

Oxbash for Windows PC

B. Alex Brown

*Department of Physics and Astronomy
and National Superconducting Cyclotron Laboratory,
Michigan State University,
East Lansing, Michigan 48824-1321, USA*

Abstract

This manual describes the 2004 version of Oxbash that can be used on any Windows PC without the use of any other software. For information on obtaining a copy contact the author at brown@nscl.msu.edu. Updates of this manual can be found at www.nscl.msu.edu/~brown.

Contents

1	Introduction	4
2	Setup and login	5
2.1	Setup for the first time	5
2.2	Logging into oxbash	5
3	Using the batch file system	6
3.1	Some useful Windows commands	6
3.2	Special commands (the *.bat files in oxbash-exe)	6
4	Basic input commands	7
4.1	Doing a shell-model calculation	7
5	Input libraries	7
5.1	Sample inputs for shell	8
6	Model space tuncations	13
7	Description of the programs	14
7.1	Sequence of programs for energies	14
7.2	Sequence of programs for overlaps and transitions	15
7.3	Other programs	16
8	File names	17
8.1	Files made by shell	17
8.2	Names for the wavefunction and overlap files	18
8.3	*.ext names for the wavefunction	19
8.4	*.ext names for the overlaps	20
9	Cpu times	21
10	Short history of Oxbash	22
11	Sample formats for the *.sps and *.int files	23
11.1	General information on the *.sps and *.int files	23
11.2	Sample SPS file in isospin formalism (p.sps)	24
11.3	Sample SPS file in proton/neutron formalism (ppn.sps)	25
11.4	Sample INT file in isospin formalism (ckpot.int)	27
11.5	Sample INT file in proton/neutron formalism (ckpotpn.int)	28
12	Sample inputs for Dens	29
12.1	Ground state densities	29
12.2	B(EL) and B(ML) values	29
12.3	B(GT) values	31
12.4	Nested inputs for Dens, the *.den files	31

13 Special subfolders	32
13.1 Beta decay	32

1 Introduction

Oxbash is set of codes for carrying out shell-model calculations with dimensions up to about 50,000 in the J-T scheme and about 2,000,000 in the M-scheme. Oxbash comes with a library of model spaces and interactions. Any paper that is written using the Oxbash code should contain a sentence of the type – “The calculations were carried out in the x model space with the y hamiltonian [1] using the code Oxbash [2], where [1] is the reference for the hamiltonian and [2] is the reference for Oxbash.” If the model space is truncated in any way that is different from what is used in the reference for the hamiltonian it must be discussed and justified in the text.

The complete listing of model spaces and hamiltonians is given in the text file label.dat in the sps folder. The folder sps contains some of the well established hamiltonians with references. The reference for the hamiltonians is usually given at the top of the *.int files. For most model spaces there are many hamiltonians (*.int files). In order to determine which one to use, start with those that have the most recent publications and read the discussion in these papers. For some purposes an older interaction may be preferable. A recent review for interactions in light nuclei is given in [3].

[2] B. A. Brown, A. Etchegoyen, N. S. Godwin, W. D. M. Rae, W. A. Richter, W. E. Ormand, E. K. Warburton, J. S. Winfield, L. Zhao and C. H. Zimmerman, MSU-NSCL report number 1289.

[3] The Nuclear Shell Model Towards the Drip Lines, B. A. Brown, Progress in Particle and Nuclear Physics 47, 517 (2001).

2 Setup and login

2.1 Setup for the first time

Put the oxbash folder on the PC. It should be a folder of the type

```
c:\.....\oxbash
```

where c: the drive. The number of characters between c: and oxbash should not be more than 20. The oxbash folder is about 50 MB in size. I suggest to make a backup copy in a folder like oxbash-backup in case of accidents.

In Windows go the directory

```
...\oxbash\login
```

Edit the file login.bat in notepad. The first two lines of this file should be changed to correspond to the current drive and oxbash folder names.

Edit the file oxbash-dir.dat in notepad. The folder names in this file should be changed to current oxbash folder names.

Move the Oxbash-command-prompt shortcut to the desktop. Right click on the shortcut and then click on “properties.” Under the shortcut-tab in the line “start in” put the name of the current oxbash-login folder on your computer. (You may also want to change the font, layout and colors options for the default screen.)

Left click on the command-prompt setup as above. You will be in the oxbash-login folder.

This part only needs to be done once.

2.2 Logging into oxbash

The following must be done every time the Oxbash-command-prompt window is opened.

Left click on the command-prompt (setup as above). You will be in the oxbash-login folder.

Type **login**. This initializes the path so that the exe and bat files in the oxbash-exe folder can be used. Some special *.bat files in this oxbash-exe folder are listed below. During login a file called oxbash-dir.dat is copied to the root directory of the disk (for example C:). Thus calculations can only be carried on this disk.

3 Using the batch file system

3.1 Some useful Windows commands

dir – make a listing of all files in the current folder

cd x – go to folder x, where x specifies a subfolder or the full directory structure.

cd .. - go to the parent folder

md x – make a subfolder with the name x

copy x y – will copy files from x to y, e.g.

copy c:\aaa\oxbash\sps\sd.sps c:\aaa\oxbash\rsh\sd\sd.sps

3.2 Special commands (the *.bat files in oxbash-exe)

sd x – if x is one of the following, the command prompt will move directly to one of the oxbash folders

login - login files

help - help files

exe - exe files

sps - well documented *.sps and *.int files and the label.dat file

spsn - complete list of *.sps and *.int (many undocumented)

rsh - the folder suggested for the sub-folders for running oxbash

If x is not in this list it is the same as **cd x**.

shell – runs the oxbash code for making the *.bat files for calculations

ed x – is the same as **notepad x** where x is the file name.

(Note – do not use **edit**. By Windows default this brings up a screen for an old and hard to use editor)

di x – makes a listing of all files in the current folder with extension x. If x is left out then all files are listed.

co x y - copy files y from the folder oxbash-rsh-x to the current folder.

4 Basic input commands

4.1 Doing a shell-model calculation

After logging into oxbash type the following (replace x and y with your own names)

```
sd rsh    - goes to the oxbash-rsh folder
md x      - make the subfolder x (if it is not already made)
sd x      - goes to the subfolder x
shell     - answer the oxbash questions - call the batch file "y"
y         - runs the batch file made by shell
```

Once the subfolder is made, one can go to this folder and use it for any number of batch files. Shell makes unique names for all of the files (see below).

5 Input libraries

The folders sps and spsn contain previously made input files for the model space (*.sps) and hamiltonians (*.int). The file label.dat in the folder spsn contains a list of available model space and hamiltonian combinations. The file label.dat is used to make names for the output files.

When shell is run with a given model space and interaction it looks for the input files (label.dat, *.sps and *.int) in the following folders

- 1) first in the current folder
- 2) next in the spsn folder
- 3) next in the sps folder.

When the *.int files are modified it is best to copy them from spsn or sps folder to the current folder.

5.1 Sample inputs for shell

Answers to shell for the ^{20}Ne $J=0^+$ to $J=8^+$, $T=0$ wavefunctions in the sd shell.

```
ne20    - the batch file name
lpe     - the option to calculate wave functions (a)
sd      - model space name (b)
4       - number of valence particles
<blank> - no restrictions (yes/no)
w       - interaction name
0.,8.   - min,max J (this must be in floating-point format)
0.      - min,max T (do not need to enter max value if max=min)
<blank> - bad JT (c) (<blank> means no bad JT)
0       - parity
st      - stop
```

Notes:

<blank> can be used in place of **n** (no), **0.**, **0** or **st**.

The energies can be compared against those given at

<http://www.nsl.msu.edu/~brown/resources/SDE.HTM>

a) The default value for the number of wavefunctions is 10. If you want a larger number type lpe,N where N is the number you want.

b) The default label for the model space symbol is in the file label.dat. If you want to use another symbol type (in this example) sd,a where “a” will be the new symbol.

c) The calculation will automatically run for the entire list of J-T values given above. If you want to eliminate one or more from this list type the J-T value in this line with <blank>, type the next bad J-T in the next line etc. When the list done type <blank>.

Answers to shell for the ^{20}Ne $J=0^+$ and $J=2^+$, $T=0$ wavefunctions in the sd shell space followed by a calculation for the 0^+ to 2^+ and 2^+ to 2^+ one-body transition density for M1 and E2.

```

ne20t    - the batch file name
lpe      - the option to calculate wave functions
sd       - model space name
4        - number of valence particles
<blank>  - no restrictions (yes/no)
w        - interaction name
0.,2.    - min,max J (this must be in floating-point format)
0.       - min,max T (do not need to enter max value if max=min)
1.,0.    - bad J-T value (do not calculate 1+)
<blank>  - end list of bad J-T values
0        - parity
----- (a)
den      - the option to calculate overlaps between two wavefunctions
t        - the one-body transition density overlap option
bxx04w   - the name of the initial state (b)
1,1      - min,max number of initial states
byy04w   - the name of the final state
-1       - min,max number of final states (-1 means all final states)
0.,2.    - min,max J for initial state
0.       - min,max T for initial state
1.,0.    - bad J-T value (do not calculate 1+)
<blank>  - end list of bad J-T values
2.       - min,max J for final state
0.       - min,max T for final state
<blank>  - bad JT (none in this case)
y        - restrict the tensor ranks for the one-body operator (c)
1.,2.    - min, max spacial tensor for M1 and E2
0.,1.    - min, max isotpin tensor (for any one-body operator)
st       - stop

```

Notes:

a) If the wavefunctions have been calculated previously you may start here to make a batch file for the transition density. (It is recommended to give the wavefunction and one-body transition batch files difference names).

b) In the names bxx04w and byy04w the xx and yy are dummy letters that will be filled in after the min,max questions are answered.

c) If you want all possible tensor ranks type <blank> here and the next two lines will be skipped.

Answers to shell for the spectroscopic factor between ^{19}F $J=1/2^+$ $T=1/2$ and ^{20}Ne $J=0^+$ $T=0$ in the sd shell. The dashed lines separate different part of the input.

```
ne20s    - the batch file name
-----
lpe      - the option to calculate wave functions
sd       - model space name
4        - number of valence particles
<blank> - no restrictions (yes/no)
w        - interaction name
0.       - min,max J
0.       - min,max T
<blank> - bad J-T values
0        - parity
-----
lpe      - the option to calculate wave functions
sd       - model space name
3        - number of valence particles
<blank> - no restrictions (yes/no)
w        - interaction name
0.5      - min,max J
0.5      - min,max T
<blank> - bad J-T values
0        - parity
-----
den      - the option to calculate overlaps between two wavefunctions
1        - the spectroscopic factor overlap option
bxx03w   - the name of the initial (core) state
1,1      - min,max number of initial states
byy04w   - the name of the final (overlap) state
-1       - min,max number of final states (-1 means all final states)
0.5      - min,max J for initial state
0.5      - min,max T for initial state
<blank> - bad J-T (none in this case)
0.       - min,max J for final state
0.       - min,max T for final state
<blank> - bad J-T (none in this case)
<blank> - do not restrict the tensor rank
st       - stop
```

Answers to shell for the ^{20}Ne $J=0^+$ to $J=8^+$, $T_z=0$ wavefunctions in the sdpn shell.

ne20pn - the batch file name
lpe - the option to calculate wave functions
sdpn - model space name (b)
2 - number of valence protons
4 - number of valence particles (protons+neutrons)
<blank> - no restrictions (yes/no)
wpm - interaction name
0.,8. - min,max J (this must be in floating-point format)
<blank> - bad JT (c) (<blank> means no bad JT)
0 - parity
st - stop

Answers to shell for the spectroscopic factor between ^{19}F $J=1/2^+$ $T=1/2$ and ^{20}Ne $J=0^+$ $T_z=0$ in the sdpn shell. The dashed lines separate different part of the input.

```

ne20pns - the batch file name
-----
lpe      - the option to calculate wave functions
sdpn     - model space name
2        - number of valence protons
4        - number of valence particles (protons+neutrons)
<blank> - no restrictions (yes/no)
wpn      - interaction name
0.       - min,max J
<blank> - bad J-T values
0        - parity
-----
lpe      - the option to calculate wave functions
sdpn     - model space name
1        - number of valence protons
3        - number of valence particles (protons+neutrons)
<blank> - no restrictions (yes/no)
wpn      - interaction name
0.5     - min,max J
<blank> - bad J-T values
0        - parity
-----
den      - the option to calculate overlaps between two wavefunctions
1        - the spectroscopic factor overlap option
bxx03u  - the name of the initial (core) state
1,1     - min,max number of initial states
byy04u  - the name of the final (overlap) state
-1      - min,max number of final states (-1 means all final states)
0.5     - min,max J for initial state
0.5     - min,max T for initial state (a)
<blank> - bad J-T (none in this case)
0.       - min,max J for final state
0.       - min,max T for final state
<blank> - bad J-T (none in this case)
<blank> - restrict the tensor rank
st      - stop

```

a) In this case the input is for the $|T_z|$ value as given in the output to the answers to “lpe”. A negative value of T_z is indicated by increasing the parity label by two units (for example, b1123u, would be the name for ^{19}Ne with $J = 1/2$, $T_z = -1/2$ and parity = 0).

6 Model space truncations

In the examples given in the previous section the answer to the question for “any restrictions” was “n” (or <blank>) for “no”. If you want to put restrictions on partitions that are allowed answer “y” for “yes”. Shell will then ask for the type of restriction you want from the list of options

- (m) major-shell restrictions
- (s) subshell restrictions
- (h) restrictions on the values of $\hbar\omega$
- (l) use a file (*.par) that contains a list of allowed partitions
- (ms) both (m) and (s)
- (hs) both (h) and (s).

The orbitals in the sps file can be put into groups called major shells. The grouping is indicated in the first line of the *.sps file. In isospin formalism when “m” is chosen, shell will ask for the minimum and maximum number of nucleons you want in each of the major shells. In proton-neutron formalism these major shells are the groups of protons and neutrons orbitals and no further major shell divisions are possible in proton-neutron formalism.

When “s” is chosen, shell will ask for the minimum and maximum number of nucleons for each orbital. This works in both isospin and proton-neutron formalism. In proton-neutron formalism shell will all ask whether you want the restrictions to apply to the total number of proton and neutrons in a given orbital, and/or to protons and neutrons separately.

When the model space consists of more than one major oscillator group of orbits – for example, psd for 0p-1s0d, the partitions can be classified by their value of $\hbar\omega$ relative to the the lowest possible oscillator state. For a given A value the lowest oscillator energy called $0\hbar\omega$ will correspond to a given number of nucleons in each of the oscillator groups. For example, for $A = 14$ in the psd model space the lowest oscillator energy is (10,0) for 10 nucleons in the p-shell and zero nucleons in the sd shell (these states have positive parity). For this example, the $1\hbar\omega$ states (that have negative parity) will correspond to the the higher-energy oscillator distribution with (9,1). When “h” is chosen, shell will ask for a list of $\hbar\omega$ values to be used for the truncation. For example, for $A = 14$ the list “0,2” will allow for the positive parity states with the distributions (10,0) and (8,2) between the (p,sd) shells. The dimension for a given $N\hbar\omega$ may be zero. For example for $A = 14$ in the psd model space the only values of T allowed for $0\hbar\omega$ are 0 and 1. The lowest oscillator energy for $T = 2$ (corresponding to negative parity states in ^{14}B) will have $1\hbar\omega$.

When “l” is chosen for the restrictions, shell will ask for the name of the *.par file in which one list of the specific partitions to be considered. Typically one runs shell with no restrictions. The *.paa file that comes out of BASIS is contains the complete list of partitions. Renamed the *.paa file to *.par and then modify the list by hand or with a program.

7 Description of the programs

7.1 Sequence of programs for energies

SHELL makes a batch file **y.bat** that coordinates the program sequence and their inputs. The input is the answers to a series of questions. Alternatively, the input is from a **y.ans** file that has been made by a previous run of shell or by hand.

BASIS makes a list of all possible M-scheme basis states for a given model space together with a given set of restrictions. The restrictions are based on minimum and maximum occupancies of orbitals or major shell, by the possible $\hbar\omega$ values, or by an explicit list of partitions given in a *.par file.

PROJ makes linear combinations of the M-scheme basis states that have good J values in p/n formalism or good J and T in isospin formalism. The number of linear combinations is the J (or JT) dimension. The linear combinations are for states within a given partition. The loop over partitions is ordered by the J (or JT) dimension starting with the largest. Information on the projection is given in the *.per file.

PROJQ When the M-scheme dimension is very large for a single partition or when the partition involves a high-j orbit, the precision of the projection may be lost and the calculation fails. The failure is indicated by eigenvalues for the lowest-spin states that are very different from those of the higher-spin states. The failure is corroborated by warning messages in the *.per file. The failure might be corrected by using doing the calculation the higher precision version of proj called projq. Projq takes about 7 times longer to run than proj. This is done by making a file called “proj.dat” with 1 in the first line and then running shell again. For some model spaces such as “fp” and “fppn” projq is the default. To use the lower precision (but faster) version proj put 0 in the first line of the “proj.dat” file.

MATRIX makes the matrix corresponding to the J (or JT) dimension of the problem. Only the non-zero matrix elements are stored.

LANCZOS find the lowest N eigenvalues for the matrix. The default value of N is 10. The value of N can be changed by changing the first 0 in the first line of *.ans file to the desired value and then running shell. The output of Lanczos is are the eigenvalues and the eigenvectors in the projected basis.

7.2 Sequence of programs for overlaps and transitions

SHELL makes a batch file **y.bat** that coordinates the program sequence and their inputs. The input is the answers to a series of questions. Alternatively, the input is from a **y.ans** file that has been made by a previous run of shell or by hand.

MVEC reads the output of Proj and Lanczos to make the eigenvectors in the M-scheme basis. This is used as the input to Tramp.

TRAMP calculates overlaps between two wavefunctions. It can be used with the “den” option within shell to obtain:

T - one-body transition densities
1 - one-nucleon spectroscopic factors
2 - two-nucleon spectroscopic factors
D - direct overlaps
C - cluster overlaps
IS - isoscaler two-body transition densities
IV - isovector two-body transition densities
IT - isotensor two-body transition densities

TBTDOP makes an intermediate file needed for two-body transition densities.

7.3 Other programs

LEVEL provides a level scheme for a given set of J values. The input for the ^{20}Ne example is:

```
B***4W - generic name for the wavefunctions
0.,2. - min, max J value
0. - min, max T value
0 - min, max parity (0,1 for both positive and negative)
```

PN converts a given set of *.sps and *.int files in isospin formalism into their equivalent in proton/neutron formalism.

V generates a set of JT-coupled two-body matrix elements for a wide variety of two-body potential models.

DENS calculates the radial wavefunctions for a given nucleus with oscillator, Woods-Saxon or Skyrme Hartree-Fock potentials and reads the *.obd from oxbash to calculate B(EL), B(ML) and B(GT) values. Sample inputs for dens are given in a separate section.

PRA calculates values for Clebsch-Gordan, 3j, 6j and 9j coefficients.

8 File names

8.1 Files made by shell

y.bat – a batch files recognized by Windows.

y.inp – the data file corresponding to the order of the programs in y.bat. (This is a copy of the ne20.com file produced in the vms version of Oxbash.)

y.ans – a copy of the answers to the shell questions. If the *.ans file already exists it can be modified by notepad, and then used to answer the shell questions as follows.

shell

y.ans

For a series of calculations it may be efficient to generate the *.ans file (or files) with another program.

y.mit – a list of the hamiltonians and their normalizations. In some cases the hamiltonian may be broken into two or more parts. To change the details of the hamiltonian input copy the y.mit file to a file called “input.mit”, then edit this file. The values in “input.mit” will override the default values from shell.

8.2 Names for the wavefunction and overlap files

The first part of the file name for wavefunctions has the six letter form **AJTPNB** where:

A - symbol for the model space found in label.dat
 J - symbol for two times the spin from the list below
 T - symbol for two times the isospin from the list below
 P - symbol for the parity (0 for + and 1 for -)
 N - symbol for the number of valence particle from the list below
 B - symbol for the interaction found in label.dat

The first part of the names for overlaps has the nine letter form **AJTPNBjtp**, where **AJTPNB** is the name of the initial state and **jtp** is part of the name of the final state. The full name of the final state is implicitly given by the type of overlap.

In proton/neutron formalism the third letter is $|T_z|$. The sign of T_z is indicated by the parity - (0,1) for positive T_z and (2,3) for negative T_z .

If the model space and interaction name do not exist in label.dat they will be assigned the names **X** and **Y**, respectively.

The symbols for J, T and N correspond to the following numbers

0 - 0	K - 20	F - 40	0 - 60	K - 80	F - 100	0 - 120
1 - 1	L - 21	G - 41	1 - 61	L - 81	G - 101	1 - 121
2 - 2	M - 22	H - 42	2 - 62	M - 82	H - 102	2 - 122
3 - 3	N - 23	I - 43	3 - 63	N - 83	I - 103	3 - 123
4 - 4	O - 24	J - 44	4 - 64	O - 84	J - 104	4 - 124
5 - 5	P - 25	K - 45	5 - 65	P - 85	K - 105	5 - 125
6 - 6	Q - 26	L - 46	6 - 66	Q - 86	L - 106	6 - 126
7 - 7	R - 27	M - 47	7 - 67	R - 87	M - 107	7 - 127
8 - 8	S - 28	N - 48	8 - 68	S - 88	N - 108	8 - 128
9 - 9	T - 29	O - 49	9 - 69	T - 89	O - 109	9 - 129
A - 10	U - 30	P - 50	A - 70	U - 90	P - 110	A - 130
B - 11	V - 31	Q - 51	B - 71	V - 91	Q - 111	
C - 12	W - 32	R - 52	C - 72	W - 92	R - 112	
D - 13	X - 33	S - 53	D - 73	X - 93	S - 113	
E - 14	Y - 34	T - 54	E - 74	Y - 94	T - 114	
F - 15	A - 35	U - 55	F - 75	A - 95	U - 115	
G - 16	B - 36	V - 56	G - 76	B - 96	V - 116	
H - 17	C - 37	W - 57	H - 77	C - 97	W - 117	
I - 18	D - 38	X - 58	I - 78	D - 98	X - 118	
J - 19	E - 39	Y - 59	J - 79	E - 99	Y - 119	

8.3 *.ext names for the wavefunction

action = d - deleted by "cleanup"

action = * - most important files to keep

ex	action		program	
		form		
com		text	shell	vms command file (not used in Windows)
inp		text	shell	copy of com file used for Windows input
ans		text	shell	answer to the shell questions
bat		text	shell	Windows batch file
bas	d	binary	basis	m-scheme basis
inf	d	binary	basis	info on the calculation
lpb		text	basis	info on the partitions and dimensions
paa	d	text	basis	list of partitions (for *.par file)
*prj.bin	d	binary	proj	JT projection coefficient
red	d	binary	proj	JT projection overlaps
per		text	proj	info on possible errors
op		binary	oper	interaction in the m-scheme
*ma1.bin	d	binary	matrix	odd rows of the non-zero matrix
*ma2.bin	d	binary	matrix	even rows of the non-zero matrix
mah	d	binary	matrix	info on the calculation
mad	d	text	matrix	diagonal matrix elements
eig	d	binary	lanczos	wavefunctions in projected basis
lpe	*	text	lanczos	info on wavefunctions
lpi		text	lanczos	info in the top of eig file (for level)
gnd	d	text	lanczos	energy of the first JT state
lpt	*	text	level	collected energies from lpi files

8.4 *.ext names for the overlaps

vec	d	binary	mvec	wavefunction in the m-scheme basis
veh	d	binary	mvec	info on the calculation
trd	d	binary	tramp	tramp overlaps
lrd	d	text	tramp	info on the wavefunctions in tramp
tna	*	text	tramp	two-nucleon overlaps
lsf	*	text	tramp	one-nucleon overlaps
lsa	*	text	tramp	one-nucleon overlaps amplitudes
csa	*	text	tramp	cluster overlaps
loc	*	text	tramp	direct overlaps
lol	d	text	tramp	direct overlaps with all info
tol	d	binary	tramp	direct overlaps with all info
obd	*	text	rtrd	one-body transition densities from trd

9 Cpu times

The cpu time for oxbash depends on a combination of the M-scheme and JT-scheme dimensions. Most of the time is divided between proj, matrix and lanczos. The times below are for the lowest 10 eigenstates.

Windows XP with the pentium-4 2.8GH 512MB RAM.

model space	N	restriction	J,T	M-dim	JT-dim	----- time (s) -----			
						proj	matrix	lanczos	total
sd	12	none	2,0	90,736	3,276	26	15	6	48
sd	12	none	2,1	60,170	5,768	43	76	19	139
spsdpf	16	0,2,4hw	0,0	340,557	4,255	30	34	11	75
						----- time (m) -----			
spsdpf	16	0,2,4hw	2,0	279,051	15,894	4	11	14	29
N6	8	0,2,4,6hw	0,0	2,200,818	26,032	22	35	39	96
fppn	8	none*	0	1,963,461	41,355	570	30	270	870

Windows 2000 with the pentium-4 2.8GH 2GB RAM.

model space	N	restriction	J,T	M-dim	JT-dim	----- time (s) -----			
						proj	matrix	lanczos	total
sd	12	none	2,0	90,736	3,276	22	12	5	39
sd	12	none	2,1	60,170	5,768	36	59	18	113
spsdpf	16	0,2,4hw	0,0	340,557	4,255	26	28	10	64
						----- time (m) -----			
spsdpf	16	0,2,4hw	2,0	279,051	15,894	1.5	5.3	2.2	9
N6	8	0,2,4,6hw	0,0	2,200,818	26,032	3.5	28	1.8	33
fppn	8	none*	0	1,963,461	41,355	480	18	7	505

* In this example (^{48}Cr , $J=0$, $T=0$) proj and projq fails to give an accurate wave function in isospin formalism. The results here are for the high-precision version of proj (projq) in proton/neutron formalism.

10 Short history of Oxbash

- 1976 Conception (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,
(A. Etchegoyen, B. A. Brown, W. A. Richter,
N. S. Godwin and J. S. Winfield).
- 1982 General overall upgrade, multiple-integer words used
for partitions (A. Etchegoyen and W. D. M. Rae).
- 1983 BASIS speeded up (J. S. Winfield). Documentation and
general organization added (B. A. Brown).
- 1986 Upgrade to standard FORTRAN and preparation for
use on the FPS-164 array processor (W. E. Ormand).
(As of Nov 1988 the FPS versions are no longer being
used.)
- 1988 Changes made to make OXBASH CRAY compatible (L. Zhao).
General two-body interaction code added (B. A. Brown).
- 1994 Unix version and anonymous FTP (M. Horoi).
- 2004 Windows PC version (B. A. Brown).

11 Sample formats for the *.sps and *.int files

11.1 General information on the *.sps and *.int files

The basic input files for oxbash are *.sps and *.int. The *.sps file specifies the m-scheme single partilce states. The *.int file specifies the hamiltonian. Sample files formats are given below.

The *.int file contains the single-particle energies (SPE) and the J-coupled two-body matrix elements (TBME). The program **oper** converts the J-coupled two body matrix elements into M-scheme basis. For a given set of TBME oper only needs to be run one time to create a *.op file. Thus when shell runs it looks for the *.op file. If the *.op file does not exist it runs oper as part of the batch job. If the *.op file does exist it does not run oper. Thus anytime any of the TBME are changed in the *.int file, the *.op file must be deleted and oper must be run again:

```
oper - runs oper.exe from the exe folder
      - leave blank for no extra output
sps  - the sps file name
int  - the int file name
```

The SPE are always read from the *.int file. Thus, if only the SPE are changed in the *.int file the *.op file does not have to be deleted.

11.2 Sample SPS file in isospin formalism (p.sps)

```
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, 2j, 2j_z, 2t_z values.

The “dummy” input is not used by any program.

11.3 Sample SPS file in proton/neutron formalism (ppn.sps)

```

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

```

first line

12 = number of m states

4 = number of j orbitals

2 = number of m states in first orbit (p1/2 protons)

4 = number of m states in second orbit (p3/2 protons)

2 = number of m states in third orbit (p1/2 neutrons)

4 = number of m states in fourth orbit (p3/2 neutrons)

2 = number of major shells (always 2 in p/n)

2 = number of j orbits in proton shell

2 = number of j orbits in neutron shell

4 = A of core

2 = Z of core

second line - J-orbits names in A9 format

following lines - dummy and n, l, 2j, 2j_z, 2t_z values.

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. 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 fixed.

11.4 Sample INT file in isospin formalism (ckpot.int)

```
15          2.420, sd 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,tbme where I is the orbit index number. The input is always read in “free” format. Tbme is the value of the two

11.5 Sample INT file in proton/neutron formalism (ckpotpn.int)

```
! 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
```

see ckpotpn.int for the rest of this input

The proton-proton and neutron-neutron matrix elements are just those for $T=1$ in the isospin formalism. **The proton-neutron matrix elements are not in the standard formalism, but must be converted from the isospin $T=0$ and $T=1$ matrix elements as in this example – compare with the ckpot.int matrix elements.**

12 Sample inputs for Dens

12.1 Ground state densities

To obtain the ground state density for ^{208}Pb with the SKX Skyrme hamiltonian:

```
dens
az      - to input A and Z values
208,82  - the A and Z values
cp      - to setup the potential model (change potential)
SK20    - the SKX Skyrme interaction
gd      - do the calculation
st      - stop
```

Capital letters will always work and lower case letters will usually work (in this example capital letters are needed for SK20).

To find all of the potential model available type H after CP. In this list you will see that SK20 corresponds to SKX.

12.2 B(EL) and B(ML) values

To calculate the B(M1) in ^{20}Ne between the first 2^+ state and the second 2^+ state

```
dens
az      - to input A and Z values
20,10   - the A and Z values
mh      - start the calculation for oscillator B value
M1,10   - 10 indicates the how the obd will be read
2.,2.   - J_i, J_f, T_i, T_f (if different from the defaults)
1,2,1,1,16,b4004w400 - n_i, n_f, p_i, p_f, A_c, name of OBD file
st      - stop
```

The n_i and the n_f is the state number, $-n$ will loop from 1, to n. When $n=-999$ the loop if from 1 to all states in the obd file. p_i and p_f are the parities, 1 for + and -1 for -. A_c is the mass of core for the model space (^{16}O in this case).

To calculate the B(E2) in ^{20}Ne between the first 0^+ state (ground state) and the first 2^+ using the default oscillator value of $\hbar\omega = 45A^{-1/3}-25A^{-2/3}$:

```

dens
az
20,10
mh      - start the calculation for oscillator B value
E2,10  - 10 indicates the how the obd will be read
0.,2.  - J_i, J_f, T_i, T_f (if different from the defaults)
1,1,1,1,16,b0004w400 - n_i, n_f, p_i, p_f, A_c, name of OBD file
st      - stop

```

To use a b-value of 2.5 fm for the oscillator, do the following just before MH:

```

CP
H0,2.5

```

To use an $\hbar\omega$ -value of 14.0 MeV for the oscillator, do the following just before MH:

```

CP
H0,-14.0

```

To use another potential model such as SKX, do the following:

```

dens
az
20,10
cp
SK20
gd
td
E2,10
0.,2.  - J_i, J_f, T_i, T_f (if different from the defaults)
1,1,1,1,16,b0004w400
st

```

To calculate the electron scattering form factor for the ^{20}Ne transition with SKX:

```

DENS
az
20,10
cp
SK20
gd
ff      - start the form-factor calculation
E2,10
0.,2.
1,1,1,1,16,b0004w400
st

```

12.3 B(GT) values

To calculate B(GT) values from state J_i to all states with J_f

```
DENS
AZ
20,10,9      - the A, Z_i, Z_f values
GT
M1,10        - 10 indicates the how the obd will be read
2.,2.,0.,1. - J_i, J_f, T_i, T_f (if different from the defaults)
1,-999,1,1,16,B4004W420 - n_i, n_f, p_i, p_f, A_c, name of OBD file
ST
```

12.4 Nested inputs for Dens, the *.den files

Any of the inputs given in the example above can be made in terms of a *.den file. For example, the ^{20}Ne electron scattering calculation can be done as

```
dens
ne
ne20ff
st
```

where NE20FF refers to another file called ne20ff.den that contains

```
az
20,10
cp
SK20
gd
ff      - start the form-factor calculation
E2,10
0.,2.
1,1,1,1,16,b0004w400
RE
```

The *.den file can end with “re” that means return to the main input stream or “st” that ends the calculation. These type of inputs can be nested to any level. They are convenient for defining files that define aspects of the calculation like the potential models and the effective operator parameters.

13 Special subfolders

The Oxbash and Dens can be combined with other programs for special calculations.

13.1 Beta decay

The subfolder “beta” gives examples of how to use Oxbash and Dens together with the program beta.exe to calculate logft, half-life and branching ratios for Gamow-Teller beta decay. See the “help” file in the folder for details.