NuShellX

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, Oxfordshire, OX44, UK

Contents

1 Introduction

This manual describes the version of NuShellX that can be used on Windows or Linux (64 bit).

NuShellX is set of programs for carrying out shell-model calculations with dimensions up to about 10,000,000. The main code NuShellX was written by W. D. M. Rae and is documented in the file NuShellX.pdf. NuShellX@MSU contains wrapper codes for input and output similar in style to the Oxbash code, and comes with a library of model spaces and interactions. Any paper that is written using the NuShellX@MSU code should contain a sentence of the type - "The calculations were carried out in the x model space with the y Hamiltonian (give source reference for the interaction) using the code NuShellX [1]". If the model space, model-space truncation or Hamiltonian is changed from the original reference it must be discussed and justified in the text.

Note that all calculations with NuShellX are in the proton-neutron formalism. When the *.int files in isospin formalism are used, they are converted to proton-neutron formalism. Calculations in isospin formalism only can be carried out with the NuShell@MSU or Oxbash codes.

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. A review for interactions in light nuclei is given in [2].

[1] The Shell-Model Code NuShellX, B. A. Brown and W. D. M. Rae,
Nuclear Data Sheets 120, 115 (2014).

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

2 Installation

2.1 Windows setup

Unzip the folders to

```
c:\aaa\
```

so that the directory structure for nushellx is

```
c:\aaa\nushellx\windows\  (windows login, exe and bat files)
c:\aaa\nushellx\help\
c:\aaa\nushellx\sps\    (input files - same for windows and linux)
c:\aaa\nushellx\toi\    (table of isotopes data - same for windows and linux)
```

Go the nushellx-login folder

```
c:\aaa\nushellx\windows\login
```

Edit the file login.bat (if needed).

You may edit the file ed.bat to point to your text editor.

Copy the nushellx-login shortcut file 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 aaa-nushellx-login folder on your computer. You may also want to change the font, layout and color options for the default screen.

This part only needs to be done once.

2.2 Linux setup

First you must install gfortran. Then unzip the folders to

```
...aaa/
```

so that the directory structure for nushellx is

```
...aaa/nushellx/linux/bin/   (linux executables)
...aaa/nushellx/mac/bin-mac-intel/   (mac executables)
...aaa/nushellx/help/
...aaa/nushellx/sps/    (input files - same for windows and linux)
...aaa/nushellx/toi/    (table of isotopes data - same for windows and linux)
```

2.3 Logging into nushellx with Windows

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

2.4 Logging into nushellx with Linux

You will need a

`.bash_profile`

file that contains (example given by my path names):

```
nuxhome=/mnt/home/brownbo
export PATH=$nuxhome/aaa/nushellx/linux/nushellx-gfortran-bin:$PATH
export nushellx_sps=$nuxhome/aaa/nushellx/sps/
export mass_data=$nuxhome/aaa/nushellx/toi/mass-data/
export toi_data=$nuxhome/aaa/nushellx/toi/toi-data/
```

There is a sample bash file in the linux folder. The is a similar bash file for mac in the mac folder. To make linux (mac) commands alias names for some common windows commands the followings can be defined in the bash file:

```
alias copy=cp
alias del=rm
alias q='qstat -u brownbo'
alias qall='checknode -v ifi-003'
alias sd='cd'
alias di='ls -g -l -a'
alias dis='ls -tr -l -g -a'
alias d='ls -l | egrep ^d'
alias ed='vi'
alias md='mkdir'
```

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.

3.2 Special commands (the *.bat files in nushellx-login)

sd x (Windows only) - if x is one of the following, the command prompt will move directly to one of the nushellx folders

```
login - login files
help  - help files
sps   - *.mod and *.int files and the label.dat file
rsh   - c:\rsh-nushellx, the suggested scratch folder name
```

shell - runs the nushellx code for making the *.bat files for calculations

ed x - is **editor x** where x is the file name and **editor** is the name path specified in ed.bat in the login folder.

(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.

save - copy all of the files to be kept to the subfolder save using the list from save.dat in the sps folder.

4 Basic input commands

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

```
sd rsh   - goes to the rsh-nushellx folder (make the folder if it is not already made).
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, "test"
test     - 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 contains previously made input files for the model space (*.mod) 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, *.mod and *.int) in the following folders
1) first in the current folder
2) next in the sps folder.

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

6 The *.ans files

In the examples below, when the questions are answered they are recorded in the file x.ans, where x is the name for the batch file. You can then modify the x.ans file and rerun shell using x.ans for the name of the batch file. You can add many *.ans file together, delete the "st" line and then run shell to do many calculations in sequence. You can make the *.ans file with another computer program to loop over many cases.

7 Sample inputs for shell

Answers to shell for the ^{20}Ne J=0^+ to J=4^+, wavefunctions in the sd shell.

```
ne20                    the name for the batch file
wav                   ! option for wavefunctions, neig (0 or blank for 10) (a)
sd                    ! model space (*.mod) name (a8)                      (b)
n                     ! any restrictions (y/n)
usdb                  ! interaction (*.int or *.ham) name (a8)             (c)
 10                   ! number of protons
 20                   ! number of nucleons
 0.0, 4.0             ! min J, max J, del J                                (d)
  0  0                ! [parity (0,1,2) for (+,-,both)], [(0,1,2,3)]       (e)
st                    ! option (wav, den, or stop)
```

The energies for all levels are in the file ne20b.lpt. The occupation numbers for all levels are in the ne20b.occ file. The energies are compared to experiment in the ne20b.eps postscript picture. The full list of output files is given below.

a) The default number of converged wavefunctions is 10. For example, use wav,3 if you want 3 converged. Use wav,-1 if you want all states converged.

b) One of the model-space names listed in label.dat.

c) One of the interactions listed in label.dat. If the model space and/or interaction are not in label.dat the "x" is used for the model space label and "y" is used for the interaction label.

d) The default value for del J is 1.0

e) The second number controlas the overlap option. If the overlap option is 1, the wavefunctions will be calculated, then the overlap of t2.int is calculated, and the resulting isospin label is added to the *.lpt file. t2.int is the T^2 operator.

If the overlap option is 2, the wavefunctions will be calculated, then the overlaps of all scalar two-body operators listed in parts.nux will be calculated. The results for the overlaps are given in the aaaaaa.x files, where aaaaaa is the name of the wavefunction and x is the name of the two-body scalar operator.

If the overlap option is 3, the wavefunction will not be calculated and the overlaps of all scalar x.int files listed in parts.nux will be obtained with previously existing wavefunctions.

Answers to shell for the ^{19}F J=$1/2^+$ to J=$9/2^+$, wavefunctions in the sd shell.

```
f19                    the name for the batch file
wav                   ! option for wavefunctions, neig (0 or blank for 10)
sd                    ! model space (*.mod) name (a8)
n                     ! any restrictions (y/n)
usdb                  ! interaction (*.int or *.ham) name (a8)
  9                   ! number of protons
 19                   ! number of nucleons
 0.5, 4.5             ! min J, max J, del J
  0  0                ! [parity (0,1,2) for (+,-,both)], [(0,1,2,3)]
st                    ! option (wav, den, or stop)
```

Answers to shell for the spectroscopic factor between the first ^{19}F J=$1/2^+$ state and all ^{20}Ne J=0-4^+ states.

```
f19s                    name for batch file
den                   ! option (wav, den, or stop)
1                     ! 1, 2 or t                        (a)
f_19b                 ! initial file name                (b)
   1                  ! max number (-1 for all)
ne20b                 ! final file name
  -1                  ! max number (-1 for all)
 0.5                  ! min, max J, del J for bb1301
 0.0, 4.0             ! min, max J, del J for bb2400
n                     ! range of tensor ranks
st                    ! option (wav, den, or stop)
```

The outputs are in the *.lsf files.

a) 1 for one-nucleon transfer. and 2 for two-nucleon transfer.

b) This is the name of the *.lpt file. The initial nucleus must be the lighter one of the two.

Answers to shell for the gamma decay scheme for ^{20}Ne.

```
ne20g                 the name for the batch file
den                  ! option (wav, den, or stop)
t                    ! 1, 2 or t                       (a)
ne20b                ! initial file name               (b)
  -1                 ! max number (-1 for all)
ne20b                ! final file name
  -1                 ! max number (-1 for all)
 0.0, 4.0            ! min, max J, del J for bb2400    (c)
 0.0, 4.0            ! min, max J, del J for bb2400
n                    ! range of tensor ranks           (d)
st                   ! option (wav, den, or stop)
```

The one-body transition densities are in the *.obd files. The gamma decay scheme is in the *.deo file.

a) t for one-body transition density

b) This is the name of *.lpt file.

c) The default for del J is 1.0

d) The default is 0 to 2. If you want other values put "y" and in the next line enter the min,max values for the range.

8 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
(s) subshell restrictions
(m) restriction in selected two groups of orbitals
(ms) both (s) and (m)

9 The *.nux files

Files labeled *.nux are special control inputs.

parts.nux: listing of the x.int files for the overlaps. The name of x are listed in a column.

fastin.nux (before running shell): A MPI type batch for the den(t) option was implemented for windows in Feb 2017. Inside shell, ifast = 1 uses the new batch system, ifast = 0 uses the old sequential system. The defaults are ifast = 1 fow windows and ifast = 0 for linux. To override this default for windows make a file called fastin.nux and put 0 or 1 at the top.

maxj2.nux (before running shell): The default value for the maximum value for 2xJp and 2xJn in (JpxJn)J are the maximum allowed in the model space. For example for 21Ne in the sd-shell the 2xJpmax=8 and 2xJnmax=11. To decrease these values make a file called maxj2.nux and put the new values for 2xJpmax and 2xJnmax in the first line.

nimin.nux: (nutra.f90) The default value for the minimum (min) value of the initial state index for transition matrix element is one (min=1). (The max value is entered in shell). To change the value make a file called nimin.nux and put the value for min in the first line.

maxdj.nux (nupom.f90). The default value is the maximum possible for the a+(j1)a(j2) operator determined by the j1 and j2 values. To reduce this make a file called maxdj.nux and put the max value in the first line.

converge.nux (nucorex.f90, nulnz). Use this to change the converge criteria for the change in the eigenvalues from its default value of 0.0005 keV during the run. Make a file called converge.nux and put the new value in the first line.

iter.nux (nucorex.f90, nulnz). Use this to change the maximum number of iterations during the run. The default value depends on the case and is printed in the *.cpu file during the run. Make a file called iter.nux and put the new value in the first line.

10 Description of some other programs

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.

tbme 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 nushellx to calculated B(EL), B(ML) and B(GT) values. Sample inputs for dens are given in a separate section.

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

11 Phase conventions

Phase conventions enter into the one-body transition densities. They are defined by the calculation of the Hamiltonian matrix elements. The radial wavefunction is positive at near $r = 0$. $\mid j >= \mid [\ell \otimes s]^j >$ The wavefunctions are real and do not contain any $(\mathrm{i})^\ell$ factor. The Edmonds reduced matrix element convention is used.

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

The conversion of normalized isospin wavefunctions to normalized 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 $c \neq d$):

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

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

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

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

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

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

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

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

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

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

In the proton-neutron interaction (*.int) file there is a label T'. For the pp and nn matrix elements you must have $T' = 1$. If you are given a set of normalized proton-neutron matrix elements they must be converted into unnormalized matrix elements to use in nushell:

$$< aa, J, T' \mid V \mid cc, J, T' >_{upn} = < aa, J \mid V \mid cc, J >_{pn} \quad (J + T' = \text{odd})$$

$$< ab, J, T' \mid V \mid cc, J, T' >_{upn} = \sqrt{2} \; < ab, J \mid V \mid cc, J >_{pn} \quad (J + T' = \text{odd})$$

$$< aa, J, T' \mid V|cd, J, T' >_{upn} = \sqrt{2} < aa, J \mid V|cd, J >_{pn} \quad (J + T' = \text{odd})$$
$$< ab, J, T' \mid V|cd, J, T' >_{upn} = 2 < ab, J \mid V|cd, J >_{pn} \quad (T' = 0, 1)$$

When the upn matrix elements are obtained from a set of good isospin matrix elements both matrix elements (with $T = 0$ and $T = 1$) can be used with $T' = T$. **When the upn matrix element is obtained from normalized pn matrix elements one must have T=1 for pp and nn and J+T′=odd for pn except for the last line above where T′=0 can be used.**

13 File names

13.1 Files made by shell

y.bat - Windows batch file for program sequence

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.

13.2 Names for the wavefunction and overlap files

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

```
a - symbol for the model space found in label.dat
b - symbol  for the interaction found in label.dat
x - sympol for the number of valence protons
y - symbol for the number of valence nucleons
p - symbol for the parity (0 for + and 1 for -)
j - symbol for two times the spin from the list below
```

The first part of the names for overlaps has the nine letter form **ABXYPJ**$_x$**ypj** where **ABXYPJ** is the name of the initial state and **xypj** is part of the name of the final state.

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	h - 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	

13.3 *.ext names for the outputs

The "save" command will copy all of these files to a subdirectory called save. The files saved (the list below) are in the save.dat file in the sps folder.

```
sps is the sps data folder, x is the nushellx program run in the batch
           from     to
*.ab       x                  tri-diagaonal matrix
*.ans                 shell    shell input file with answers to questions
*.bat      shell     x        shell output with scripts for running x
*.bei      dens      beta     made with *.ben file for GT
*.ben      shell     dens     input for GT beta decay matrix elements
*.beq      shell     beta     beta input for GT beta decay
*.beo      beta               beta output for GT beta decay
*.bgt      dens               made with *.ben file
*.bm1      gamma              list of B(M1) values, Ef-Ei, B, sum B, Ji, Jf
*.be2      gamma              list of B(E2) values, Ef-Ei, B, sum B, Ji, Jf
*.cpu      x                  log of *.bat that contains cpu times
*.dei      dens      gamma    made with *.den file for M1 and E2
*.dai      sps       beta     xfg and fgtw input files for beta phase space
*.dat                         input control files for shell and x
*.den      shell     dens     input for M1 and E2 matrix elements
*.deo      gamma              gamma decay scheme for M1 and E2
*.det      gamma              gamma decay half-lives
*.dim      shell              dimensions
*.eps      levp               postscript figure of energy levels, B(M1) and B(E2)
*.pdf      cps                pdf version of eps files
*.fid      shell              list of files to make accumulated *.lsf with nulsf
*.fil      shell              list of files to make *.lpt with nulev
*.int      sps                Hamiltonian files
*.iso      nuxpt              Information on isospin and isospin mixing (if iso.nux is p
*.lev      nulev     levp     theory energies input to levp
*.lpe      x                  list of eigenfunction properties
*.lpt      nulev              accumulated level scheme based on *.fil input
*.ls       x                  Jp Jn decomposition of wf
*.lp       x                  partition decomposition of wf
*.amp      x                  amplitude decomposition of wf (made if amp.nux if present)
*.lsa      nulsf              spectroscopic amplitudes (with sign)
*.lsf      nulsf              spectroscopic factors
*.mit      shell              list of Hamiltonians to be added together
*.map      map                makes *.eps for B(M1) and B(E2)
*.npar     shell              list of neutron partitions
*.ppar     shell              list of proton partitions
*.nux                         special condition files
*.plt      dens               data to plot with top(cps)
```

*.obd	nulsf	one-body transition densities
*.occ	x	accumulated levels with occupation numbers
*.ov	shellx2	expectation values of two-body scalar operators
*.mod	sps	model space information
*.tna	nulsf	two-nucleon amplitudes
*.top	levp	input to make the *.eps figures

14 Sample formats for the *.mod and *.int files

14.1 General information on the *.mod and *.int files

The basic input files for nushellx are *.mod and *.int. The *.mod file specifies the single particle states. The *.int file specifies the Hamiltonian. Sample files formats are given below.

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.

14.2 Sample mod file

The mod files for nushellx must be in proton-neutron format.

```
p.mod file. If *.int files are used the orbital labels must be in the
same order as those listed in this file.

4 2               a,z of core
-1 0 1 1          tz,n,l,j
-1 0 1 3
1 0 1 1
1 0 1 3
```

14.3 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.

14.4 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.**

15 Orbit labels

k	n	l	j	magic numbers	group names
1	0	s	1/2	2	
2	0	p	3/2		
3	0	p	1/2	8	p
4	0	d	5/2	14	
5	0	d	3/2		sd
6	1	s	1/2	20	
7	0	f	7/2	28	
8	0	f	5/2		fp
9	1	p	3/2		
10	1	p	1/2	40	j4
11	0	g	9/2	50	
12	0	g	7/2		
13	1	d	5/2		
14	1	d	3/2		j5
15	2	s	1/2		
16	0	h	11/2	82	
17	0	h	9/2		
18	1	f	7/2		
19	1	f	5/2		j6
20	2	p	3/2		
21	2	p	1/2		
22	0	i	13/2	126	
23	0	i	11/2		
24	1	g	9/2		
25	1	g	7/2		
26	2	d	5/2		j7
27	2	d	3/2		
28	3	s	1/2		
29	0	j	15/2	184	

16 Sample inputs for Dens

16.1 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.

16.2 B(EL) and B(ML) values

(In many cases this will automaticaly be run to produce a gamma decay scheme.)

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,bw2404_2404   - 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 $\hbar\omega = 45\mathrm{A}^{-1/3}$-$25\mathrm{A}^{-2/3}$:

```
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,bw2400_2404   - 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,bw2400_2404
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,bw2400_2404
st
```

16.3 B(GT) values

(In many cases this will automatically be run.)

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,bw2404_1404   - n_i, n_f, p_i, p_f, A_c, name of OBD file
st
```

16.4 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,bw2400_2404
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.

17 Gamma decay

(In many cases the following will automatically be run.)

Proceedure for generating a gamma decay spectrum for nucleus x

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

2) Edit the *.den file if needed.

3) Run dens by doing with the following input

dens
ne
x

7) Type

gamma x

8) Data output is in x.deo.
```

18 Comparison to Oxbash

NuShellX does not work for $\hbar\omega$ type truncations. For this you will need to use the code Oxbash or NuShell.)

19 Short history of NushellX

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 is the
nushell core program with input and output
formats in the Oxbash style.

2008 NuShellX is a completely new code written code in Fortran
95 by W. D. M. Rae. It starts with input from Nushell
to create proton-neutron basis states.
NuShellX@MSU is the NuShellx core program with input and output
formats in the Oxbash style.

2009- many changes and additions to the code by Alex Brown