:51  (RWE,RWE,RE,RE)
MISC.DIR;1                   9/10      30-SEP-1983 10:25  (RE,RWE,RE,RE)
MOSH.DIR;1                   1/2        9-NOV-1983 15:23  (RE,RWE,RE,RE)
OBD.DIR;1                    1/2       24-NOV-1984 13:17  (RWE,RWE,RWE,RWE)
OP.DIR;1                     1/2       24-NOV-1984 10:31  (RWE,RWE,RWE,RWE)
OP2.DIR;1                    1/2       30-SEP-1983 10:25  (RE,RWE,RE,RE)
OP2FPS.DIR;1                 1/2       11-NOV-1986 10:51  (RWE,RWE,RE,RE)
OPFPS.DIR;1                  1/2       11-NOV-1986 10:51  (RWE,RWE,RE,RE)
PNCH.DIR;1                   3/4        6-NOV-1983 14:53  (RE,RWE,RE,RE)
PNCP.DIR;1                   1/2       14-NOV-1986 18:19  (RWE,RWE,RWE,RWE)
PRIOR.DIR;1                  3/16      30-SEP-1983 10:25  (RE,RWE,RE,RE)
PROJ.DIR;1                  10/10      15-DEC-1984 11:50  (RWE,RWE,RWE,RWE)
PUALL.COM;9                  1/4        9-JUN-1986 10:20  (RE,RWED,RE,RE)
RSH.DIR;1                    1/2       11-NOV-1986 10:56  (RWE,RWE,RWE,RWE)
SHELL.DIR;1                 10/16       5-MAR-1984 18:06  (RE,RWED,RE,RE)
SPS.DIR;1                   18/24      30-SEP-1983 10:25  (RE,RWE,RE,RE)
TRAMP.DIR;1                 10/10      30-SEP-1983 10:25  (RE,RWE,RE,RE)
TRANS.DIR;1                  6/10      30-SEP-1983 10:25  (RE,RWE,RE,RE)
TRD.DIR;1                    1/2       30-SEP-1983 10:25  (RE,RWE,RE,RE)
TXT.DIR;1                    4/4        5-MAR-1984 18:04  (RWE,RWE,RWE,RWE)

Total of 51 files, 300/446 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.BASIS]

BASCOM.FPS;2                 4/4       10-JUN-1986 12:04  (RE,RWED,RE,RE)
```

```
BASCOM.INC;54              4/4        3-JUN-1986 19:32  (RE,RWED,RE,RE)
BASIS.COM;23               1/2        9-JUN-1986 20:49  (RE,RWED,RE,RE)
BASIS.FOR;73              80/80      21-OCT-1986 10:51  (RE,RWED,RE,RE)
BASIS.FPS;5                1/4        9-JUN-1986 12:31  (RE,RWED,RE,RE)
BWRITE.FOR;6               2/2       12-NOV-1986 13:51  (RE,RWED,RE,RE)
CHNBAS.COM;2               1/4       10-JUN-1986 14:14  (RE,RWED,RE,RE)
CONVERT.COM;5              1/4        3-JUN-1986 16:09  (RE,RWED,RE,RE)
LFBAS.COM;3                1/2       17-OCT-1986 10:38  (RE,RWED,RE,RE)
LNKBAS.COM;18              1/2       29-MAY-1986 18:12  (RE,RWED,RE,RE)
LNKBAS.FPS;2               1/4        6-JUN-1986 17:32  (RE,RWED,RE,RE)
NEW.FPS;1                  1/4       16-NOV-1986 18:02  (RE,RWED,RE,RE)
PART.FOR;30                8/8       17-OCT-1986 10:37  (RE,RWED,RE,RE)
WOTAL.FOR;5               14/16       2-JUN-1986 14:01  (RE,RWED,RE,RE)
WOTHBW.FOR;13              7/8       10-JUN-1986 15:54  (RE,RWED,RE,RE)
WOTL.FOR;14                8/8       17-OCT-1986 10:37  (RE,RWED,RE,RE)
WOTPART.FOR;29            12/12      10-JUN-1986 15:54  (RE,RWED,RE,RE)
WOTPTS.FOR;22              9/10      17-OCT-1986 10:36  (RE,RWED,RE,RE)

Total of 18 files, 156/178 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.EXE]

BASIS.EXE;9               74/74      16-NOV-1986 23:58  (RE,RWED,RE,RE)
CHANGE.EXE;5              37/38      17-NOV-1986 00:16  (RE,RWED,RE,RE)
CMOSH.EXE;3               23/24      17-NOV-1986 00:21  (RE,RWED,RE,RE)
HCM.EXE;5                 24/24      17-NOV-1986 00:17  (RE,RWED,RE,RE)
ISOV.EXE;9                48/48      17-NOV-1986 00:23  (RE,RWED,RE,RE)
LANCZOS.EXE;19            74/74      17-NOV-1986 00:07  (RE,RWED,RE,RE)
LEVEL.EXE;5               32/32      17-NOV-1986 00:19  (RE,RWED,RE,RE)
MATRIX.EXE;8              60/60      17-NOV-1986 00:04  (RE,RWED,RE,RE)
MVEC.EXE;8                22/22      17-NOV-1986 00:08  (RE,RWED,RE,RE)
OPER.EXE;8                50/50      17-NOV-1986 00:05  (RE,RWED,RE,RE)
PN.EXE;5                  41/42      17-NOV-1986 00:18  (RE,RWED,RE,RE)
PNCME.EXE;4               46/46      17-NOV-1986 00:25  (RE,RWED,RE,RE)
PRCO.EXE;4                66/66      17-NOV-1986 00:22  (RE,RWED,RE,RE)
PRDV.EXE;4                56/56      17-NOV-1986 00:22  (RE,RWED,RE,RE)
PREDICT.EXE;9             58/58      16-NOV-1986 23:59  (RE,RWED,RE,RE)
PRG.EXE;3                 92/92      17-NOV-1986 00:24  (RE,RWED,RE,RE)
PRIORPROJ.EXE;3           81/82      17-NOV-1986 00:03  (RE,RWED,RE,RE)
PROJ.EXE;9                79/80      17-NOV-1986 00:01  (RE,RWED,RE,RE)
RTRD.EXE;7                47/48      17-NOV-1986 00:15  (RE,RWED,RE,RE)
SHELL.EXE;9              115/116     16-NOV-1986 23:56  (RE,RWED,RE,RE)
SU3INT.EXE;5              19/20      17-NOV-1986 00:18  (RE,RWED,RE,RE)
TBTDOP.EXE;8              30/30      17-NOV-1986 00:12  (RE,RWED,RE,RE)
TRAMP.EXE;8               99/100     17-NOV-1986 00:11  (RE,RWED,RE,RE)
TRANS.EXE;7               93/94      17-NOV-1986 00:14  (RE,RWED,RE,RE)

Total of 24 files, 1366/1376 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.INT]

INT.TXT;2                  1/2       12-NOV-1986 21:25  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.INTH]

LFPRDV.COM;2               1/2       19-NOV-1983 11:45  (RE,RWED,RE,RE)
LNKPRDV.COM;2              1/2       19-NOV-1983 11:45  (RE,RWED,RE,RE)
LNKPRJJ.COM;2              1/2        8-OCT-1984 13:23  (RE,RWED,RE,RE)
```

```
PRDV.FOR;81                12/12       23-JAN-1986 13:25   (RE,RWED,RE,RE)
PRJJ.FOR;17                10/10       29-NOV-1984 10:21   (RE,RWED,RE,RE)

Total of 5 files, 25/28 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.ISOH]

BSPR.ISO;2                  6/6         1-MAR-1984 15:01   (RE,RWED,RE,RE)
COFS.RAD;1                  1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
COG.FOR;12                 15/16       29-NOV-1984 10:21   (RE,RWED,RE,RE)
CONFS.RAD;1                 1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
CONST.RAD;3                 1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
CR.ISO;1                    6/6        14-DEC-1983 21:44   (RE,RWED,RE,RE)
F7BSIT.ISO;1                1/2         7-OCT-1984 16:32   (RE,RWED,RE,RE)
ISOV.DAO;1                  2/2        15-NOV-1986 16:54   (RE,RWED,RE,RE)
LFPRCO.COM;3                1/2        15-NOV-1985 10:44   (RE,RWED,RE,RE)
LNKPRCO.COM;5               1/2        12-SEP-1984 11:17   (RE,RWED,RE,RE)
MAG.RAD;1                   1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
MBKT.DAT;13                21/22       11-JAN-1985 17:03   (RE,RWED,RE,RE)
PIE.RAD;1                   1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
PIESR.RAD;1                 1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
PRCO.COM;4                  1/2         5-JUN-1983 16:02   (RE,RWED,RE,RE)
PRCO.DAO;2                  8/8        14-JAN-1985 13:22   (RE,RWED,RE,RE)
PRCO.FOR;21                 9/10       11-JAN-1985 16:21   (RE,RWED,RE,RE)
ROSR.RAD;1                  1/2         1-SEP-1982 15:07   (RE,RWED,RE,RE)
SDCO.ISO;2                  6/6        10-JUL-1983 13:37   (RE,RWED,RE,RE)
VMAG.F10;1                  6/6        10-MAR-1981 14:31   (RE,RWED,RE,RE)

Total of 20 files, 90/104 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.ISOP]

FINDC.FOR;2                 2/2        14-NOV-1986 18:00   (RE,RWED,RE,RE)
ISOV.EXE;2                 47/48       14-NOV-1986 18:16   (RE,RWED,RE,RE)
ISOV.FOR;82                38/38       15-NOV-1986 17:34   (RE,RWED,RE,RE)
LFISOV.COM;16               1/2        14-NOV-1986 18:16   (RE,RWED,RE,RE)
LNKISOV.COM;14              1/2        15-NOV-1986 17:34   (RE,RWED,RE,RE)
ORDER.FOR;5                 3/4        14-NOV-1986 18:15   (RE,RWED,RE,RE)

Total of 6 files, 92/96 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.LANCZOS]

CHNLAN.COM;1                1/4        10-JUN-1986 14:19   (RE,RWED,RE,RE)
CONVERT.COM;1               1/4         3-JUN-1986 16:15   (RE,RWED,RE,RE)
GNDLAN.FOR;17               4/4        10-JUN-1986 15:56   (RE,RWED,RE,RE)
HEADER.FPS;1                7/8        10-JUN-1986 12:05   (RE,RWED,RE,RE)
HEADER.INC;12               7/8        29-MAY-1986 18:18   (RE,RWED,RE,RE)
HEADLAN.FOR;50              3/4        13-NOV-1986 13:18   (RE,RWED,RE,RE)
LANC.FOR;204               38/38       10-JUN-1986 15:56   (RE,RWED,RE,RE)
LANCOM.FPS;1                2/4        10-JUN-1986 12:05   (RE,RWED,RE,RE)
LANCOM.INC;3                2/4         5-JUN-1986 21:36   (RE,RWED,RE,RE)
LANCZOS.COM;18              1/2        29-MAY-1986 18:18   (RE,RWED,RE,RE)
LANCZOS.FOR;142            32/32       14-NOV-1986 10:00   (RE,RWED,RE,RE)
LANCZOS.FPS;1               1/4         6-JUN-1986 17:34   (RE,RWED,RE,RE)
LNKLAN.COM;38               1/2        29-MAY-1986 18:18   (RE,RWED,RE,RE)
LNKLAN.FPS;1                1/4         9-JUN-1986 23:09   (RE,RWED,RE,RE)
NEW.FPS;1                   1/4        16-NOV-1986 18:02   (RE,RWED,RE,RE)
SPELAN.FOR;80               7/8        10-JUN-1986 15:56   (RE,RWED,RE,RE)
WCDUMP.FOR;2                6/6        10-JUN-1986 15:56   (RE,RWED,RE,RE)
```

Total of 17 files, 115/140 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.LEV]

| | | | |
|---|---|---|---|
| BADJTFM.FOR;2 | 3/4 | 29-NOV-1984 10:21 | (RE,RWED,RE,RE) |
| LABSCH.FOR;28 | 11/12 | 16-NOV-1986 18:24 | (RE,RWED,RE,RE) |
| LEVCOM.FOR;9 | 3/4 | 12-NOV-1986 13:57 | (RE,RWED,RE,RE) |
| LEVEL.COM;4 | 1/2 | 20-DEC-1984 13:38 | (RE,RWED,RE,RE) |
| LEVEL.FOR;51 | 25/26 | 12-NOV-1986 21:48 | (RE,RWED,RE,RE) |
| LNKLEV.COM;14 | 1/2 | 13-NOV-1986 08:55 | (RE,RWED,RE,RE) |

Total of 6 files, 44/50 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.LIBR]

| | | | |
|---|---|---|---|
| LFLIBR.COM;1 | 1/2 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| LIBRA.FOR;40 | 34/34 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| LIBRA.OBJ;14 | 27/28 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| LIBRB.FOR;138 | 39/40 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| LIBRB.OBJ;14 | 44/44 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| OPENBAB.FOR;17 | 4/4 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |
| OPENBAB.OBJ;19 | 4/4 | 17-NOV-1986 00:27 | (RE,RWED,RE,RE) |

Total of 7 files, 153/156 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MAIN]

| | | | |
|---|---|---|---|
| ALL.SIZ;785 | 8/8 | 17-NOV-1986 00:19 | (RE,RWED,RE,RE) |
| ALLS.SIZ;39 | 9/10 | 17-NOV-1986 00:25 | (RE,RWED,RE,RE) |
| BWRITE.FOR;1 | 2/2 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| CHNOPEN.COM;3 | 1/2 | 11-NOV-1986 14:53 | (RE,RWED,RE,RE) |
| DATA.INC;2 | 2/2 | 11-NOV-1986 17:15 | (RE,RWED,RE,RE) |
| EMULATE.FOR;1 | 1/2 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| FOR.DIR;1 | 1/2 | 11-NOV-1986 10:48 | (RWE,RWE,RE,RE) |
| FPS.DIR;1 | 4/4 | 11-NOV-1986 10:48 | (RWE,RWE,RE,RE) |
| FREEFM.FOR;2 | 22/22 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| INSERT.FOR;1 | 2/4 | 10-JUN-1986 12:10 | (RE,RWED,RE,RE) |
| MAC.DIR;1 | 3/4 | 11-NOV-1986 10:50 | (RWE,RWE,RE,RE) |
| MAKEALL.SIZ;5 | 1/2 | 16-NOV-1986 18:18 | (RE,RWED,RE,RE) |
| MAP.COM;20 | 1/2 | 12-NOV-1986 09:45 | (RE,RWED,RE,RE) |
| NAME.FOR;1 | 1/2 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENALL.OLB;60 | 90/100 | 16-NOV-1986 23:53 | (RE,RWED,RE,RE) |
| OPENBASIS.FOR;1 | 2/4 | 9-JUN-1986 10:52 | (RE,RWED,RE,RE) |
| OPENFILE.FOR;1 | 5/6 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENFOR.COM;1 | 1/4 | 30-MAY-1986 14:32 | (RE,RWED,RE,RE) |
| OPENINT.FOR;2 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENLAB.FOR;3 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENLIB.COM;3 | 1/4 | 29-MAY-1986 18:08 | (RE,RWED,RE,RE) |
| OPENOP.FOR;2 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENOP2.FOR;2 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| OPENSPS.FOR;3 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| ORDER.FOR;5 | 3/4 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| PARAM.HLP;3 | 10/10 | 16-NOV-1986 18:19 | (RE,RWED,RE,RE) |
| PARAM.INC;1 | 2/4 | 5-JUN-1986 15:48 | (RE,RWED,RE,RE) |
| PUT.COM;2 | 1/4 | 29-MAY-1986 18:08 | (RE,RWED,RE,RE) |
| SKIPHEAD.FOR;2 | 1/2 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| TIMER.FOR;1 | 1/2 | 10-JUN-1986 15:54 | (RE,RWED,RE,RE) |
| TITLE.FOR;18 | 5/6 | 11-NOV-1986 15:28 | (RE,RWED,RE,RE) |

```
Total of 31 files, 195/238 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MAIN.FOR]

CHNINT.COM;1                1/4        10-JUN-1986 14:13   (RE,RWED,RE,RE)
INTCOMP.COM;1               1/2        30-MAY-1986 17:42   (RE,RWED,RE,RE)
INTLIB.COM;1                1/2        30-MAY-1986 17:42   (RE,RWED,RE,RE)
LANC.INC;1                  1/2         9-JUN-1986 21:04   (RE,RWED,RE,RE)
MACALL.OLB;1               76/100      12-JUN-1986 23:52   (RE,RWED,RE,RE)
MACFOR.INC;1                1/4         9-JUN-1986 10:44   (RE,RWED,RE,RE)
XARITH.FOR;1                2/2        10-JUN-1986 15:54   (RE,RWED,RE,RE)
XLOGIC.FOR;1                2/2        10-JUN-1986 16:49   (RE,RWED,RE,RE)
XSHIFT.FOR;1                3/4        10-JUN-1986 16:51   (RE,RWED,RE,RE)
XTEST.FOR;1                12/12       10-JUN-1986 16:49   (RE,RWED,RE,RE)
XTESTBAT.FOR;1              1/2        10-JUN-1986 15:54   (RE,RWED,RE,RE)
XWHERE.FOR;1                1/2        10-JUN-1986 16:50   (RE,RWED,RE,RE)

Total of 12 files, 102/138 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MAIN.FPS]

APZAP.FOR;1                 1/4         3-JUN-1986 16:55   (RE,RWED,RE,RE)
BWRITE.FOR;1                2/4        10-JUN-1986 14:24   (RE,RWED,RE,RE)
CHNOPEN.COM;1               1/4        10-JUN-1986 14:12   (RE,RWED,RE,RE)
CONVERT.FOR;1              13/14       11-NOV-1986 11:58   (RE,RWED,RE,RE)
CONVERT.TRA;2              13/14       16-NOV-1986 18:12   (RE,RWED,RE,RE)
DATA.FPS;1                  2/4        10-JUN-1986 14:09   (RE,RWED,RE,RE)
DATA.INC;1                  2/4         9-JUN-1986 10:12   (RE,RWED,RE,RE)
DATE.FOR;1                  1/4        10-JUN-1986 14:27   (RE,RWED,RE,RE)
EMULATE.FOR;1               1/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
FREEFM.FOR;1               22/24       10-JUN-1986 14:25   (RE,RWED,RE,RE)
FUNCTION.FOR;1              1/4        10-JUN-1986 14:27   (RE,RWED,RE,RE)
INCLUDE.FOR;1               2/2        11-NOV-1986 11:58   (RE,RWED,RE,RE)
INTCOMP.FPS;1               1/4         6-JUN-1986 17:40   (RE,RWED,RE,RE)
INTLIB.FPS;1                1/4         6-JUN-1986 19:41   (RE,RWED,RE,RE)
LANC.FPS;1                  1/4        10-JUN-1986 12:05   (RE,RWED,RE,RE)
MACFOR.FPS;1                1/4        10-JUN-1986 12:05   (RE,RWED,RE,RE)
MACFOR.INC;1                1/4         3-JUN-1986 16:40   (RE,RWED,RE,RE)
MAINCOM.FPS;1              12/12       10-JUN-1986 12:05   (RE,RWED,RE,RE)
MAINCOM.INC;1              12/12        3-JUN-1986 16:20   (RE,RWED,RE,RE)
NAME.FOR;1                  1/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
NEW.FPS;1                   1/4         6-JUN-1986 17:38   (RE,RWED,RE,RE)
OPENFILE.FOR;1              6/8        10-JUN-1986 14:25   (RE,RWED,RE,RE)
OPENFOR.FPS;1               1/4         9-JUN-1986 10:47   (RE,RWED,RE,RE)
OPENINT.FOR;1               2/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
OPENLAB.FOR;1               3/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
OPENLIB.FPS;1               1/4         9-JUN-1986 10:47   (RE,RWED,RE,RE)
OPENOP.FOR;1                2/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
OPENOP2.FOR;1               2/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
OPENSPS.FOR;1               3/4        10-JUN-1986 14:25   (RE,RWED,RE,RE)
ORDER.FOR;1                 3/4        10-JUN-1986 14:26   (RE,RWED,RE,RE)
OXFOR.FPS;1                 2/4         6-JUN-1986 17:38   (RE,RWED,RE,RE)
OXLIBR.FPS;1                1/4         6-JUN-1986 17:38   (RE,RWED,RE,RE)
PARAM.FPS;1                 3/4        10-JUN-1986 12:05   (RE,RWED,RE,RE)
PARAM.INC;1                 3/4         5-JUN-1986 21:19   (RE,RWED,RE,RE)
SKIPHEAD.FOR;1              1/4        10-JUN-1986 14:26   (RE,RWED,RE,RE)
TIME.FOR;1                  4/4         3-JUN-1986 16:56   (RE,RWED,RE,RE)
TIMER.FOR;1                 5/8        10-JUN-1986 14:27   (RE,RWED,RE,RE)
TIMETEST.FOR;1              1/4         3-JUN-1986 16:56   (RE,RWED,RE,RE)
TITLE.FOR;1                 5/8        10-JUN-1986 14:27   (RE,RWED,RE,RE)
```

```
WHEREBIT.FOR;1              11/12       3-JUN-1986 17:02   (RE,RWED,RE,RE)
XARITH.FOR;1                 2/4       10-JUN-1986 14:27   (RE,RWED,RE,RE)
XLOGIC.FOR;1                 2/4       10-JUN-1986 14:27   (RE,RWED,RE,RE)
XMAC.FOR;1                   1/4        3-JUN-1986 16:56   (RE,RWED,RE,RE)
XSHIFT.FOR;1                 3/4       10-JUN-1986 14:27   (RE,RWED,RE,RE)
XTEST.FOR;1                 12/12      10-JUN-1986 14:27   (RE,RWED,RE,RE)
XTESTBAT.FOR;1               1/4       10-JUN-1986 14:27   (RE,RWED,RE,RE)
XWHERE.FOR;1                 1/4        3-JUN-1986 17:36   (RE,RWED,RE,RE)

Total of 47 files, 173/270 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MAIN.MAC]

INTCOMP.COM;1                1/2       30-MAY-1986 17:52   (RE,RWED,RE,RE)
INTLIB.COM;1                 1/2       30-MAY-1986 17:43   (RE,RWED,RE,RE)
LANC.INC;2                   1/2       15-NOV-1986 16:21   (RE,RWED,RE,RE)
MACALL.OLB;12               59/100     16-NOV-1986 23:53   (RE,RWED,RE,RE)
MACFOR.INC;1                 1/4        9-JUN-1986 10:45   (RE,RWED,RE,RE)
XARITH.MAR;1                11/12      29-MAY-1986 18:10   (RE,RWED,RE,RE)
XLOGIC.MAR;1                 8/8       29-MAY-1986 18:10   (RE,RWED,RE,RE)
XSHIFT.MAR;1                11/12      29-MAY-1986 18:10   (RE,RWED,RE,RE)
XTEST.MAR;1                 18/18      29-MAY-1986 18:10   (RE,RWED,RE,RE)
XTESTBAT.MAR;1               3/4       29-MAY-1986 18:10   (RE,RWED,RE,RE)
XWHERE.FOR;1                 1/4       30-MAY-1986 17:50   (RE,RWED,RE,RE)

Total of 11 files, 115/168 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MATRIX]

CHNMAT.COM;1                 1/4       10-JUN-1986 14:19   (RE,RWED,RE,RE)
CHNOPR.COM;1                 1/4       10-JUN-1986 14:18   (RE,RWED,RE,RE)
CLEBG.FOR;15                 9/10      10-JUN-1986 15:55   (RE,RWED,RE,RE)
FACTOR.COM;2                 1/2       12-MAR-1985 15:58   (RE,RWED,RE,RE)
FACTOR.FOR;3                 4/4       29-MAY-1986 18:17   (RE,RWED,RE,RE)
FINDMAT.FOR;4                2/2       10-JUN-1986 15:55   (RE,RWED,RE,RE)
HOPER.FOR;4                 11/12      10-JUN-1986 15:55   (RE,RWED,RE,RE)
LNKFACTOR.COM;3              1/2       29-MAY-1986 18:17   (RE,RWED,RE,RE)
LNKMAT.COM;4                 1/2       29-MAY-1986 18:17   (RE,RWED,RE,RE)
LNKMAT.FPS;1                 1/4        6-JUN-1986 17:31   (RE,RWED,RE,RE)
LNKOPR.COM;4                 1/2       29-MAY-1986 18:17   (RE,RWED,RE,RE)
LNKOPR.FPS;1                 1/4        6-JUN-1986 19:50   (RE,RWED,RE,RE)
MAT.FOR;13                  26/26      10-JUN-1986 15:55   (RE,RWED,RE,RE)
MATCOM.FPS;1                 6/8       10-JUN-1986 12:05   (RE,RWED,RE,RE)
MATCOM.INC;4                 6/6       12-JUN-1986 16:17   (RE,RWED,RE,RE)
MATRIX.COM;7                 1/2       30-MAY-1986 10:42   (RE,RWED,RE,RE)
MATRIX.FOR;13               23/24      10-JUN-1986 15:55   (RE,RWED,RE,RE)
MATRIX.FPS;1                 1/4        6-JUN-1986 17:32   (RE,RWED,RE,RE)
NEUTPROT.FOR;4               5/8        3-JUN-1986 17:09   (RE,RWED,RE,RE)
OPER.COM;8                   1/2       30-MAY-1986 10:42   (RE,RWED,RE,RE)
OPER.FOR;22                 69/70      10-JUN-1986 15:55   (RE,RWED,RE,RE)
OPER.FPS;1                   1/4        9-JUN-1986 17:42   (RE,RWED,RE,RE)
OPRCOM.FPS;1                 1/4       10-JUN-1986 12:05   (RE,RWED,RE,RE)
OPRCOM.INC;3                 1/2       29-MAY-1986 18:17   (RE,RWED,RE,RE)
ORTHOG.FOR;5                10/10      10-JUN-1986 15:55   (RE,RWED,RE,RE)
ORTOUT.FOR;7                 3/4       10-JUN-1986 15:55   (RE,RWED,RE,RE)

Total of 26 files, 188/226 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.ME]
```

```
SHELLME.COM;1                1/2        6-JUN-1986 12:12  (RE,RWED,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MISC]

BDATA.FOR;1                  2/2       10-FEB-1986 09:31  (RE,RWED,RE,RE)
CHANGE.COM;18                1/2       16-NOV-1985 16:15  (RE,RWED,RE,RE)
CHANGE.FOR;62               30/30      10-FEB-1986 09:31  (RE,RWED,RE,RE)
CHANGEF.FOR;3               30/30      10-FEB-1986 09:31  (RE,RWED,RE,RE)
CHNCOM.INC;3                 1/2       12-NOV-1986 13:50  (RE,RWED,RE,RE)
CONV.COM;4                   1/2        5-NOV-1984 11:05  (RE,RWED,RE,RE)
CONV.FOR;20                  4/4       10-FEB-1986 09:31  (RE,RWED,RE,RE)
CONVISO.FOR;3                8/8       12-NOV-1986 13:54  (RE,RWED,RE,RE)
CORE.FOR;27                 24/24      10-FEB-1986 09:32  (RE,RWED,RE,RE)
FINDPCH.FOR;15               2/2       10-FEB-1986 09:32  (RE,RWED,RE,RE)
HCM.COM;14                   1/2       30-SEP-1983 11:53  (RE,RWED,RE,RE)
HCM.FOR;28                  17/18      10-FEB-1986 09:32  (RE,RWED,RE,RE)
HCMCOM.INC;3                 1/2       12-NOV-1986 13:58  (RE,RWED,RE,RE)
INPUT.DAT;2                  1/2       10-JAN-1986 12:06  (RE,RWED,RE,RE)
INTER.COM;2                  1/2        9-NOV-1983 13:15  (RE,RWED,RE,RE)
INTER.FOR;18                 5/6       10-FEB-1986 09:32  (RE,RWED,RE,RE)
INTERCOM.INC;3               1/2       12-NOV-1986 13:50  (RE,RWED,RE,RE)
ISOCOMP.FOR;3               14/14      12-NOV-1986 13:55  (RE,RWED,RE,RE)
LNKCHN.COM;23                1/2       13-NOV-1986 08:44  (RE,RWED,RE,RE)
LNKCHNF.COM;2                1/2        2-JAN-1986 21:33  (RE,RWED,RE,RE)
LNKCORE.COM;3                1/2       19-FEB-1985 10:21  (RE,RWED,RE,RE)
LNKHCM.COM;18                1/2       11-NOV-1986 17:10  (RE,RWED,RE,RE)
LNKINTER.COM;1               1/2        9-NOV-1983 13:17  (RE,RWED,RE,RE)
LNKMIT.COM;1                 1/2       12-FEB-1984 12:27  (RE,RWED,RE,RE)
LNKPN.COM;5                  1/2       11-NOV-1986 17:10  (RE,RWED,RE,RE)
LNKREADIN.COM;5              1/2       19-FEB-1985 10:22  (RE,RWED,RE,RE)
MIT.FOR;43                  10/10      10-FEB-1986 09:32  (RE,RWED,RE,RE)
NINEJ.FOR;14                17/18      10-FEB-1986 09:32  (RE,RWED,RE,RE)
OUTPUT.DAT;2                 1/2       10-JAN-1986 12:06  (RE,RWED,RE,RE)
PN.COM;2                     1/2       15-NOV-1984 15:42  (RE,RWED,RE,RE)
PN.FOR;61                   17/18      12-NOV-1986 13:53  (RE,RWED,RE,RE)
PN.OLD;1                    13/14      14-AUG-1985 11:16  (RE,RWED,RE,RE)
PN.WEO;1                    17/18      12-NOV-1986 11:34  (RE,RWED,RE,RE)
RATIO.FOR;3                  2/2       10-FEB-1986 09:32  (RE,RWED,RE,RE)
READIN.FOR;24               15/16      12-NOV-1986 09:49  (RE,RWED,RE,RE)
SPSCH.FOR;9                  8/8       10-FEB-1986 09:32  (RE,RWED,RE,RE)
SU3INT.COM;2                 1/2       10-FEB-1986 09:38  (RE,RWED,RE,RE)
SU3INT.FOR;21               33/34      10-FEB-1986 09:32  (RE,RWED,RE,RE)
TESTBATCH.FOR;1              1/2       10-FEB-1986 09:32  (RE,RWED,RE,RE)

Total of 39 files, 288/316 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.MOSH]

CMOSH.FOR;15                39/40      29-NOV-1984 10:21  (RE,RWED,RE,RE)
LFCMOSH.COM;1                1/2        9-NOV-1983 16:23  (RE,RWED,RE,RE)
LFRMOCH.COM;2                1/2       11-NOV-1983 16:19  (RE,RWED,RE,RE)
MBKT.DAT;13                 21/22      11-JAN-1985 17:03  (RE,RWED,RE,RE)
MO.DAT;1                     5/6        9-NOV-1983 15:31  (RE,RWED,RE,RE)
MO.FOR;1                     8/8       29-NOV-1984 10:21  (RE,RWED,RE,RE)
MOSH.DAT;9                  13/14      11-JAN-1985 17:03  (RE,RWED,RE,RE)
RMOSH.FOR;9                 10/10      29-NOV-1984 10:21  (RE,RWED,RE,RE)
RUNCMOSH.COM;16              1/2       11-JAN-1985 17:03  (RE,RWED,RE,RE)
```

```
Total of 9 files, 99/106 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.OBD]

OBD.TXT;2                    1/2       12-NOV-1986 21:50  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.OP]

OP.TXT;2                     1/2       12-NOV-1986 21:25  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.OP2]

OP2.TXT;1                    1/2       12-NOV-1986 21:25  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.PNCH]

ADDPNC.FOR;3                 2/2       29-NOV-1984 10:21  (RE,RWED,RE,RE)
LFPRG.COM;5                  1/2       15-NOV-1985 10:44  (RE,RWED,RE,RE)
LNKPRG.COM;3                 1/2       15-NOV-1985 10:44  (RE,RWED,RE,RE)
MBKT.DAT;3                   5/6        9-NOV-1983 16:15  (RE,RWED,RE,RE)
PRG.FOR;4                   21/22       5-FEB-1985 12:38  (RE,RWED,RE,RE)
PVG.FOR;2                   29/30      11-JAN-1985 12:56  (RE,RWED,RE,RE)
PVSZ.INT;1                   3/4       16-APR-1982 12:33  (RE,RWED,RE,RE)
PVSZ.PNC;1                   3/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVSZ1.PNC;1                  3/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVSZ2.PNC;1                  3/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVSZ3.PNC;1                  3/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVSZ4.PNC;1                  3/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVTZ.PNC;1                   1/2       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVTZ1.PNC;2                  1/2       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVTZ2.PNC;3                  1/2       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ.PNC;1                   4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ1.PNC;1                  4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ2.PNC;1                  4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ3.PNC;1                  4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ4.PNC;1                  4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
PVVZ5.PNC;1                  4/4       15-NOV-1986 17:57  (RE,RWED,RE,RE)
RAD.FOR;2                   12/12      20-JAN-1985 14:24  (RE,RWED,RE,RE)
RUNPRG.COM;2                 2/2        9-NOV-1983 15:09  (RE,RWED,RE,RE)

Total of 23 files, 118/132 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.PNCP]

FINDPC.FOR;1                 4/4       14-NOV-1986 18:19  (RE,RWED,RE,RE)
FINDPCCOM.INC;1              9/10      14-NOV-1986 18:19  (RE,RWED,RE,RE)
LFPNCME.COM;7                1/2       14-NOV-1986 18:19  (RE,RWED,RE,RE)
LNKPNCME.COM;16              1/2       14-NOV-1986 18:19  (RE,RWED,RE,RE)
ORDER.FOR;5                  3/4       14-NOV-1986 18:19  (RE,RWED,RE,RE)
PNCME.EXE;2                 46/46      14-NOV-1986 18:19  (RE,RWED,RE,RE)
PNCME.FOR;22                33/34      14-NOV-1986 18:19  (RE,RWED,RE,RE)

Total of 7 files, 97/102 blocks.
```

```
Directory SYS$THEORY:[BROWN.SHELLGROU.PRIOR]

FEDUP.FOR;20                11/12       12-NOV-1986 22:21  (RE,RWED,RE,RE)
JMJP.FOR;16                 18/18       12-NOV-1986 22:25  (RE,RWED,RE,RE)
LNKPRJ.COM;13                1/2        12-NOV-1986 22:35  (RE,RWED,RE,RE)
PRIORPROJ.COM;7              1/2        16-DEC-1985 14:55  (RE,RWED,RE,RE)
PRIORPROJ.FOR;55            60/60       12-NOV-1986 22:23  (RE,RWED,RE,RE)
PRJCOM.INC;9                 8/8        12-NOV-1986 22:14  (RE,RWED,RE,RE)
PRJSBS.FOR;20               15/16       12-NOV-1986 22:32  (RE,RWED,RE,RE)

Total of 7 files, 114/118 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.PROJ]

CHNPRD.COM;1                 1/4        10-JUN-1986 14:16  (RE,RWED,RE,RE)
CHNPRJ.COM;1                 1/4        10-JUN-1986 14:17  (RE,RWED,RE,RE)
FACSET.FOR;17                6/6        10-JUN-1986 15:55  (RE,RWED,RE,RE)
INFWRITE.FOR;1               8/8        29-MAY-1986 18:16  (RE,RWED,RE,RE)
INVT.FOR;3                   3/4        10-JUN-1986 15:55  (RE,RWED,RE,RE)
JMJPPRED.FOR;16             13/14       10-JUN-1986 15:55  (RE,RWED,RE,RE)
JPLUS.FOR;3                 23/24       10-JUN-1986 15:55  (RE,RWED,RE,RE)
LNKPRD.COM;29                1/2        29-MAY-1986 18:16  (RE,RWED,RE,RE)
LNKPRD.FPS;1                 1/4         6-JUN-1986 17:32  (RE,RWED,RE,RE)
LNKPRDWR.COM;1               1/4         3-JUN-1986 16:13  (RE,RWED,RE,RE)
LNKPRJ.COM;8                 1/2        11-NOV-1986 15:25  (RE,RWED,RE,RE)
LNKPRJ.FPS;1                 1/4         6-JUN-1986 17:33  (RE,RWED,RE,RE)
NEW.FPS;1                    1/4        16-NOV-1986 18:03  (RE,RWED,RE,RE)
ORDERP.FOR;3                 6/6        10-JUN-1986 15:55  (RE,RWED,RE,RE)
PATPARPRD.FOR;16             2/2        10-JUN-1986 15:55  (RE,RWED,RE,RE)
PATT.FOR;3                   2/2        10-JUN-1986 15:55  (RE,RWED,RE,RE)
PRDCOM.FPS;1                 5/8        10-JUN-1986 12:05  (RE,RWED,RE,RE)
PRDCOM.INC;4                 5/8         3-JUN-1986 21:48  (RE,RWED,RE,RE)
PREDICT.COM;25               1/2        29-MAY-1986 18:16  (RE,RWED,RE,RE)
PREDICT.FOR;32              31/32       10-JUN-1986 15:55  (RE,RWED,RE,RE)
PREDICT.FPS;1                1/4         9-JUN-1986 17:36  (RE,RWED,RE,RE)
PRJCOM.FPS;1                 7/8        10-JUN-1986 12:05  (RE,RWED,RE,RE)
PRJCOM.INC;2                 7/8        12-JUN-1986 16:17  (RE,RWED,RE,RE)
PROJ.COM;10                  1/2        29-MAY-1986 18:16  (RE,RWED,RE,RE)
PROJ.FOR;12                 57/58       10-OCT-1986 18:03  (RE,RWED,RE,RE)
PROJ.FPS;1                   1/4         6-JUN-1986 17:33  (RE,RWED,RE,RE)
READBAS.FOR;2               12/12       10-JUN-1986 15:55  (RE,RWED,RE,RE)
READBASWR.COM;1              1/4         3-JUN-1986 16:13  (RE,RWED,RE,RE)
SBS.FOR;5                   20/20       10-JUN-1986 15:55  (RE,RWED,RE,RE)
SOLVJT.FOR;25                9/10       10-JUN-1986 15:55  (RE,RWED,RE,RE)
TEMP.FPS;1                   1/4        10-JUN-1986 12:05  (RE,RWED,RE,RE)
TEMP.INC;1                   1/4         3-JUN-1986 19:58  (RE,RWED,RE,RE)
TMTP.FOR;3                  13/14       10-JUN-1986 15:55  (RE,RWED,RE,RE)
TMTPPRED.FOR;16              8/8        10-JUN-1986 15:55  (RE,RWED,RE,RE)

Total of 34 files, 252/304 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.RSH]

RSH.TXT;2                    1/2        12-NOV-1986 21:23  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.SHELL]

BADJT.FOR;18                 5/6        12-JUN-1986 15:44  (RE,RWED,RE,RE)
```

```
CONVSHL.COM;1                1/2     12-JUN-1986 15:12  (RE,RWED,RE,RE)
DEL.FOR;18                   2/2     12-JUN-1986 15:44  (RE,RWED,RE,RE)
GROUSHL.FOR;25               3/4     12-JUN-1986 15:44  (RE,RWED,RE,RE)
LABEL.FOR;72                16/16    16-NOV-1986 18:22  (RE,RWED,RE,RE)
LNKSHL.COM;31                1/2     11-NOV-1986 14:53  (RE,RWED,RE,RE)
RBAS.FOR;116                43/44    12-JUN-1986 15:44  (RE,RWED,RE,RE)
RTRAMP.FOR;143              56/56    12-JUN-1986 15:44  (RE,RWED,RE,RE)
SHELL.COM;26                 1/2     11-NOV-1986 14:40  (RE,RWED,RE,RE)
SHELL.FOR;158               52/52    12-JUN-1986 15:43  (RE,RWED,RE,RE)
SHELLOLD.FOR;1              39/40     3-MAR-1986 15:30  (RWED,RWED,RE,)
SHLCOM.INC;4                 4/4      5-JUN-1986 16:05  (RE,RWED,RE,RE)
SHSP.FOR;39                 21/22    19-OCT-1986 17:15  (RE,RWED,RE,RE)
SHSP2.FOR;2                 10/10    10-AUG-1985 16:56  (RE,RWED,RE,RE)
SPECH.FOR;25                 3/4     12-JUN-1986 15:44  (RE,RWED,RE,RE)

Total of 15 files, 257/266 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.SPS]

A80.INT;3                    9/10    29-MAY-1986 19:02  (RE,RWED,RE,RE)
A80.SPS;3                    3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
A80PN.INT;3                 37/38    29-MAY-1986 19:02  (RE,RWED,RE,RE)
A80PN.SPS;3                  4/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
BKBFP.INT;3                 14/14    29-MAY-1986 19:02  (RE,RWED,RE,RE)
BKUOSD.INT;2                 5/6     29-MAY-1986 19:02  (RE,RWED,RE,RE)
BRSD.INT;2                   5/6     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CKI.INT;3                    2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CKII.INT;3                   2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CKIIM.INT;7                  2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CKPOT.INT;11                 2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CKPOTPN.INT;3                4/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
CW.INT;11                    6/6     29-MAY-1986 19:02  (RE,RWED,RE,RE)
D3F7.SPS;12                  2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
D3F7CONP.INT;4               2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
D3F7COPN.INT;5               2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
D3F7PN.SPS;6                 2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F5P.SPS;6                    2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F5PV.INT;5                   6/6     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F7.SPS;6                     1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F742.INT;3                   1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F742PN.INT;8                 2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F748.INT;4                   1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F748BPN.INT;4                2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F748PN.INT;8                 2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F748S.INT;3                  1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F754.INT;3                   1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F7MBZ.INT;4                  1/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
F7P3.SPS;1                   2/2     10-JUN-1986 16:38  (RE,RWED,RE,RE)
F7P3COSNP.INT;1              2/2     10-JUN-1986 16:21  (RE,RWED,RE,RE)
F7P3COSPN.INT;1              2/2     10-JUN-1986 16:20  (RE,RWED,RE,RE)
F7P3COSRC.INT;1              2/2      4-JUN-1986 10:44  (RE,RWED,RE,RE)
F7P3PN.SPS;1                 2/2      4-JUN-1986 10:29  (RE,RWED,RE,RE)
F7PN.SPS;16                  2/2     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FEP.INT;4                    3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FEPP.INT;3                   3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FEPPZ.INT;4                  3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FEPQ.INT;3                   3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FP.SPS;6                     3/4     29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPCONP.INT;4                 8/8     13-NOV-1986 12:02  (RE,RWED,RE,RE)
FPCOPN.INT;4                 8/8     13-NOV-1986 12:02  (RE,RWED,RE,RE)
```

```
FPE.INT;8                15/16      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPEO.INT;3               15/16      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPM.INT;9                13/14      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPN.INT;1                 3/4       12-JUN-1986 10:12  (RE,RWED,RE,RE)
FPN.SPS;6                 2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPNITEN.INT;1             5/6       12-JUN-1986 10:13  (RE,RWED,RE,RE)
FPNPN.INT;1               7/8       12-JUN-1986 10:13  (RE,RWED,RE,RE)
FPPN.SPS;5                3/4       29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPUSD.INT;3              14/14      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPV.INT;7                16/16      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPV2.INT;3               16/16      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPVH.INT;5               17/18      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPVN.INT;4                8/8       29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPVPN.INT;3              56/56      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPVTEN.INT;3             21/22      29-MAY-1986 19:02  (RE,RWED,RE,RE)
FPY.INT;6                16/16      29-MAY-1986 19:02  (RE,RWED,RE,RE)
GL.INT;12                 5/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
GL.SPS;10                 2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
GLB.INT;4                 5/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HASN.INT;4               10/10      29-MAY-1986 19:02  (RE,RWED,RE,RE)
HASP.INT;5               10/10      29-MAY-1986 19:02  (RE,RWED,RE,RE)
HASP.SPS;6                2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HASPPN.INT;2             29/30      29-MAY-1986 19:02  (RE,RWED,RE,RE)
HASPPN.SPS;5              3/4       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HBGF.INT;6                5/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HBUFP.INT;2              14/14      16-OCT-1986 13:14  (RE,RWED,RE,RE)
HBUSD.INT;2               5/6       16-OCT-1986 13:14  (RE,RWED,RE,RE)
HO.INT;3                  5/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HO.SPS;5                  1/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HW.INT;12                 3/4       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HWOLD.INT;2               3/4       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HWPN.INT;2                6/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
HWTEN.INT;2               2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
I13.INT;3                 2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
I13.SPS;3                 2/2       29-MAY-1986 19:02  (RE,RWED,RE,RE)
KBFP.INT;2               14/14      29-MAY-1986 19:02  (RE,RWED,RE,RE)
KUOSD.INT;4               6/6       29-MAY-1986 19:02  (RE,RWED,RE,RE)
LABEL.DAT;206            37/38      16-NOV-1986 18:21  (RE,RWED,RE,RE)
LABELJ.DAT;13             1/2       16-NOV-1986 18:22  (RE,RWED,RE,RE)
MP.INT;3                  2/2       29-MAY-1986 19:03  (RE,RWED,RE,RE)
N82.SPS;7                 3/4       29-MAY-1986 19:03  (RE,RWED,RE,RE)
N82K.INT;2               11/12      29-MAY-1986 19:03  (RE,RWED,RE,RE)
P.SPS;12                  1/2       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PFG.SPS;7                 5/6       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PFGPN.SPS;11              6/6       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PFV.INT;9                11/12      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PFVN.INT;5               18/18      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PN.INT;3                 17/18      10-JUN-1986 16:36  (RE,RWED,RE,RE)
PN.SPS;1                  1/2       10-JUN-1986 16:36  (RE,RWED,RE,RE)
PPN.SPS;6                 1/2       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSD.SPS;6                 3/4       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDCM.INT;4              19/20      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDCMPN.INT;2            59/60      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDMK.INT;3              19/20      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDMWK.INT;7             17/18      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDMWKPN.INT;8           59/60      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDPN.SPS;11              3/4       29-MAY-1986 19:03  (RE,RWED,RE,RE)
PSDZ.INT;5               18/18      29-MAY-1986 19:03  (RE,RWED,RE,RE)
PW.INT;11                 6/6       29-MAY-1986 19:03  (RE,RWED,RE,RE)
REWIL.INT;6               3/4       29-MAY-1986 19:03  (RE,RWED,RE,RE)
```

```
REWILE.INT;4              19/20    29-MAY-1986 19:03   (RE,RWED,RE,RE)
REWILPN.INT;5              8/8     29-MAY-1986 19:03   (RE,RWED,RE,RE)
RN.SPS;3                   2/2     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SAS.INT;3                  3/4     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SASPN.INT;4                6/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SD.SPS;16                  2/2      4-JUN-1986 10:28   (RE,RWED,RE,RE)
SDCONP.INT;5              11/12    13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOPN.INT;7              11/12    13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOSRC.INT;2              3/4     13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOSRCNP.INT;3            3/4     13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOSRCPN.INT;3            3/4     13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOTNP.INT;9             11/12    13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDCOTPN.INT;7             11/12    13-NOV-1986 12:02   (RE,RWED,RE,RE)
SDM.INT;2                  5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDMPN.INT;2               17/18    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDMTEN.INT;2               6/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPF.SPS;7                 4/4     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFCM.INT;4              19/20    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFM40.INT;3             37/38    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFMK.INT;6              37/38    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFMW.INT;5              37/38    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFVH.INT;2              15/16    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFVHB.INT;5             15/16    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFVHW.INT;2             15/16    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFVP.INT;3              33/34    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFVPNP.INT;2            33/34    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPFW.INT;2                6/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDPN.SPS;27                2/2     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SDRHO.INT;4               11/12    13-NOV-1986 12:03   (RE,RWED,RE,RE)
SDW.SPS;3                  2/2     16-OCT-1986 13:14   (RE,RWED,RE,RE)
SG16F.INT;7              163/164   13-NOV-1986 12:04   (RE,RWED,RE,RE)
SG40F.INT;2              163/164   13-NOV-1986 12:04   (RE,RWED,RE,RE)
SLG.SPS;8                  2/2     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGM.INT;3                 5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGML.INT;3                5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGMT0.INT;4               5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGO.INT;5                 5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGT.INT;4                 2/2     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SLGT.SPS;5                 2/2     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSD.SPS;16                3/4     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDCM.INT;5              24/24    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDMK.INT;3              24/24    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDMKT.INT;3             24/24    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDMWK.INT;14            25/26    29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDPF.SPS;6               5/6     29-MAY-1986 19:03   (RE,RWED,RE,RE)
SPSDPFCM.INT;4            42/42    29-MAY-1986 19:04   (RE,RWED,RE,RE)
STE.INT;2                  3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
STF.INT;2                  3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
SU3P.INT;3                 2/2     29-MAY-1986 19:04   (RE,RWED,RE,RE)
SU3PSD.INT;5              19/20    29-MAY-1986 19:04   (RE,RWED,RE,RE)
SU3SD.INT;3                6/6     29-MAY-1986 19:04   (RE,RWED,RE,RE)
SU3Z.INT;4                 3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
SU3ZBM.INT;3               3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
VP.INT;6                   7/8     29-MAY-1986 19:04   (RE,RWED,RE,RE)
VPNP.INT;2                 7/8     29-MAY-1986 19:04   (RE,RWED,RE,RE)
VPTH.INT;2                 7/8     29-MAY-1986 19:04   (RE,RWED,RE,RE)
W.INT;11                   7/8      4-JUN-1986 10:29   (RE,RWED,RE,RE)
W0.INT;4                   3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
W4.INT;3                   3/4     29-MAY-1986 19:04   (RE,RWED,RE,RE)
WISOTEN.INT;3             11/12    29-MAY-1986 19:04   (RE,RWED,RE,RE)
```

```
WISOVEC.INT;4                5/6        29-MAY-1986 19:04   (RE,RWED,RE,RE)
WP.INT;3                     2/2        29-MAY-1986 19:04   (RE,RWED,RE,RE)
WPN.INT;61                  17/18       29-MAY-1986 19:04   (RE,RWED,RE,RE)
WTEN.INT;2                   6/6        29-MAY-1986 19:04   (RE,RWED,RE,RE)
WTEST.ISO;1                  6/6         4-JUN-1986 10:29   (RE,RWED,RE,RE)
WZ.INT;3                     1/2        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBM.SPS;6                    2/2        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBMC.INT;12                  2/2        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBMI.INT;3                   3/4        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBMII.INT;3                  3/4        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBMO.INT;3                   3/4        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZBMPN.SPS;8                  2/2        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZWM.INT;3                    3/4        29-MAY-1986 19:04   (RE,RWED,RE,RE)
ZWME.INT;3                  19/20       29-MAY-1986 19:04   (RE,RWED,RE,RE)

Total of 175 files, 1891/1994 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.TRAMP]

CHNMVEC.COM;1                1/4        10-JUN-1986 14:21   (RE,RWED,RE,RE)
CHNTBTDOP.COM;1              1/4        10-JUN-1986 14:23   (RE,RWED,RE,RE)
CHNTRP.COM;1                 1/4        10-JUN-1986 14:21   (RE,RWED,RE,RE)
CLEBG.FOR;15                 9/10       10-JUN-1986 15:56   (RE,RWED,RE,RE)
CROSS.FOR;24                11/12       10-JUN-1986 15:56   (RE,RWED,RE,RE)
FINDPTR.FOR;16               3/4        10-JUN-1986 15:56   (RE,RWED,RE,RE)
HEADER.FPS;1                 7/8        10-JUN-1986 12:06   (RE,RWED,RE,RE)
HEADER.INC;1                 7/8        30-MAY-1986 13:47   (RE,RWED,RE,RE)
HEADTRP.FOR;37               2/2        10-JUN-1986 15:56   (RE,RWED,RE,RE)
LNKMVEC.COM;6                1/2        30-MAY-1986 13:47   (RE,RWED,RE,RE)
LNKMVEC.FPS;1                1/4         6-JUN-1986 17:35   (RE,RWED,RE,RE)
LNKTBTDOP.COM;8              1/2        30-MAY-1986 15:35   (RE,RWED,RE,RE)
LNKTBTDOP.FPS;1              1/4         6-JUN-1986 17:35   (RE,RWED,RE,RE)
LNKTRP.COM;30                1/2        30-MAY-1986 13:47   (RE,RWED,RE,RE)
LNKTRP.FPS;1                 1/4         6-JUN-1986 17:35   (RE,RWED,RE,RE)
MANAG.FOR;91                13/14       10-JUN-1986 15:56   (RE,RWED,RE,RE)
MVEC.COM;5                   1/2        30-MAY-1986 13:47   (RE,RWED,RE,RE)
MVEC.FOR;22                 15/16       10-JUN-1986 15:56   (RE,RWED,RE,RE)
MVEC.FPS;1                   1/4         6-JUN-1986 17:35   (RE,RWED,RE,RE)
MVECCOM.FPS;1                1/4        10-JUN-1986 12:06   (RE,RWED,RE,RE)
MVECCOM.INC;2                1/4         5-JUN-1986 22:16   (RE,RWED,RE,RE)
NEW.FPS;1                    1/4         6-JUN-1986 17:35   (RE,RWED,RE,RE)
OPENX.FOR;1                  2/2        28-FEB-1985 12:53   (RE,RWED,RE,RE)
OPSET.FOR;16                 7/8        10-JUN-1986 15:56   (RE,RWED,RE,RE)
SPEC.FOR;56                 30/30       10-JUN-1986 15:56   (RE,RWED,RE,RE)
TBTD.FOR;20                 14/14       10-JUN-1986 15:56   (RE,RWED,RE,RE)
TBTDOP.COM;4                 1/2        30-MAY-1986 13:47   (RE,RWED,RE,RE)
TBTDOP.FOR;20               28/28       10-JUN-1986 15:56   (RE,RWED,RE,RE)
TBTDOP.FPS;1                 1/4         6-JUN-1986 17:36   (RE,RWED,RE,RE)
TBTDOPCOM.FPS;1              1/4        10-JUN-1986 12:06   (RE,RWED,RE,RE)
TBTDOPCOM.INC;2              1/4         5-JUN-1986 22:28   (RE,RWED,RE,RE)
TRAMP.COM;20                 1/2        30-MAY-1986 13:47   (RE,RWED,RE,RE)
TRAMP.FOR;68                50/50       12-JUN-1986 10:42   (RE,RWED,RE,RE)
TRAMP.FPS;1                  1/4         6-JUN-1986 17:36   (RE,RWED,RE,RE)
TRMSBS.FOR;2                 3/4         6-MAR-1985 17:32   (RE,RWED,RE,RE)
TRPCOM.FPS;1                 8/8        10-JUN-1986 12:06   (RE,RWED,RE,RE)
TRPCOM.INC;25                8/8         5-JUN-1986 22:21   (RE,RWED,RE,RE)

Total of 37 files, 237/294 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.TRANS]
```

```
CHNRTRD.COM;1              1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LFRTRD.COM;7               1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LFRTRD.FPS;1               1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LFTRANS.COM;9              1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LFTRANS.FPS;1              1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LIBRA.FOR;2               34/34       11-NOV-1986 16:36  (RE,RWED,RE,RE)
LIBRB.FOR;1               39/40       11-NOV-1986 15:55  (RE,RWED,RE,RE)
LNKRTRD.COM;9              1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LNKRTRD.FPS;1              1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LNKTRANS.COM;18            1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
LNKTRANS.FPS;1             1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
NEW.FPS;1                  1/2        11-NOV-1986 15:55  (RE,RWED,RE,RE)
OPENBAB.FOR;1              3/4        11-NOV-1986 15:55  (RE,RWED,RE,RE)
RTRD.FOR;123              21/22       11-NOV-1986 15:55  (RE,RWED,RE,RE)
RTRDCOM.FOR;3              3/4        29-NOV-1984 10:22  (RE,RWED,RE,RE)
RTRDCOM.FPS;1              3/4        11-NOV-1986 15:55  (RE,RWED,RE,RE)
RTRDCOM.INC;1              3/4        11-NOV-1986 15:55  (RE,RWED,RE,RE)
TRANS.FOR;103             55/56       11-NOV-1986 16:35  (RE,RWED,RE,RE)
TRANSCOM.FOR;11            3/4        21-MAR-1985 13:37  (RE,RWED,RE,RE)
TRANSCOM.FPS;1             3/4        11-NOV-1986 15:55  (RE,RWED,RE,RE)
TRANSCOM.INC;1             3/4        11-NOV-1986 15:55  (RE,RWED,RE,RE)
TRANSTRD.FOR;30            8/8        11-NOV-1986 16:28  (RE,RWED,RE,RE)

Total of 22 files, 188/208 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.TRD]

TRD.TXT;1                  1/2        12-NOV-1986 21:51  (RE,RWE,RE,RE)

Total of 1 file, 1/2 blocks.

Directory SYS$THEORY:[BROWN.SHELLGROU.TXT]

B0004W.LPE;4              13/14       14-NOV-1985 20:15  (RE,RWED,RE,RE)
B0004W400.LRD;3           12/12       14-NOV-1985 20:15  (RE,RWED,RE,RE)
B0004W400.OBD;5            6/6        14-NOV-1985 20:16  (RE,RWED,RE,RE)
B4004W.LPE;4              13/14       14-NOV-1985 20:15  (RE,RWED,RE,RE)
DIFF.COM;2                 1/2        29-NOV-1984 09:45  (RE,RWED,RE,RE)
HEB.TXT;5                  1/2        29-NOV-1984 09:45  (RE,RWED,RE,RE)
HEBOF.TXT;1                1/2        29-NOV-1984 09:45  (RE,RWED,RE,RE)
HEBOL.TXT;1                4/4        29-NOV-1984 09:45  (RE,RWED,RE,RE)
HESH.COM;6                 1/2        29-NOV-1984 09:45  (RE,RWED,RE,RE)
HESH.TXT;5                 1/2        29-NOV-1984 09:45  (RE,RWED,RE,RE)
LET2FEB85.TXT;2            3/4        26-FEB-1985 12:01  (RE,RWED,RE,RE)
LETFEB85.TXT;3             5/6        13-FEB-1985 09:32  (RE,RWED,RE,RE)
NE20.COM;3                 8/8        14-NOV-1985 20:01  (RE,RWED,RE,RE)
NE20.LOG;3                24/24       14-NOV-1985 20:18  (RE,RWE,RE,RE)
NE20.REC;4                 5/6        14-NOV-1985 20:12  (RE,RWED,RE,RE)
PRALL.COM;10               1/2        14-NOV-1986 15:53  (RE,RWED,RE,RE)
SHELL.COM;3                1/2        14-AUG-1985 20:07  (RE,RWED,RE,RE)
SHELL.REC;1                2/2        14-AUG-1985 20:07  (RE,RWED,RE,RE)
SHELL.TXT;144            170/170      16-NOV-1986 18:31  (RE,RWED,RE,RE)
TED.COM;2                  1/2        14-AUG-1985 16:01  (RE,RWED,RE,RE)

Total of 20 files, 273/286 blocks.

Grand total of 32 directories, 676 files, 6935/7754 blocks.
```

```
Input to SHELL for the NE20 calculation
Input is shown in " "   <CR> stands for carriage return

 14-NOV-85 19:54:52

    |-------------------- OXBASH (Aug 1985) --------------------|
    |------- The Oxford-Buenos-Aires-MSU Shell-Model Code -------|
    |                                                           |
    | OXBASH82     : A.Etchegoyen, W.D.M.Rae and N.S.Godwin     |
    |              : contributors: W.A.Richter and C.H.Zimmerman |
    | MSU version  : B.A.Brown, W.E.Ormand and J.S.Winfield     |
    |-----------------------------------------------------------|

 Version with up to  96 m-states and  12 j-states allowed

 Macro versions of extended-integer subroutines are being used


 Name for command file (<CR>=SHELL) : "NE20"

 Option (VEC,DEN,STOP OR HELP) : "VEC"
 Single-particle state file : "SD"
 Total number of valence particles : "4"
 Any restrictions ? (Y/N) : "N" [or "<CR>"
 Interaction filename : "W"
 Min. J, Max. J : (2F) : "0.,2."
 Min. T, Max. T : (2F) : "0.,0."  [or "<CR>"]
                  [If nothing is entered for MAX then MAX=MIN]
 Bad J, T values (2F) (or <CR> for no more) : "1.,0."
 Bad J, T values (2F) (or <CR> for no more) : "<CR>"
 Parity (0=+VE, 1=-VE) : "0"

 B0004W    :BASIS PRD PROJ MATRIX LANCZOS
 Name of ground-state file (<CR> for present matrix filename ): "<CR>"

 B4004W    :BASIS PRD PROJ MATRIX LANCZOS

 Option (VEC,DEN,STOP OR HELP) : "DEN"
 D,C,1,2,T,4,IS,IV,IT,AT OR (H)ELP ? : "T"
 Initial state m-scheme eigenfunctions filename : "B0004W"
 Min.,Max. vectors reqd. (-1 FOR ALL) : "1,1"
 Final state m-scheme eigenfunctions filename : "B4004W"
 Min.,Max. vectors reqd. (-1 FOR ALL) : "1,5"
 B0004W    Min. J, Max. J : (F) : "0."
 B0004W    Min. T, Max. T : (F) : "0."
 Bad J, T values (2F) (or <CR> for no more) : "<CR>"
 B4004W    Min. J, Max. J : (F) : "2."
 B4004W    Min. T, Max. T : (F) : "0."
 Bad J, T values (2F) (or <CR> for no more) : "<CR>"
 Restrict coupling for operator :
 MVEC B0004W
 MVEC B4004W
 TRAMP B0004W B4004W     B0004W400.TRD


 Option (VEC,DEN,STOP OR HELP) : "EXIT"

 Type @NE20 to run small job at TTY ( < 3 mins CPUtime ) or
 type SUB NE20 to submit to SYS$BATCH
```

File for "NE20.COM"

```
$SET NOON              ! do not crash on errors
$SET VERIFY
$SET DEFAULT RSH:
$RUN SHEXE:BASIS
   0,   0,
SD                     ! SPSFIL
B0004W                 ! BASFIL
   4,                  ! number of particles
N                      ! Any restrictions ? (Y/N)
   0,   0,   0,        ! Jz*2, Tz*2 and parity
$RUN SHEXE:BASIS
   0,   0,
SD                     ! SPSFIL
B2004W                 ! BASFIL
   4,                  ! number of particles
N                      ! Any restrictions ? (Y/N)
   2,   0,   0,        ! Jz*2, Tz*2 and parity
$RU SHEXE:PREDICT
0                      ! ILPT (0,1,2)
B0004W                 ! BASFIL
B0004W                 ! PRJFIL name
   0,   0,             ! J*2, T*2
$RUN SHEXE:PROJ
   0,   0,   0,        ! IOUT,ILIM,IROU
B0004W                 ! BASFIL
B0004W                 ! PRJFIL
   0,   0,             ! J*2, T*2
$RUN SHEXE:MATRIX
0                      ! Line printer output (0,1, or 2)
W                      ! OPFIL
  1.00000,  0.96889,   ! SPE_NORM,ME_NORM
                       ! OPFIL
B0004W                 ! PRJFIL
B0004W                 ! MATFIL
$RUN SHEXE:LANCZOS
1,0,0,0,0,0,           ! ILPT,IWC,ICFN,IRAN
B0004W                 ! MATFIL
  10,   0,   0,        ! NEIG,NPRE,NFULL
Y                      ! Make ground-state reference file ? (Y/N)
B0004W                 ! Ground-state file name
Y                      ! Change SPE ? (Y/N)
      0.0000           ! Delta SPE(1D3/2    ) (will use old value if 0.)
      0.0000           ! Delta SPE(1D5/2    ) (will use old value if 0.)
      0.0000           ! Delta SPE(2S1/2    ) (will use old value if 0.)
$RUN SHEXE:BASIS
   0,   0,
SD                     ! SPSFIL
B4004W                 ! BASFIL
   4,                  ! number of particles
N                      ! Any restrictions ? (Y/N)
   4,   0,   0,        ! Jz*2, Tz*2 and parity
$RUN SHEXE:BASIS
   0,   0,
SD                     ! SPSFIL
B6004W                 ! BASFIL
   4,                  ! number of particles
N                      ! Any restrictions ? (Y/N)
   6,   0,   0,        ! Jz*2, Tz*2 and parity
```

```
$RU SHEXE:PREDICT
0                       ! ILPT (0,1,2)
B4004W                  ! BASFIL
B4004W                  ! PRJFIL name
   4,   0,              ! J*2, T*2
$RUN SHEXE:PROJ
   0,   0,   0,         ! IOUT,ILIM,IROU
B4004W                  ! BASFIL
B4004W                  ! PRJFIL
   4,   0,              ! J*2, T*2
$RUN SHEXE:MATRIX
0                       ! Line printer output (0,1, or 2)
W                       ! OPFIL
  1.00000,  0.96889,    ! SPE_NORM,ME_NORM
                        ! OPFIL
B4004W                  ! PRJFIL
B4004W                  ! MATFIL
$RUN SHEXE:LANCZOS
1,0,0,0,0,0,            ! ILPT,IWC,ICFN,IRAN
B4004W                  ! MATFIL
  10,   0,   0,         ! NEIG,NPRE,NFULL
N                       ! Make ground-state reference file ? (Y/N)
B0004W                  ! Ground-state file name
Y                       ! Change SPE ? (Y/N)
      0.0000            ! Delta SPE(1D3/2    ) (will use old value if 0.)
      0.0000            ! Delta SPE(1D5/2    ) (will use old value if 0.)
      0.0000            ! Delta SPE(2S1/2    ) (will use old value if 0.)
$RUN SHEXE:MVEC
B0004W                  ! EFIL initial
   1,   1,,             ! MIN,MAX
$RUN SHEXE:MVEC
B4004W                  ! EFIL final
   1,   5,,             ! MIN,MAX
$RUN SHEXE:TRAMP
1,1,                    ! ILPT (0,1), RESTRICT COUPLING (0,1)
T                       ! D,C,1,2,T,4,IS,IV,IT
B0004W                  ! EFIL initial A9
   1,   1,,             ! MIN,MAX
B4004W                  ! EFIL final A9
   1,   5,,             ! MIN,MAX
 0.0, 0.0, 0.0, 0.0,    ! DJMIN,DJMAX,DTMIN,DTMAX
B0004W400               ! TRDFIL
$RU SHEXE:RTRD
B0004W400.TRD           ! TRDFIL
$SET NOVERIFY
```

```
File: "NE20.LOG"

$SET DEFAULT RSH:
$RUN SHEXE:BASIS
Control integers ILPT,IBAS,IBPAIRS,MAX_BPAIRS :
  0  0  0  0
Single-particle state file :
                              SD.SPS
Name for basis data file :
                             B0004W.BAS
Number of particles :
                            4
Any restrictions ? (Y/N) :
                           N
Jz*2, Tz*2 and Parity (0=+VE, 1=-VE) :
                                        0   0   0

NO. OF TRIAL PATTERNS           :        7483
M-SCHEME BASIS DIMENSION        :         640
NO. OF BASIS PARTITIONS STORED  :          15
NO. LOOPS FOR TABLE LOOKUP      :        3732


NO. OF STATES WITH
     2*J :     0     2     4     6     8    10    12    14    16    18

2*T =  0 :    21    31    56    45    44    24    17     5     3     0
2*T =  2 :    16    54    66    69    50    34    16     7     1     0
2*T =  4 :     9    12    21    15    15     6     3     0     0     0

 CPU TIME = 0:00:02.85
 ELAPSED TIME = 00:00:06.76



$RUN SHEXE:BASIS
Control integers ILPT,IBAS,IBPAIRS,MAX_BPAIRS :
  0  0  0  0
Single-particle state file :
                              SD.SPS
Name for basis data file :
                             B2004W.BAS
Number of particles :
                            4
Any restrictions ? (Y/N) :
                           N
Jz*2, Tz*2 and Parity (0=+VE, 1=-VE) :
                                        2   0   0

NO. OF TRIAL PATTERNS           :        6169
M-SCHEME BASIS DIMENSION        :         594
NO. OF BASIS PARTITIONS STORED  :          14
NO. LOOPS FOR TABLE LOOKUP      :        3732


NO. OF STATES WITH
     2*J :     2     4     6     8    10    12    14    16    18    20

2*T =  0 :    31    56    45    44    24    17     5     3     0     0
2*T =  2 :    54    66    69    50    34    16     7     1     0     0
```

```
2*T =  4 :     12     21     15     15      6      3      0      0      0      0

 CPU TIME = 0:00:02.68
 ELAPSED TIME = 00:00:06.11



$RU SHEXE:PREDICT
ILPT (0,1,2), Logical solution for T (Y/N):
0
Name of basis :
B0004W.BAS
Filename for good m-scheme set :
B0004W.PRD
Value of J*2, T*2 :
  0  0
 CPU TIME = 0:00:10.09
 ELAPSED TIME = 00:00:13.24



$RUN SHEXE:PROJ
This is "final" PROJ (based on POSTPROJ)
Control integers ILPT,ILIM,IROU :
  0  0  0
Name of basis :
B0004W.BAS
Name for projected-basis :
B0004W.PRJ
Value of J*2, T*2 :
  0  0
***** Running with prediction file

Projection calculation successfully completed

CPU TIME spent in JTPROJ    = 0: 0: 3.48
CPU TIME spent in ORTHOG    = 0: 0: 0. 3
 CPU TIME = 0:00:06.68
 ELAPSED TIME = 00:00:15.39



$RUN SHEXE:MATRIX
Lineprinter file (0=None,1=<Oi|H|Oj>M.E.,2=as 1 plus <m|H|Oj>M.E.) :
0
List of up to 9 interactions (*.OP files), <CR> to end list
Name of 2-body interaction #1 :
W         .OP
One and two body Hamiltonians normalisations (2F) :
   1.0000   0.9689
Name of 2-body interaction #2 :
Name of projected-basis :
B0004W  .PRJ
Name for matrix :
B0004W.MAT
 ( Using W         .OP from area         SHOP:          )
MATRIX DIMENSION  =    21
 CPU TIME = 0:00:09.36
```

```
 ELAPSED TIME = 00:00:12.28

$RUN SHEXE:LANCZOS

ILPT = 1 for formatted *.LPE output; = 0 for no output
IWC  = 1 for output to Etchegoyens weak coupling program
ICFN = # of states on *.LPE with partition occupations
       default is 10 (ICFN=0) and ICFN=-1 for none
IRAN = # for multiplicative factor to added random matix
       in units of keV/10

Control integers ILPT,IWC,ICFN,IRAN :
   1   0   0   0
Matrix filename :
B0004W.MAT
NEIG,NPRE,NFULL: No. of E-vects. (Max=  21,-1 for all), PRE-ITER, CONVERGENC
CRITERIA (keV)
Defaults =   10   3    0.50 :
MATDIM,NEIG,NPRE,CRITERIA =   21  10   3    0.50
  10   3    0.50
Calculate ground state energy (Y/N) ? :
Y
Name of ground-state file :
B0004W.GND
 ( Using W        .INT from area        SHSPS:        )
Change the single_particle energies (Y/N) ? :
Y
Single-particle energies :
1D3/2     Shell :     1.647 MeV  NONZERO VALUE FOR NEW ENERGY :
1D5/2     Shell :    -3.948 MeV  NONZERO VALUE FOR NEW ENERGY :
2S1/2     Shell :    -3.164 MeV  NONZERO VALUE FOR NEW ENERGY :

      210 OFF DIAGONAL MATRIX ELEMENTS
      144 NON ZERO OFF DIAGONAL MATRIX ELEMENTS
      141 OFF DIAGONAL MATRIX ELEMENTS > 0.0001
      141 OFF DIAGONAL MATRIX ELEMENTS > 0.001
    1.4549 AVERAGE V**2 FOR OFF DIAGONAL ME
ALL FILES HAVE NOW BEEN OPENED
MATRIX DIMENSION =         21
LOWEST EIGENVALUE =   -40.492  FOR VECTOR B0004W

Lowest eigenvalues obtained from LANCZOS in next to last iteration
   -40.492   -33.736   -28.516   -26.661   -24.473
   -24.083   -22.685   -20.149   -18.981   -17.772


Lowest eigenvalues obtained from LANCZOS in last iteration
   -40.492   -33.736   -28.516   -26.661   -24.473
   -24.083   -22.685   -20.149   -18.983   -17.777


Eigenvalue of ground state :    -40.492

Energy levels :
     0.000     6.756    11.977    13.831    16.020
    16.409    17.807    20.343    21.509    22.715

# OF ITERATIONS :    21
CPU TIME spent in EIGENH    = 0: 0: 0. 0
CPU TIME spent in LANCZOS   = 0: 0:12.99
```

```
 CPU TIME = 0:00:14.53
 ELAPSED TIME = 00:00:26.44

$RUN SHEXE:BASIS
Control integers ILPT,IBAS,IBPAIRS,MAX_BPAIRS :
  0  0  0  0
Single-particle state file :
                              SD.SPS
Name for basis data file :
                             B4004W.BAS
Number of particles :
                           4
Any restrictions ? (Y/N) :
                            N
Jz*2, Tz*2 and Parity (0=+VE, 1=-VE) :
                                        4   0   0

NO. OF TRIAL PATTERNS           :        4662
M-SCHEME BASIS DIMENSION        :         497
NO. OF BASIS PARTITIONS STORED  :          14
NO. LOOPS FOR TABLE LOOKUP      :        3732


NO. OF STATES WITH
     2*J :     4     6     8    10    12    14    16    18    20    22

2*T =  0 :    56    45    44    24    17     5     3     0     0     0
2*T =  2 :    66    69    50    34    16     7     1     0     0     0
2*T =  4 :    21    15    15     6     3     0     0     0     0     0

 CPU TIME = 0:00:02.21
 ELAPSED TIME = 00:00:05.37



$RUN SHEXE:BASIS
Control integers ILPT,IBAS,IBPAIRS,MAX_BPAIRS :
  0  0  0  0
Single-particle state file :
                              SD.SPS
Name for basis data file :
                             B6004W.BAS
Number of particles :
                           4
Any restrictions ? (Y/N) :
                            N
Jz*2, Tz*2 and Parity (0=+VE, 1=-VE) :
                                        6   0   0

NO. OF TRIAL PATTERNS           :        3203
M-SCHEME BASIS DIMENSION        :         354
NO. OF BASIS PARTITIONS STORED  :          13
NO. LOOPS FOR TABLE LOOKUP      :        3732


NO. OF STATES WITH
     2*J :     6     8    10    12    14    16    18    20    22    24

2*T =  0 :    45    44    24    17     5     3     0     0     0     0
2*T =  2 :    69    50    34    16     7     1     0     0     0     0
```

```
2*T =  4 :     15     15      6      3      0      0      0      0      0      0

 CPU TIME = 0:00:01.86
 ELAPSED TIME = 00:00:04.84


$RU SHEXE:PREDICT
ILPT (0,1,2), Logical solution for T (Y/N):
0
Name of basis :
B4004W.BAS
Filename for good m-scheme set :
B4004W.PRD
Value of J*2, T*2 :
  4  0
 CPU TIME = 0:00:09.83
 ELAPSED TIME = 00:00:11.22



$RUN SHEXE:PROJ
This is "final" PROJ (based on POSTPROJ)
Control integers ILPT,ILIM,IROU :
  0  0  0
Name of basis :
B4004W.BAS
Name for projected-basis :
B4004W.PRJ
Value of J*2, T*2 :
  4  0
***** Running with prediction file

Projection calculation successfully completed

CPU TIME spent in JTPROJ    = 0: 0: 3.39
CPU TIME spent in ORTHOG    = 0: 0: 0.15
 CPU TIME = 0:00:07.24
 ELAPSED TIME = 00:00:15.79



$RUN SHEXE:MATRIX
Lineprinter file (0=None,1=<Oi|H|Oj>M.E.,2=as 1 plus <m|H|Oj>M.E.) :
0
List of up to 9 interactions (*.OP files), <CR> to end list
Name of 2-body interaction #1 :
W         .OP
One and two body Hamiltonians normalisations (2F) :
   1.0000   0.9689
Name of 2-body interaction #2 :
Name of projected-basis :
B4004W  .PRJ
Name for matrix :
B4004W.MAT
 ( Using W         .OP from area         SHOP:          )
MATRIX DIMENSION  =    56
 CPU TIME = 0:00:20.34
```

```
 ELAPSED TIME = 00:00:24.24

$RUN SHEXE:LANCZOS

ILPT = 1 for formatted *.LPE output; = 0 for no output
IWC  = 1 for output to Etchegoyens weak coupling program
ICFN = # of states on *.LPE with partition occupations
         default is 10 (ICFN=0) and ICFN=-1 for none
IRAN = # for multiplicative factor to added random matix
         in units of keV/10

Control integers ILPT,IWC,ICFN,IRAN :
   1   0   0   0
Matrix filename :
B4004W.MAT
NEIG,NPRE,NFULL: No. of E-vects. (Max=  56,-1 for all), PRE-ITER, CONVERGENC
CRITERIA (keV)
Defaults =   10   3    0.50 :
MATDIM,NEIG,NPRE,CRITERIA =   56  10   3    0.50
  10   3    0.50
Calculate ground state energy (Y/N) ? :
N
Name of ground-state file :
B0004W.GND
 ( Using W        .INT from area        SHSPS:        )
Change the single_particle energies (Y/N) ? :
Y
Single-particle energies :
1D3/2     Shell :     1.647 MeV  NONZERO VALUE FOR NEW ENERGY :
1D5/2     Shell :    -3.948 MeV  NONZERO VALUE FOR NEW ENERGY :
2S1/2     Shell :    -3.164 MeV  NONZERO VALUE FOR NEW ENERGY :

     1540 OFF DIAGONAL MATRIX ELEMENTS
     1212 NON ZERO OFF DIAGONAL MATRIX ELEMENTS
     1078 OFF DIAGONAL MATRIX ELEMENTS > 0.0001
     1078 OFF DIAGONAL MATRIX ELEMENTS > 0.001
    0.7863 AVERAGE V**2 FOR OFF DIAGONAL ME
ALL FILES HAVE NOW BEEN OPENED
MATRIX DIMENSION =         56
LOWEST EIGENVALUE =   -38.716  FOR VECTOR B4004W

Lowest eigenvalues obtained from LANCZOS in next to last iteration
   -38.716   -33.177   -30.053   -29.756   -27.537
   -27.185   -26.930   -25.535   -25.236   -24.138


Lowest eigenvalues obtained from LANCZOS in last iteration
   -38.716   -33.177   -30.053   -29.756   -27.537
   -27.185   -26.930   -25.535   -25.236   -24.138


Eigenvalue of ground state :    -40.492

Energy levels :
     1.776     7.316    10.439    10.736    12.955
    13.308    13.563    14.958    15.256    16.355

# OF ITERATIONS :    54
CPU TIME spent in EIGENH   = 0: 0: 0. 0
CPU TIME spent in LANCZOS  = 0: 0:59.25
```

```
 CPU TIME = 0:01:01.29
 ELAPSED TIME = 00:01:49.12

$RUN SHEXE:MVEC
Name of .EIG file :
B0004W.EIG
Min and Max no. of vectors reqd. (-1 for all)  :
   1   1
 CPU TIME = 0:00:00.40
 ELAPSED TIME = 00:00:03.23

$RUN SHEXE:MVEC
Name of .EIG file :
B4004W.EIG
Min and Max no. of vectors reqd. (-1 for all)  :
   1   5
 CPU TIME = 0:00:01.40
 ELAPSED TIME = 00:00:10.53

$RUN SHEXE:TRAMP
ILPT (0,1), RESTRICT COUPLING (0 for Y, 1 for N) :
 1 1
D,C,1,2,T,4,IS,IV,IT or (H)elp ? :
T
Initial state m-scheme eigenfunctions filename  :
B0004W.VEC
Min. and max. no. of vectors reqd. (-1 for all)  :
    1    1
Final state m-scheme eigenfunctions filename :
B4004W.VEC
Min. and max. no. of vectors reqd. (-1 for all)  :
    1    5
Tensor ranks for operator restriction: DJMIN,DJMAX,DTMIN,DTMAX (4F)
Output filename  (-1 for none) :
B0004W400.

 CPU TIME = 0:00:13.50
 ELAPSED TIME = 00:00:31.09

$RU SHEXE:RTRD

NAME FOR (*.TRD OR *.T2N) INPUT FILE (A13) (<CR> TO EXIT) :
Output is in file B0004W400.OBD
FORTRAN STOP
$SET NOVERIFY
  BROWN        job terminated at 14-NOV-1985 20:01:50.94

  Accounting information:
  Buffered I/O count:          917      Peak working set size:    256
  Direct I/O count:           1867      Peak page file size:    28463
  Page faults:               21741      Mounted volumes:            0
  Elapsed CPU time:     0 00:03:21.63   Elapsed time:     0 00:05:52.59
```

File: "B0004W.LPE"

OUTPUT OF LANCZOS

14-NOV-85 19:57:48

```
   |-------------------- OXBASH (Aug 1985) ---------------------|
   |------- The Oxford-Buenos-Aires-MSU Shell-Model Code -------|
   |                                                            |
   | OXBASH82     : A.Etchegoyen, W.D.M.Rae and N.S.Godwin      |
   |              : contributors: W.A.Richter and C.H.Zimmerman |
   | MSU version  : B.A.Brown, W.E.Ormand and J.S.Winfield      |
   |------------------------------------------------------------|
```

Version with up to 96 m-states and 12 j-states allowed

Macro versions of extended-integer subroutines are being used

# of valence particles : 4

J : 0 (Projected from state with Jz = 0)
T : 0 (Projected from state with Tz = 0)
Parity : +VE

Model space (*.SPS) name : SD
M-scheme basis (*.BAS) name : B0004W

| | | SPE_NORM | ME_NORM |
|---|---|---|---|
| TBME (*.INT) name (1) | : W | 1.00000 | 0.96889 |

Eigenvector (*.EIG) name : B0004W
Ground state (*.GND) name : B0004W -40.492 = Eigenvalue of grou
Matrix dimension : 21

# of particles allowed in each j-level  Single-particle energies for eac

| SHELL | SUB SHELL MIN | SUB SHELL MAX | MAJOR SHELL MIN | MAJOR SHELL MAX | Total SPE | Delta SPE from input | ------ SPE 1 |
|---|---|---|---|---|---|---|---|
| 1D3/2 | 0 | 4 | 4 | 4 | 1.6466 | 0.0000 | 1.6466 |
| 1D5/2 | 0 | 4 | | | -3.9478 | 0.0000 | -3.9478 |
| 2S1/2 | 0 | 4 | | | -3.1635 | 0.0000 | -3.1635 |

# iterations required : 21
Convergence criteria (keV) : 0.50

Lowest eigenvalues obtained from LANCZOS in next to last iteration

| | | | | |
|---|---|---|---|---|
| -40.492 | -33.736 | -28.516 | -26.661 | -24.473 |
| -24.083 | -22.685 | -20.149 | -18.981 | -17.772 |

Lowest eigenvalues obtained from LANCZOS in last iteration

| | | | | |
|---|---|---|---|---|
| -40.492 | -33.736 | -28.516 | -26.661 | -24.473 |
| -24.083 | -22.685 | -20.149 | -18.983 | -17.777 |

Eigenvalue of ground state : -40.492

CONFIGURATIONS (1D3/2 ,1D5/2 ,2S1/2 )
[Number of broken pairs]

STATE 1 ( 0.000 MEV ) :

| | | | |
|---|---|---|---|
| 3.1%( 1, 2, 1)[ 2] | 13.3%( 2, 2, 0)[ 0] | 7.8%( 1, 3, 0)[ 2] | 1.4%( 2, 1 |
| 8.6%( 0, 3, 1)[ 2] | 0.1%( 3, 1, 0)[ 2] | 9.2%( 1, 1, 2)[ 2] | 26.1%( 0, 4 |
| 22.5%( 0, 2, 2)[ 0] | 0.2%( 3, 0, 1)[ 2] | 2.4%( 2, 0, 2)[ 0] | 0.4%( 4, 0 |
| 5.1%( 0, 0, 4)[ 0] | | | |

STATE 2 ( 6.756 MEV ) :

| | | | |
|---|---|---|---|
| 0.5%( 1, 2, 1)[ 2] | 8.0%( 2, 2, 0)[ 0] | 4.8%( 1, 3, 0)[ 2] | 0.4%( 2, 1 |
| 1.0%( 0, 3, 1)[ 2] | 0.0%( 3, 1, 0)[ 2] | 9.2%( 1, 1, 2)[ 2] | 34.3%( 0, 4 |
| 16.7%( 0, 2, 2)[ 0] | 0.0%( 3, 0, 1)[ 2] | 2.7%( 2, 0, 2)[ 0] | 0.2%( 4, 0 |
| 22.3%( 0, 0, 4)[ 0] | | | |

STATE 3 ( 11.977 MEV ) :

| | | | |
|---|---|---|---|
| 10.4%( 1, 2, 1)[ 2] | 4.2%( 2, 2, 0)[ 0] | 0.6%( 1, 3, 0)[ 2] | 2.5%( 2, 1 |
| 42.8%( 0, 3, 1)[ 2] | 0.0%( 3, 1, 0)[ 2] | 1.4%( 1, 1, 2)[ 2] | 2.6%( 0, 4 |
| 2.4%( 0, 2, 2)[ 0] | 0.2%( 3, 0, 1)[ 2] | 0.4%( 2, 0, 2)[ 0] | 0.2%( 4, 0 |
| 32.3%( 0, 0, 4)[ 0] | | | |

STATE 4 ( 13.831 MEV ) :

| | | | |
|---|---|---|---|
| 3.3%( 1, 2, 1)[ 2] | 26.7%( 2, 2, 0)[ 0] | 12.8%( 1, 3, 0)[ 2] | 4.4%( 2, 1 |
| 0.0%( 0, 3, 1)[ 2] | 0.1%( 3, 1, 0)[ 2] | 5.6%( 1, 1, 2)[ 2] | 29.2%( 0, 4 |
| 13.4%( 0, 2, 2)[ 0] | 0.5%( 3, 0, 1)[ 2] | 0.5%( 2, 0, 2)[ 0] | 1.8%( 4, 0 |
| 1.5%( 0, 0, 4)[ 0] | | | |

STATE 5 ( 16.020 MEV ) :

| | | | |
|---|---|---|---|
| 4.1%( 1, 2, 1)[ 2] | 5.6%( 2, 2, 0)[ 0] | 0.7%( 1, 3, 0)[ 2] | 0.6%( 2, 1 |
| 17.1%( 0, 3, 1)[ 2] | 0.0%( 3, 1, 0)[ 2] | 1.1%( 1, 1, 2)[ 2] | 2.5%( 0, 4 |
| 37.3%( 0, 2, 2)[ 0] | 0.0%( 3, 0, 1)[ 2] | 3.9%( 2, 0, 2)[ 0] | 0.7%( 4, 0 |
| 26.4%( 0, 0, 4)[ 0] | | | |

STATE 6 ( 16.409 MEV ) :

| | | | |
|---|---|---|---|
| 2.0%( 1, 2, 1)[ 2] | 5.8%( 2, 2, 0)[ 0] | 1.2%( 1, 3, 0)[ 2] | 1.4%( 2, |
| 0.3%( 0, 3, 1)[ 2] | 0.0%( 3, 1, 0)[ 2] | 3.1%( 1, 1, 2)[ 2] | 76.5%( 0, |
| 6.2%( 0, 2, 2)[ 0] | 0.4%( 3, 0, 1)[ 2] | 0.3%( 2, 0, 2)[ 0] | 0.0%( 4, |
| 2.7%( 0, 0, 4)[ 0] | | | |

STATE 7 ( 17.807 MEV ) :

```
   8.2%( 1, 2, 1)[ 2]  12.8%( 2, 2, 0)[ 0]  19.2%( 1, 3, 0)[ 2]   4.7%( 2, 1
   2.3%( 0, 3, 1)[ 2]   0.0%( 3, 1, 0)[ 2]   0.6%( 1, 1, 2)[ 2]   0.4%( 0, 4
  39.9%( 0, 2, 2)[ 0]   0.9%( 3, 0, 1)[ 2]   4.7%( 2, 0, 2)[ 0]   2.1%( 4, 0
   4.3%( 0, 0, 4)[ 0]
```

STATE 8 ( 20.343 MEV ) :

```
   4.0%( 1, 2, 1)[ 2]   8.7%( 2, 2, 0)[ 0]  21.5%( 1, 3, 0)[ 2]  11.9%( 2, 1
   1.6%( 0, 3, 1)[ 2]   0.8%( 3, 1, 0)[ 2]  18.9%( 1, 1, 2)[ 2]  15.5%( 0, 4
   9.1%( 0, 2, 2)[ 0]   0.4%( 3, 0, 1)[ 2]   3.6%( 2, 0, 2)[ 0]   0.1%( 4, 0
   3.8%( 0, 0, 4)[ 0]
```

STATE 9 ( 21.509 MEV ) :

```
  38.6%( 1, 2, 1)[ 2]  13.2%( 2, 2, 0)[ 0]   3.5%( 1, 3, 0)[ 2]  13.5%( 2, 1
   6.5%( 0, 3, 1)[ 2]   2.1%( 3, 1, 0)[ 2]   7.2%( 1, 1, 2)[ 2]   1.4%( 0, 4
   4.4%( 0, 2, 2)[ 0]   0.7%( 3, 0, 1)[ 2]   8.2%( 2, 0, 2)[ 0]   0.0%( 4, 0
   0.7%( 0, 0, 4)[ 0]
```

STATE 10 ( 22.715 MEV ) :

```
  47.4%( 1, 2, 1)[ 2]   4.0%( 2, 2, 0)[ 0]   0.6%( 1, 3, 0)[ 2]  17.7%( 2, 1
  12.5%( 0, 3, 1)[ 2]   0.1%( 3, 1, 0)[ 2]   5.8%( 1, 1, 2)[ 2]   0.2%( 0, 4
   7.1%( 0, 2, 2)[ 0]   2.5%( 3, 0, 1)[ 2]   1.2%( 2, 0, 2)[ 0]   0.1%( 4, 0
   0.5%( 0, 0, 4)[ 0]
```

| NO | ENERGY LEVEL | AVERAGE NO OF PARTICLES IN EACH J-LEVEL | | |
|---|---|---|---|---|
| | | 1D3/2 | 1D5/2 | 2S1/2 |
| | S-P ENERGY: | 1.647 | -3.948 | -3.164 |
| 1 | 0.000 | 0.5662 | 2.4179 | 1.0159 |
| 2 | 6.756 | 0.3752 | 2.1440 | 1.4808 |
| 3 | 11.977 | 0.2778 | 1.7865 | 1.9356 |
| 4 | 13.831 | 0.9420 | 2.5236 | 0.5344 |
| 5 | 16.020 | 0.2901 | 1.5907 | 2.1192 |
| 6 | 16.409 | 0.2259 | 3.4317 | 0.3423 |
| 7 | 17.807 | 0.8350 | 1.9301 | 1.2349 |
| 8 | 20.343 | 0.9678 | 2.0697 | 0.9625 |
| 9 | 21.509 | 1.2757 | 1.7076 | 1.0167 |
| 10 | 22.715 | 1.0815 | 1.8132 | 1.1052 |

File: "B4004W.LPE"

OUTPUT OF LANCZOS

14-NOV-85 20:00:54

```
   |-------------------- OXBASH (Aug 1985) --------------------|
   |------- The Oxford-Buenos-Aires-MSU Shell-Model Code -------|
   |                                                           |
   | OXBASH82     : A.Etchegoyen, W.D.M.Rae and N.S.Godwin     |
   |              : contributors: W.A.Richter and C.H.Zimmerman |
   | MSU version  : B.A.Brown, W.E.Ormand and J.S.Winfield     |
   |-----------------------------------------------------------|
```

Version with up to 96 m-states and 12 j-states allowed

Macro versions of extended-integer subroutines are being used

# of valence particles : 4

J : 2 (Projected from state with Jz = 2)
T : 0 (Projected from state with Tz = 0)
Parity : +VE

Model space (*.SPS) name : SD
M-scheme basis (*.BAS) name : B4004W

| | | SPE_NORM | ME_NORM |
|---|---|---|---|
| TBME (*.INT) name (1) | : W | 1.00000 | 0.96889 |

Eigenvector (*.EIG) name : B4004W
Ground state (*.GND) name : B0004W -40.492 = Eigenvalue of grour
Matrix dimension : 56

# of particles allowed in each j-level Single-particle energies for eacl

| SHELL | SUB SHELL MIN | SUB SHELL MAX | MAJOR SHELL MIN | MAJOR SHELL MAX | Total SPE | Delta SPE from input | ------ SPE F 1 |
|---|---|---|---|---|---|---|---|
| 1D3/2 | 0 | 4 | 4 | 4 | 1.6466 | 0.0000 | 1.6466 |
| 1D5/2 | 0 | 4 | | | -3.9478 | 0.0000 | -3.9478 |
| 2S1/2 | 0 | 4 | | | -3.1635 | 0.0000 | -3.1635 |

# iterations required : 54
Convergence criteria (keV) : 0.50

Lowest eigenvalues obtained from LANCZOS in next to last iteration
-38.716 -33.177 -30.053 -29.756 -27.537
-27.185 -26.930 -25.535 -25.236 -24.138

Lowest eigenvalues obtained from LANCZOS in last iteration
-38.716 -33.177 -30.053 -29.756 -27.537
-27.185 -26.930 -25.535 -25.236 -24.138

Eigenvalue of ground state : -40.492

CONFIGURATIONS (1D3/2 ,1D5/2 ,2S1/2 )
[Number of broken pairs]

STATE 1 ( 1.776 MEV ) :

| | | | |
|---|---|---|---|
| 15.2%( 1, 2, 1)[ 2] | 5.3%( 2, 2, 0)[ 0] | 6.3%( 1, 3, 0)[ 2] | 5.1%( 2, 1 |
| 26.4%( 0, 3, 1)[ 2] | 0.3%( 3, 1, 0)[ 2] | 2.8%( 1, 1, 2)[ 2] | 16.0%( 0, 4 |
| 8.7%( 0, 2, 2)[ 0] | 0.7%( 3, 0, 1)[ 2] | 0.6%( 2, 0, 2)[ 0] | 0.1%( 4, 0 |
| 10.1%( 0, 1, 3)[ 2] | 2.4%( 1, 0, 3)[ 2] | | |

STATE 2 ( 7.316 MEV ) :

| | | | |
|---|---|---|---|
| 5.4%( 1, 2, 1)[ 2] | 11.4%( 2, 2, 0)[ 0] | 6.3%( 1, 3, 0)[ 2] | 1.8%( 2, 1 |
| 9.1%( 0, 3, 1)[ 2] | 0.2%( 3, 1, 0)[ 2] | 0.3%( 1, 1, 2)[ 2] | 53.4%( 0 |
| 2.4%( 0, 2, 2)[ 0] | 0.1%( 3, 0, 1)[ 2] | 0.0%( 2, 0, 2)[ 0] | 0.2%( |
| 8.0%( 0, 1, 3)[ 2] | 1.4%( 1, 0, 3)[ 2] | | |

STATE 3 ( 10.439 MEV ) :

| | | | |
|---|---|---|---|
| 3.9%( 1, 2, 1)[ 2] | 3.1%( 2, 2, 0)[ 0] | 32.7%( 1, 3, 0)[ 2] | 0.5%( 2, 1 |
| 6.8%( 0, 3, 1)[ 2] | 2.5%( 3, 1, 0)[ 2] | 2.0%( 1, 1, 2)[ 2] | 20.7%( 0, 4 |
| 17.6%( 0, 2, 2)[ 0] | 0.0%( 3, 0, 1)[ 2] | 0.1%( 2, 0, 2)[ 0] | 0.1%( 4, 0 |
| 7.0%( 0, 1, 3)[ 2] | 3.0%( 1, 0, 3)[ 2] | | |

STATE 4 ( 10.736 MEV ) :

| | | | |
|---|---|---|---|
| 1.5%( 1, 2, 1)[ 2] | 2.6%( 2, 2, 0)[ 0] | 13.7%( 1, 3, 0)[ 2] | 0.1%( 2, 1 |
| 9.3%( 0, 3, 1)[ 2] | 1.0%( 3, 1, 0)[ 2] | 10.3%( 1, 1, 2)[ 2] | 25.5%( 0, 4 |
| 16.7%( 0, 2, 2)[ 0] | 0.0%( 3, 0, 1)[ 2] | 1.8%( 2, 0, 2)[ 0] | 0.0%( 4, 0 |
| 13.4%( 0, 1, 3)[ 2] | 4.0%( 1, 0, 3)[ 2] | | |

STATE 5 ( 12.955 MEV ) :

| | | | |
|---|---|---|---|
| 4.7%( 1, 2, 1)[ 2] | 3.6%( 2, 2, 0)[ 0] | 4.5%( 1, 3, 0)[ 2] | 0.7%( 2, 1 |
| 3.4%( 0, 3, 1)[ 2] | 0.2%( 3, 1, 0)[ 2] | 9.2%( 1, 1, 2)[ 2] | 24.3%( 0, |
| 36.3%( 0, 2, 2)[ 0] | 0.1%( 3, 0, 1)[ 2] | 1.1%( 2, 0, 2)[ 0] | 0.0%( 4, |
| 11.1%( 0, 1, 3)[ 2] | 0.8%( 1, 0, 3)[ 2] | | |

STATE 6 ( 13.308 MEV ) :

| | | | |
|---|---|---|---|
| 28.9%( 1, 2, 1)[ 2] | 3.4%( 2, 2, 0)[ 0] | 9.5%( 1, 3, 0)[ 2] | 1.6%( 2, |
| 35.5%( 0, 3, 1)[ 2] | 0.6%( 3, 1, 0)[ 2] | 4.9%( 1, 1, 2)[ 2] | 0.6%( 0, |
| 7.9%( 0, 2, 2)[ 0] | 2.3%( 3, 0, 1)[ 2] | 0.2%( 2, 0, 2)[ 0] | 0.1%( 4, |
| 0.2%( 0, 1, 3)[ 2] | 4.2%( 1, 0, 3)[ 2] | | |

STATE 7 ( 13.563 MEV ) :

| | | | |
|---|---|---|---|
| 21.0%( 1, 2, 1)[ 2] | 6.8%( 2, 2, 0)[ 0] | 14.6%( 1, 3, 0)[ 2] | 7.7%( 2, 1 |
| 24.8%( 0, 3, 1)[ 2] | 0.6%( 3, 1, 0)[ 2] | 5.3%( 1, 1, 2)[ 2] | 10.7%( 0, 4 |
| 6.4%( 0, 2, 2)[ 0] | 0.1%( 3, 0, 1)[ 2] | 0.4%( 2, 0, 2)[ 0] | 0.0%( 4, 0 |
| 1.3%( 0, 1, 3)[ 2] | 0.4%( 1, 0, 3)[ 2] | | |

STATE 8 ( 14.958 MEV ) :

| | | | |
|---|---|---|---|
| 17.5%( 1, 2, 1)[ 2] | 12.5%( 2, 2, 0)[ 0] | 16.7%( 1, 3, 0)[ 2] | 1.1%( 2, 1 |
| 2.5%( 0, 3, 1)[ 2] | 0.6%( 3, 1, 0)[ 2] | 10.8%( 1, 1, 2)[ 2] | 10.2%( 0, 4 |
| 10.0%( 0, 2, 2)[ 0] | 0.5%( 3, 0, 1)[ 2] | 0.1%( 2, 0, 2)[ 0] | 0.1%( 4, 0 |
| 15.6%( 0, 1, 3)[ 2] | 1.7%( 1, 0, 3)[ 2] | | |

STATE 9 ( 15.256 MEV ) :

| | | | |
|---|---|---|---|
| 8.6%( 1, 2, 1)[ 2] | 4.9%( 2, 2, 0)[ 0] | 15.7%( 1, 3, 0)[ 2] | 0.8%( 2, 1 |
| 20.5%( 0, 3, 1)[ 2] | 0.7%( 3, 1, 0)[ 2] | 4.9%( 1, 1, 2)[ 2] | 21.8%( 0, 4 |
| 5.5%( 0, 2, 2)[ 0] | 0.1%( 3, 0, 1)[ 2] | 0.3%( 2, 0, 2)[ 0] | 0.1%( 4, 0 |
| 12.0%( 0, 1, 3)[ 2] | 4.0%( 1, 0, 3)[ 2] | | |

STATE 10 ( 16.355 MEV ) :

| | | | |
|---|---|---|---|
| 11.7%( 1, 2, 1)[ 2] | 6.1%( 2, 2, 0)[ 0] | 26.2%( 1, 3, 0)[ 2] | 7.4%( 2, 1 |
| 21.7%( 0, 3, 1)[ 2] | 1.8%( 3, 1, 0)[ 2] | 6.7%( 1, 1, 2)[ 2] | 6.0%( 0, 4 |
| 4.5%( 0, 2, 2)[ 0] | 0.4%( 3, 0, 1)[ 2] | 0.4%( 2, 0, 2)[ 0] | 0.1%( 4, 0 |
| 3.3%( 0, 1, 3)[ 2] | 3.6%( 1, 0, 3)[ 2] | | |

| NO | ENERGY LEVEL | AVERAGE NO OF PARTICLES IN EACH J-LEVEL | | |
|---|---|---|---|---|
| | | 1D3/2 | 1D5/2 | 2S1/2 |
| | S-P ENERGY: | 1.647 | -3.948 | -3.164 |
| 1 | 1.776 | 0.5226 | 2.3878 | 1.0896 |
| 2 | 7.316 | 0.4140 | 3.0861 | 0.4999 |
| 3 | 10.439 | 0.5717 | 2.6239 | 0.8043 |
| 4 | 10.736 | 0.4179 | 2.3739 | 1.2082 |
| 5 | 12.955 | 0.3098 | 2.3125 | 1.3777 |
| 6 | 13.308 | 0.6696 | 2.2543 | 1.0760 |
| 7 | 13.563 | 0.7303 | 2.4413 | 0.8285 |
| 8 | 14.958 | 0.7812 | 2.0647 | 1.1541 |
| 9 | 15.256 | 0.4803 | 2.5254 | 0.9944 |
| 10 | 16.355 | 0.8343 | 2.3142 | 0.8515 |

File: "B0004W400.LRD"

OUTPUT OF TRAMP

14-NOV-85 20:01:14

```
|-------------------- OXBASH (Aug 1985) --------------------|
|------- The Oxford-Buenos-Aires-MSU Shell-Model Code -------|
|                                                            |
| OXBASH82      : A.Etchegoyen, W.D.M.Rae and N.S.Godwin     |
|               : contributors: W.A.Richter and C.H.Zimmerman|
| MSU version   : B.A.Brown, W.E.Ormand and J.S.Winfield     |
|------------------------------------------------------------|
```

Version with up to 96 m-states and 12 j-states allowed

Macro versions of extended-integer subroutines are being used

B4004W CALCULATION

# of valence particles : 4

J : 2 (Projected from state with Jz = 2)
T : 0 (Projected from state with Tz = 0)
Parity : +VE

Model space (*.SPS) name : SD
M-scheme basis (*.BAS) name : B4004W

| | | SPE_NORM | ME_NORM |
|---|---|---|---|
| TBME (*.INT) name (1) | : W | 1.00000 | 0.96889 |

Eigenvector (*.EIG) name : B4004W
Ground state (*.GND) name : B0004W -40.492 = Eigenvalue of grou
Matrix dimension : 56

# of particles allowed in each j-level Single-particle energies for eac

| SHELL | SUB SHELL MIN | SUB SHELL MAX | MAJOR SHELL MIN | MAJOR SHELL MAX | Total SPE | Delta SPE from input | ------ SPE F 1 |
|---|---|---|---|---|---|---|---|
| 1D3/2 | 0 | 4 | 4 | 4 | 1.6466 | 0.0000 | 1.6466 |
| 1D5/2 | 0 | 4 | | | -3.9478 | 0.0000 | -3.9478 |
| 2S1/2 | 0 | 4 | | | -3.1635 | 0.0000 | -3.1635 |

B0004W CALCULATION

# of valence particles : 4

J : 0 (Projected from state with Jz = 0)
T : 0 (Projected from state with Tz = 0)
Parity : +VE

Model space (*.SPS) name : SD
M-scheme basis (*.BAS) name : B0004W

| | | SPE_NORM | ME_NORM |
|---|---|---|---|
| TBME (*.INT) name (1) | : W | 1.00000 | 0.96889 |

Eigenvector (*.EIG) name : B0004W
Ground state (*.GND) name : B0004W -40.492 = Eigenvalue of grour
Matrix dimension : 21

# of particles allowed in each j-level Single-particle energies for each

| SHELL | SUB SHELL MIN | SUB SHELL MAX | MAJOR SHELL MIN | MAJOR SHELL MAX | Total SPE | Delta SPE from input | ------ SPE F 1 |
|---|---|---|---|---|---|---|---|
| 1D3/2 | 0 | 4 | 4 | 4 | 1.6466 | 0.0000 | 1.6466 |
| 1D5/2 | 0 | 4 | | | -3.9478 | 0.0000 | -3.9478 |
| 2S1/2 | 0 | 4 | | | -3.1635 | 0.0000 | -3.1635 |

ONE- BODY TRANSITION DENSITIES

INITIAL STATE : VECTOR 1 OF B0004W 0.000 MEV
FINAL STATE : VECTOR 1 OF B4004W 1.776 MEV

| CREATED DESTROYED K, N, L,2*J | K, N, L,2*J | COUPLED TO (J,T) = ( 2, 0) | ( |
|---|---|---|---|
| 1, 1, 2, 3 | 1, 1, 2, 3 | 0.0946916 | |
| 1, 1, 2, 3 | 2, 1, 2, 5 | -0.1032379 | |
| 1, 1, 2, 3 | 3, 2, 0, 1 | -0.2177138 | |
| 2, 1, 2, 5 | 1, 1, 2, 3 | 0.0882286 | |
| 2, 1, 2, 5 | 2, 1, 2, 5 | 0.4010500 | |
| 2, 1, 2, 5 | 3, 2, 0, 1 | 0.4398625 | |
| 3, 2, 0, 1 | 1, 1, 2, 3 | 0.1532917 | |
| 3, 2, 0, 1 | 2, 1, 2, 5 | 0.3756700 | |

INITIAL STATE : VECTOR 1 OF B0004W 0.000 MEV
FINAL STATE : VECTOR 2 OF B4004W 7.316 MEV

| CREATED DESTROYED K, N, L,2*J | K, N, L,2*J | COUPLED TO (J,T) = ( 2, 0) | ( |
|---|---|---|---|
| 1, 1, 2, 3 | 1, 1, 2, 3 | 0.0587080 | |
| 1, 1, 2, 3 | 2, 1, 2, 5 | -0.0362836 | |
| 1, 1, 2, 3 | 3, 2, 0, 1 | 0.0887414 | |
| 2, 1, 2, 5 | 1, 1, 2, 3 | 0.0334508 | |
| 2, 1, 2, 5 | 2, 1, 2, 5 | 0.4354609 | |
| 2, 1, 2, 5 | 3, 2, 0, 1 | -0.2509779 | |
| 3, 2, 0, 1 | 1, 1, 2, 3 | -0.0667320 | |
| 3, 2, 0, 1 | 2, 1, 2, 5 | -0.1900750 | |

INITIAL STATE : VECTOR 1 OF B0004W 0.000 MEV
FINAL STATE : VECTOR 3 OF B4004W 10.439 MEV

| CREATED K, N, L,2*J | DESTROYED K, N, L,2*J | COUPLED TO (J,T) = ( 2, 0) | ( |
|---|---|---|---|
| 1, 1, 2, 3 | 1, 1, 2, 3 | -0.0558531 | |
| 1, 1, 2, 3 | 2, 1, 2, 5 | 0.2195020 | |
| 1, 1, 2, 3 | 3, 2, 0, 1 | -0.0319798 | |
| 2, 1, 2, 5 | 1, 1, 2, 3 | -0.2084631 | |
| 2, 1, 2, 5 | 2, 1, 2, 5 | 0.1000397 | |
| 2, 1, 2, 5 | 3, 2, 0, 1 | -0.0221069 | |
| 3, 2, 0, 1 | 1, 1, 2, 3 | 0.0677785 | |
| 3, 2, 0, 1 | 2, 1, 2, 5 | 0.1054163 | |

INITIAL STATE : VECTOR 1 OF B0004W 0.000 MEV
FINAL STATE : VECTOR 4 OF B4004W 10.736 MEV

| CREATED K, N, L,2*J | DESTROYED K, N, L,2*J | COUPLED TO (J,T) = ( 2, 0) | ( |
|---|---|---|---|
| 1, 1, 2, 3 | 1, 1, 2, 3 | -0.0432189 | |
| 1, 1, 2, 3 | 2, 1, 2, 5 | 0.1540138 | |
| 1, 1, 2, 3 | 3, 2, 0, 1 | 0.0052999 | |
| 2, 1, 2, 5 | 1, 1, 2, 3 | -0.1212498 | |
| 2, 1, 2, 5 | 2, 1, 2, 5 | 0.0500784 | |
| 2, 1, 2, 5 | 3, 2, 0, 1 | 0.1577849 | |
| 3, 2, 0, 1 | 1, 1, 2, 3 | -0.0639282 | |
| 3, 2, 0, 1 | 2, 1, 2, 5 | -0.1472211 | |

INITIAL STATE : VECTOR 1 OF B0004W 0.000 MEV
FINAL STATE : VECTOR 5 OF B4004W 12.955 MEV

| CREATED K, N, L,2*J | DESTROYED K, N, L,2*J | COUPLED TO (J,T) = ( 2, 0) | ( |
|---|---|---|---|
| 1, 1, 2, 3 | 1, 1, 2, 3 | -0.0242711 | |
| 1, 1, 2, 3 | 2, 1, 2, 5 | 0.0156775 | |
| 1, 1, 2, 3 | 3, 2, 0, 1 | -0.0217064 | |

| | | |
|---|---|---|
| 2, 1, 2, 5 | 1, 1, 2, 3 | 0.0042413 |
| 2, 1, 2, 5 | 2, 1, 2, 5 | -0.0772905 |
| 2, 1, 2, 5 | 3, 2, 0, 1 | 0.0598113 |
| 3, 2, 0, 1 | 1, 1, 2, 3 | 0.0202500 |
| 3, 2, 0, 1 | 2, 1, 2, 5 | 0.0287335 |

```
File: "B0004W400.OBD"

DT format, made from OXBASH binary output file: B0004W400.TRD
 ! Model-space name = SD
 ! Interaction name = W
 !
 !      ----------------------------------
 !      |  Ai <-> Af  =    20  <->  20   |
 !      |  Ji <-> Jf  =     0  <->   2   |
 !      |  Ti <-> Tf  =     0  <->   0   |
 !      |  Pi <-> Pf  =     +  <->   +   |
 !      |  #i <-> #f  =    21  <->  56   |
 !      ----------------------------------
 !
 ! One-body transition densities A(DJ,DT) =
 ! = <f||| [a+(k1)a(k2)]^(DJ,DT) |||i>/SQRT[(2DJ+1)(2DT+1)]
 ! For input to some DWBA codes
 ! Z = A(DJ,DT) * <Ti,Tiz,DT,DTz|Tf,Tfz>SQRT(2DT+1)/SQRT[(2Ji+1)(2Tf+1)]
 !     where <|> is the Clebsch-Gordan and DTz=Tf-Ti
 ! with Edmonds (de-Shalit Talmi) reduced matrix element convention
 !
 ! For n,l,j =   1.0  2.0  2.5  label  k =   4
 ! For n,l,j =   1.0  2.0  1.5  label  k =   5
 ! For n,l,j =   2.0  0.0  0.5  label  k =   6
 !
 ! Ji,   Jf,   Ti,   Tf,
 ! DJ,   Ni,   Nf,       Ef,       Ei,      Exi,      Exf,
 !
  0.0,  2.0,  0.0,  0.0,  0.0,  0.0,
  2.0,  1.0,  1.0,  -38.716,  -40.492,    0.000,    1.776,
 5, 5,   0.09469,   0.00000, ! k1,k2,A(DT=0),A(DT=1)
 5, 4,  -0.10324,   0.00000,
 5, 6,  -0.21771,   0.00000,
 4, 5,   0.08823,   0.00000,
 4, 4,   0.40105,   0.00000,
 4, 6,   0.43986,   0.00000,
 6, 5,   0.15329,   0.00000,
 6, 4,   0.37567,   0.00000,
 0,
  2.0,  1.0,  2.0,  -33.177,  -40.492,    0.000,    7.316,
 5, 5,   0.05871,   0.00000,
 5, 4,  -0.03628,   0.00000,
 5, 6,   0.08874,   0.00000,
 4, 5,   0.03345,   0.00000,
 4, 4,   0.43546,   0.00000,
 4, 6,  -0.25098,   0.00000,
 6, 5,  -0.06673,   0.00000,
 6, 4,  -0.19007,   0.00000,
 0,
  2.0,  1.0,  3.0,  -30.053,  -40.492,    0.000,   10.439,
 5, 5,  -0.05585,   0.00000,
 5, 4,   0.21950,   0.00000,
 5, 6,  -0.03198,   0.00000,
 4, 5,  -0.20846,   0.00000,
 4, 4,   0.10004,   0.00000,
 4, 6,  -0.02211,   0.00000,
 6, 5,   0.06778,   0.00000,
 6, 4,   0.10542,   0.00000,
 0,
  2.0,  1.0,  4.0,  -29.756,  -40.492,    0.000,   10.736,
```