Nushell@MSU

B. A. Brown

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

W. D. M. Rae

Garsington, Oxford, UK

Contents

1 Introduction

This manual describes the version of Nushell that can be used on Windows PCs.

Nushell is set of programs for carrying out shell-model calculations with dimensions up to about 100,000 in the J-T scheme and about 2,000,000 in the M-scheme. Nushell@MSU comes with a library of model spaces and interactions. Any paper that is written using the Nushell@MSU 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 Nushell@MSU [2]”, where [1] is the reference for the hamiltonian and [2] is the reference for Nushell@MSU. If the model space, model-space truncation or Hamiltonian is changed from the original reference 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 several 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] Nushell@MSU, B. A. Brown and W. D. M. Rae, MSU-NSCL report (2007).

[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

Make the following directory for Nushell

```
c:\aaa\nushell
```

where c: is the drive. aaa can be replaced by another path with a path name of 20 characters or less and with no blank spaces. However, I recommend using aaa. The nushell folder is about 70 MB in size. Make a backup copy in a folder like nushell-backup in case of accidents.

In Windows go the directory

```
\aaa\nushell\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 nushell folder names.

Edit the file nushell-dir.dat in notepad. The folder names in this file should be changed to current nushell folder names. (When "shell" is run it puts a copy of nushell-dir.dat into the working directory.)

Move the Nushell-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 nushell-login folder on your computer. (You may also want to change the font, layout and colors options for the default screen.)

This part only needs to be done once.

2.2 Logging into nushell

Left click on the command-prompt (setup as above). This command goes to the nushell-login folder and executes the login.bat command. Login.bat initializes the path so that the exe and bat files in the nushell-exe folder can be used. Some special *.bat files in this nushell-exe folder are listed below.

3 The main Nushell folders

data: Tables of masses, rms radii, evaluated nuclear levels and input to the beta decay program (fgtw.dai and xfg.dai).

exe: The execution and batch files.

help: The help file.

login: The login file.

mass: The program to read the nuclear binding energy table.

pot: The program for Woods-Saxon bound states and scattering.

rsh: Sample nushell inputs and suggested place for nushell calculation subfolders.

sps: The *.sp and *.int file inputs that come with nushell.

spsn: The *.sp and *.int files supplied by the user (it starts out empty).

speplot: Programs and samples for the HF single-particle energy plots.

4 Using the batch file system

4.1 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\nushell\sps\sd.sp c:\aaa\nushell\rsh\sd\sd.sp
```

4.2 Special commands (the *.bat files in nushell-exe)

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

```
login - login files
help  - help files
exe   - exe files
sps   - well documented *.sp and *.int files and the label.dat file
spsn  - new inputs by the user
rsh   - the folder suggested for the sub-folders for running nushell
```

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

shell - runs the nushell 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 - makes a listing of all files in the current folder. You can use x*y.z to get specific types of files.

dis - same as **di** but files are listed in order of data made with the latest being last.

d - list of directories.

5 Basic input commands

5.1 Doing a shell-model calculation

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

```
sd rsh    - goes to the nushell-rsh folder
md x      - make the subfolder x (if it is not already made)
sd x      - goes to the subfolder x
shell     - answer the questions - call the batch file, for example, "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).

6 Input libraries

The folders sps contains previously made input files for the model space (*.sp) and hamiltonians (*.int). The file label.dat in the folder sps 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, *.sp 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 first copy them from spsn or sps folder to the current folder.

6.1 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.nscl.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. lpe,-1 will give all wavefunctions.

b) The default label for the model space name 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. In the model space and interaction are not in label.dat the "x" is used for the model space label and "y" is used for the interaction label.

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 calcualted 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 nest 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)
wpn       - 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 $= 0$).

7 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 sp file can be put into groups called major shells. The grouping is indicated in the first line of the *.sp 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 majors shell 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 restictions, 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.

8 Description of the programs

8.1 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.

NUBASIS 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.

NUPROJ 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.

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

NULANCZOS 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.

8.2 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.

8.3 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 *.sp and *.int files in isospin formalism into their equivalent in proton-neutron (pn) formalism (see the section on the conversion of isospin to 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 nushell to calculated 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.

9 Conversion of isospin to proton-neutron two-body matrix elements

The conversion of normalized isospin wavefunctions to normalzied proton-neutron (pn) wavefunctions is ($a \neq b$):

$$| \; aa, J >_{pp} = | \; aa, J = even, T = 1 >$$

$$| \; ab, J >_{pp} = | \; ab, J, T = 1 >$$

$$| \; aa, J = even >_{pn} = | \; aa, J = even, T = 1 >$$

$$| \; aa, J = odd >_{pn} = | \; aa, J = odd, T = 0 >$$

$$| \; ab, J >_{pn} = \sqrt{\frac{1}{2}} [| \; ab, J, T = 1 > + \; | \; ab, J, T = 0 >]$$

Thus the normalized proton-proton (and neutron-neutron) matrix elements are just the $T = 1$ matrix elements, and the normalized proton-neutron matrix elements are ($a \neq b$ and $c \neq d$):

$$< aa, J \mid V|cc, J >_{pn} = < aa, J, T = 1 \mid V|cc, J, T = 1 > + < aa, J, T = 0 \mid V|cc, J, T = 0 >$$

$$< ab, J \mid V|cc, J >_{pn} = \sqrt{\frac{1}{2}} [< ab, J, T = 1 \mid V|cc, J, T = 1 > + < ab, J, T = 0 \mid V|cc, J, T = 0 >]$$

$$< aa, J \mid V|cd, J >_{pn} = \sqrt{\frac{1}{2}} [< aa, J, T = 1 \mid V|cd, J, T = 1 > + < aa, J, T = 0 \mid V|cd, J, T = 0 >]$$

$$< ab, J \mid V|cd, J >_{pn} = \frac{1}{2} [< ab, J, T = 1 \mid V|cd, J, T = 1 > + < ab, J, T = 0 \mid V|cd, J, T = 0 >]$$

In the first three of these equations one of the matrix elements on the right-hand side is zero.

Nushell uses unnormalized proton-neutron matrix elements (upn) of the form

$$< aa, J \mid V|cc, J >_{upn} = [< aa, J, T = 1 \mid V|cc, J, T = 1 > + < aa, J, T = 0 \mid V|cc, J, T = 0 >]$$

$$< ab, J \mid V|cc, J >_{upn} = [< ab, J, T = 1 \mid V|cc, J, T = 1 > + < ab, J, T = 0 \mid V|cc, J, T = 0 >]$$

$$< aa, J \mid V|cd, J >_{upn} = [< aa, J, T = 1 \mid V|cd, J, T = 1 > + < aa, J, T = 0 \mid V|cd, J, T = 0 >]$$

$$< ab, J \mid V|cd, J >_{upn} = [< ab, J, T = 1 \mid V|cd, J, T = 1 > + < ab, J, T = 0 \mid V|cd, J, T = 0 >]$$

The program PN converts an isospin interaction file into an unnormalized proton-neutron interaction file. In the proton-neutron file there is a label T'. For the pp and nn matrix elements you must have $T' = 1$. For the proton-neutron matrix elements the T' label must be 0 or 1 but its value does not affect the calculation. T' can be used (as in the program PN) to keep track of where the proton-neutron matrix element came from in terms of the isospin matrix elements.

If you are given a set of normalized proton-neutron matrix elements they must be converted into unnormalized matrix elements to use in nushell (set $T' = 0$):

$$< aa, J \mid V|cc, J >_{upn} = < aa, J \mid V|cc, J >_{pn}$$

$$< ab, J \mid V|cc, J >_{upn} = \sqrt{2} \ < ab, J \mid V|cc, J >_{pn}$$

$$< aa, J \mid V|cd, J >_{upn} = \sqrt{2} \ < aa, J \mid V|cd, J >_{pn}$$

$$< ab, J \mid V|cd, J >_{upn} = 2 < ab, J \mid V|cd, J >_{pn}$$

10 File names

10.1 Files made by shell

y.bat - Windows batch file for program sequence

y.inp - the data file corresponding to the order of the programs in y.bat.

y.ans - a copy of the answers to the shell questions. If the *.ans file already exits 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.

10.2 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	

10.3 *.ext names for the wavefunction

```
action = d  - deleted by "cleanup"
action = *  - most important files to keep
```

ex	action			
		form	program	
ans		text	shell	answer to the shell questions
inp		text	shell	copy of com file used for Windows input
bat		text	shell	Windows batch file for program sequence
mit		text	shell	list of interactions to be added
nus		text	nupart	intermediate nushell input
nba	d	binary	nubasis	m-scheme basis
npt		text	nubasis	info on the partitions and dimensions
prj	d	binary	nuproj	projection coefficients
ort	d	binary	nuproj	projection output
jba	d	binary	nuproj	projection output
opn	d	binary	nuoper	interaction in the m-scheme
mtx	d	binary	numatrix	
spe		text	nuaddop	total spe from all interactions
eig	d	binary	nulanczos	wavefunctions in projected basis
lp	*	text	nulanczos	info on wavefunctions
lev	*	text	nulanczos	energies
set.bat		text	set parameter files	
lpt	*	text	nuleb	collected energies from lp files

10.4 *.ext names for the overlaps

mvc	d	binary	mvec	wavefunction in the m-scheme basis
trs	d	binary	tramp	tramp overlaps
trb		text	tramp	tramp overlaps
tna	*	text	nulsf	two-nucleon overlaps
lsf	*	text	nulsf	one-nucleon overlaps
obd	*	text	nulsf	one-body transition densities from trd

11 Short history of Nushell

1976 Oxbash 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).

2007 Nushell is a completely rewritten code in Fortran 95 by W. D. M. Rae. Nushell@MSU is the nushell core program with input and output formats in the Oxbash style.

12 Sample formats for the *.sp and *.int files

12.1 General information on the *.sp and *.int files

The basic input files for oxbash are *.sp and *.int. The *.sp file specifies the single particle 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
sp   - the sp  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.

12.2 Sample SP file in isospin formalism (p.sp)

```
t            t = isospin formalism
4 2          a and z of core
2            number of orbits
1 2          number of major shell and number of orbits in each
1 1 1 1      1st orbit n,l,2j
2 1 1 3
```

12.3 Sample SP file in proton/neutron formalism (ppn.sp)

```
pn           pn = proton-neutron formalism
4 2          a and z of core
4            number of orbits
2 2 2        2 major shell, number of proton orbtis, number of neutron orbits
1 1 1 1      1st orbit n,l,2j
2 1 1 3
3 1 1 1
4 1 1 3
```

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

```
 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,tbme where I is the orbit index number. The input is aways read in “free” format. Tbme is the value of the two body matrix element.

12.5 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.**

13 Sample inputs for Dens

13.1 Orbit labels for Dens

```
 k n l  j

 1 1 s 1/2 [1]
 2 1 p 3/2

 3 1 p 1/2 [2]
 4 1 d 5/2
 5 1 d 3/2

 6 2 s 1/2 [3]
 7 1 f 7/2
 8 1 f 5/2
 9 2 p 3/2
10 2 p 1/2 [4]

11 1 g 9/2
12 1 g 7/2
13 2 d 5/2
14 2 d 3/2
15 3 s 1/2 [5]

16 1 h 11/2
17 1 h 9/2
18 2 f 7/2
19 2 f 5/2
20 3 p 3/2
21 3 p 1/2 [6]

22 1 i 13/2
23 1 i 11/2
24 2 g 9/2
25 2 g 7/2
26 3 d 5/2
27 3 d 3/2
28 4 s 1/2 [7]

29 1 j 15/2[8]
```

13.2 Ground state densities

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

```
dens
fn       - change filenane for outoput (default=dens.dao)
pb208      filename
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 for the Ground state Density
st       - stop
```

To find all of the potential model available type h after cp. In this list you will see that sk20 corresponds to Skx.

13.3 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 beore MH:

```
CP
HO,2.5
```

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

```
CP
HO,-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
```

13.4 B(GT) values

To calculate B(GT) values from state Ji to all states with Jf

```
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
```

13.5 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.

14 Special subfolders

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

14.1 Beta decay

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

14.2 Gamma decay

Proceedure for generating a gamma decay spectrum for nucleus x

```
1) Run shell with lpe and den(t) to obtain the *.obd files for
   all transitions

2) Rename the gamma.den file to x.den

3) Rename the sample.tps file to x.tps

4) Edit the x.den file and change the top few lines as indicated

5) Edit the x.tps file and change the information as needed

6) Run dens by doing with the following input

dens
ne
x

7) Type

decay x

8) Data output is in x.deo. The file x.eps is a postscript file of the
decay scheme, but it may not readable if too many transitions are allowed.
```

14.3 Cluster overlaps

The subfolder “rsh.mg24” gives sample inputs for calculating the overlaps of two wavefunctions with a cluster. The example is for the overlaps of the ^{24}Mg and ^{28}Si ground states with SU3 four-particle cluster wavefunction in the sd shell. The procedure is as follows:

1) Run shell for the three inputs mg24.ans, si28.ans and su3.ans.

2) Calculate the wavefunctions by typing “mg24, si28 and su3”.

3) Edit the file csa.inp and put the file names of the wavefunctions b0004s, b0008w and b000cw (these are already entered in the sample file). At the end put a name of up to nine characters for the output file (in this sample file the name is mg24-si28).

4) The spectroscopic factor (overlap squared) is in the file mg24-si28.csa. More details of the initial and final states and the overlap amplitude are given in the file mg24-si28.lsa.