"KSHELL" User's guide

Noritaka Shimizu
Center for Nuclear Study, the University of Tokyo
e-mail: shimizu@cns.s.u-tokyo.ac.jp

December 7, 2020

Contents

1 Introduction

The “KSHELL” code enables us to perform nuclear shell-model calculations with M-scheme representation with the thick-restart block Lanczos method. This code is easily used on a Linux PC with a many-core CPU and OpenMP library. It is also used on a state-of-the-art massively parallel computer with hybrid MPI+OpenMP parallel programming. This manual explains how to use this code.

This code is equipped with a user-friendly dialogue interface to generate a shell script to run a job. By modifying this script, the code runs on various computers including supercomputers. Intel Fortran compiler with a Linux OS is most recommended. It can compute energy levels, spin, isospin, magnetic and quadrupole moments, E2/M1 transition probabilities, and one-particle spectroscopic factors. Up to tens of billions M-scheme dimension is capable, if enough memory is available on the computers.

The ground-state energy of ^{56}Ni with the pf shell and the KB3 interaction (one billion M-scheme dimension) was computed in 145 seconds using 512 nodes (8192 CPU cores) with the FX10 supercomputer, at the University of Tokyo.

2 Licence

3 Setup

3.1 Environments

Necessary software

- Fortran compiler (Fortran 95, with ISO-TR15581 extension)
- BLAS, LAPACK library
- python ver.2.4 or later (incompatible with python ver.3)

Optional software

- OpenMP for thread parallel
- MPI-2 library (+ MPI-IO) for node parallel

Tested environments

- Intel Fortran 11.1, Intel MKL, Linux OS

- GNU gfortran 4.6.3, BLAS, LAPACK, Linux OS (gfortran ver.4.2 or later)
- PGI Fortran 13.10, AMD-ACML library, Linux OS
- Fujitsu Fortran, SSLII, FX10 / K computer
- GNU gfortran 4.8.3, BLAS, LAPACK, Cygwin on Windows 7
- GNU gfortran on Windows Subsystem for Linux, Windows 10
- Intel Fortran, Intel MKL, Mac OS

For personal use, the latest Linux OS and Intel Fortran compiler are recommended. GNU gfortran is mostly compatible (LAPACK and BLAS libraries are required in this case). On MS-Windows 10, Windows Subsystem for Linux (gcc, gfortran, BLAS, LAPACK, and python) is easiest to use. In most cases, Makefile should be modified to a certain extent.

3.2 Installation

Move to the directory where the software is installed.

```
tar xvzf kshell.tar.gz
cd kshell/src

Edit Makefile for compiler etc. (default: Intel Fortran)

make

alias kshell_ui.py=(installed directory)/bin/kshell_ui.py
```

You may have to modify FC in the head of Makefile to Fortran compiler suitable for your environment. For MPI environment, the MPI option "-DMPI" should be added. Further modifications might be needed depending on your environment.

If you run the program with many CPU cores in a single node, MPI parallel is unnecessary. Just run "make" with the enabling OpenMP option such as "-qopenmp". The program automatically detects the number of CPU cores and runs the parallel threads. At the head of the log file, the program shows

```
nprocs   36   myrank   0
OpenMP  # of threads= 16
```

if the program succeeds in running with 36 MPI processes, each of which has 16 threads.

3.3 Directories and files

All files are contained in the following subdirectories in the "kshell" directory.

- src ... source files
  Makefile, class_stopwatch.F90, lib_matrix.F90, transit.F90, block_lanczos.F90, constant.f90, model_space.f90, bp_block.F90, count_dim.f90, operator_jscheme.f90, bp_block_inc.F90, harmonic_oscillator.f90, operator_mscheme.f90, wavefunction.F90, bp_expc_val.F90, interaction.f90, partition.F90, bp_io.F90, kshell.F90, rotation_group.f90, bridge_partitions.F90, lanczos.f90, sp_matrix_element.f90

- bin ... execution binaries
  - bin/kshell_ui.py ... Dialogue-type interface to generate a shell script
  - bin/gen_partition.py ... python script to generate a partition file
  - bin/nushell2snt.py ... python script to generate “snt” file from Nushell file
  - bin/count_dim.py ... python script to count M-scheme dimension
  - bin/kshell.exe ... Binary file for the Lanczos method.
  - bin/transit.exe ... Binary file for transition probabilities
  - bin/count_dim.exe ... Binary file to count the M-scheme dimension
- snt ... directory to contain the interaction (.snt) file
  - snt/ckpot.snt ... Cohen Kurath interaction (for p shell)
  - snt/w.snt ... Wildenthal’s USD interaction (for sd shell)
  - snt/gxpf1a.snt ... GXPF1A interaction (for pf shell)
  - Several proven effective interactions are quipped in this directory. You can place your hand-made interaction file here.
- test ... Test directory (not for use.)

4 Shell script and run

First, make your working directory and change the current directory to the working one. Secondly, run kshell_ui.py to generate a shell script. Finally, run the script for the PC, or modify and submit it for the job queuing system in a supercomputer.

4.1 Generation of the shell script

Run “kshell_ui.py” in your working directory. It is a dialogue-type interface similar to OXBASH and generates a shell-script batch file.

To avoid confusion, we recommend starting with a new, empty directory.

```
mkdir work    # make working directory
cd work
kshell_ui.py
....
```

Answer the questions asked by “kshell_ui.py”. The tab key for completion is convenient to answer them such as “snt” file name. As an example, we provide the answers to compute the 10 lowest energy levels of ^{24}Mg with USD interaction and the sd shell. The underlined texts are input in the following:

```
kshell_ui.py

----------------------------
  KSHELL user interface
```

```
    to generate job script.
-----------------------------

 MPI parallel?  Y/N (default:  No) :  n      y for MPI parallel

  ... generate shell script for a single node.

 model space and interaction file name (.snt)

 (e.g.  w or w.snt, TAB key to complete) :  w.snt


*************** specify a nuclide *******************


 number of valence protons and neutrons

 (ex.  2, 3 <CR>) <CR> to quit :  4,4      Numbers of active protons and neutrons


 name for script file (default: Mg24_w ):

 J, parity, number of lowest states
  (ex. 10           for 10 +parity, 10 -parity states w/o J-proj. (default)
       -5           for lowest five -parity states,
       0+3, 2+1     for lowest three 0+ states and one 2+ states,
       1.5-, 3.5+3  for lowest one 3/2- states and three 7/2+ states) :

 +10      Compute 10 low-lying positive-parity states


 truncation for "+" parity state in  Mg24_w_p.ptn
 truncation scheme ?
     (0): No truncation (default)
     (1): particle-hole truncation for orbit(s)
     (2): hw truncation
     (3): Both (1) and (2)

 :  0      0 for no truncation.  If you apply truncation, some questions follow
(described later).


generating partition file ........... done.

 --- input parameter ---
  beta_cm = 0.d0              # Lawson beta (MeV)
  eff_charge = 1.5, 0.5,      # effective charge
  fn_int = "w.snt"            # snt file
  gl = 1.0, 0.0,              # g-factor for orbital
  gs = 3.91, -2.678,          # g-factor for spin
  hw_type = 2                 # Harmonic oscillator parameter
  max_lanc_vec = 200          # iteration for Lanczos
  maxiter = 300               # iteration for TR-Lanczos
  mode_lv_hdd = 0             # Lanczos vector save in HDD or not
  n_block = 0                 # block size for block Lanczos
  n_restart_vec = 10          # restart vec for TR-Lanczos

modify parameter?
 (e.g.  maxiter = 300 for parameter change
        <CR>          for no more modification ) :
```

_ Detailed setup for Lanczos method. Just “Enter” without further modification. For example, `eff_charge = 1.8, 0.8` for effective charge. These parameters are explained in Sect.5.3.

```
 compute transition probabilities (E2/M1/E1) for
   Mg24_w ? Y/N (default: No) :
```

y “y” for computing transition probabilities. “n” for unnecessary.

```
*************** specify a nuclide ********************


 number of valence protons and neutrons
  (ex.  2, 3 <CR>)    <CR> to quit :
```

: _ Just “Enter” to quit. Otherwise, continue to input other nuclides by specifying the number of active protons and neutrons to obtain Gamow-Teller transitions and spectroscopic factors.

```
 Finish. Execute ./Mg24_w.sh
```

Mg24_w.sh and some other files are generated or copied in this directory. For a Linux PC, run “./Mg24_w.sh”. For a parallel computer with MPI, edit Mg24_w.sh to fit your environment and submit a job such as “qsub Mg24_w.sh”.

4.2 Setup for truncation scheme

At the “kshell_ui.py”, the truncation scheme is specified. It generates a file containing the information of truncation scheme, called “partition file” (Mg24_w_p.ptn in the previous example.) The program “kshell” reads the partition file to generate M-scheme model space for computation.

In the example of ^{24}Mg, since a negative parity state does not exist, only an input truncation scheme is needed for positive parity states. (Inquiries for negative-parity state follow if necessary.)

```
 truncation for "+" parity state in  Mg24_w_p.ptn
 truncation scheme ?
      0 : No trucation (default)
      1 : particle-hole truncation for orbit(s)
      2 : hw truncation
      3 : Both (1) and (2)
```

0 for no truncation, namely any M-scheme configuration inside the model space is active.

1 for specifying minimum and maximum occupation numbers of particular single-particle orbits. For example, if excitations up to two particles from the $0d5/2$ orbits are allowed in ^{24}Mg, specify the orbit occupation of the orbits 2 and 5 as minimum 6 and maximum 8.

3 for both restrictions 1 and 2 simultaneously.

```
   1

   #    n,  l,  j, tz,    spe
   1    0   2   3  -1     1.647
   2    0   2   5  -1    -3.948
   3    1   0   1  -1    -3.164
   4    0   2   3   1     1.647
   5    0   2   5   1    -3.948
   6    1   0   1   1    -3.164
 specify # of orbit(s) and min., max. occupation numbers for restriction
```

`# of orbit(s) for restriction?  (<CR> to quit):  2,5` proton d5/2 orbit and neutron d5/2 orbit for restriction.

`min., max.  restricted occupation numbers for the orbit(s) (or max only) :  6,8` Minimum and maximum occupation numbers of sum of proton d5/2 and neutron d5/2 orbits.

`# of orbit(s) for restriction?  (<CR> to quit):  _` Additional restrictions can be added.

An arbitrary truncation scheme is realized by preparing partition file by yourself.

4.3 Run the batch file

The Lanczos computation starts by running the generated batch file (Mg24_w.sh in the example). The name of the script is shown at the end of the kshell_ui.py process such as "Finish. Execute ./Mg24_w.sh".

In the case of MPI parallel, modify the script to fit the number of nodes, job queue, different usage of "mpiexec" and so on, and submit the queue (e.g. qsub ./Mg24_w.sh).

After completion of the computation, the results of the energies and transition probabilities are summarized in "summary_Mg24_w.txt". The detailed information, such as memory usage, moments and transition probabilities, is shown in log-file, "log_Mg24_w_m0p.txt".

5 Function details

5.1 Interaction file

Model space and effective interaction is contained in the original .snt file format. Some proven effective interactions are equipped in “kshell/snt”. Your own interaction is also used. It must be placed in a current directory or kshell/snt for kshell_ui.py.

By using “nushell2snt.py”, the model-space and interaction files in Nushell (or OXBASH) are transformed into .snt format. For example, USDA interaction is transformed into usda.snt by [1]

```
kshell/bin/nushell2snt.py sd.sp usda.int
```

5.2 Interaction file (.snt)

The .snt file consists of the definition of single-particle orbits, matrix elements of the one-body interaction, and those of the two-body interaction. The line with the head letter ”!” is a comment. The Hamiltonian is assumed to be hermitian and time-reversal symmetric, while isospin symmetry is not assumed. The Hamiltonian matrix elements are anti-symmetrizedand. Theelements are provided as real numbers and are listed without redundancy. The definition of the two-body matrix elements are :

$$H = \sum_{\alpha,\beta,m_\alpha,m_\beta} \epsilon_{\alpha\beta} c^\dagger_{\alpha m_\alpha} c_{\beta m_\beta} + \sum_{\alpha\le\beta,\gamma\le\delta,J,M} \langle\alpha\beta|V|\gamma\delta\rangle_J A^\dagger(\alpha\beta JM) A(\gamma\delta JM) \quad (1)$$

$$A^\dagger(\alpha\beta JM) = \frac{1}{\sqrt{1+\delta_{\alpha\beta}}} \sum_{m_\alpha,m_\beta} \langle j_\alpha, m_\alpha, j_\beta.m_\beta|JM\rangle c^\dagger_{j_\alpha,m_\alpha} c^\dagger_{j_\beta,m_\beta}$$

$$\epsilon_{\alpha\beta} = \epsilon_{\beta\alpha} \quad (2)$$

$$\langle\alpha\beta|V|\gamma\delta\rangle_J = \langle\gamma\delta|V|\alpha\beta\rangle_J \quad (3)$$

$$\langle\alpha\beta|V|\gamma\delta\rangle_J = -(-1)^{j_\gamma+j_\delta-J}\langle\alpha\beta|V|\delta\gamma\rangle_J \quad (4)$$

Example: w.snt (referred to as USD, namely universal sd interaction by B. A. Brown and B. H. Wildenthal)

```
! "!" denote comment line
!
! model space
! proton-orbit, neutron-orbit, proton core, neutron core
   3   3     8   8
! orbit    n   l   j  tz
   1       0   2   3  -1    !  1 = p 0d_3/2
   2       0   2   5  -1    !  2 = p 0d_5/2
```

[1]In many interactions, the TBME should be scaled by mass, such as $(A/A_0)^{-0.3}$. Although there is no specification for mass dependence of TBME in int file at some interaction files, mass dependence is automatically introduced based on the file name in Nushell and OXBASH. Such int file cannot be treated appropriately in nushell2snt.py. Some results by Nushell and KSHELL should be compared in a small computation for a check.

```
   3       1   0   1  -1    !  3 = p 1s_1/2
   4       0   2   3   1    !  4 = n 0d_3/2
   5       0   2   5   1    !  5 = n 0d_5/2
   6       1   0   1   1    !  6 = n 1s_1/2
! one-body interaction
! number of lines,  method1
     6   0
!  i   j      <i|V|j>
   1   1      1.64658
   2   2     -3.94780
   3   3     -3.16354
   4   4      1.64658
   5   5     -3.94780
   6   6     -3.16354
! two-body interaction (TBME)
! # of lines,  method2  A   mass dependence factor
   158   1  18  -0.30000
!  i   j   k   l     J      <i,j| V | k,l>_J
   1   1   1   1     0     -2.18450
   1   1   1   1     2     -0.06650
```

... 158 lines continued

Two-body matrix elements (TBMEs, $\langle i,j|V|k,l\rangle_J$ are shown as `i j k l J TBME`, not using the isospin index.

The mass dependence of the TBME is

method2 = 0   No mass dependence

method2 = 1   (mass number/A) ˆ (mass dependence factor).

5.3 Parameters for "kshell.exe"

You may want to edit the shell script by yourself based on the file generated by kshell_ui.py, after some experiences of using this code. The following are the description of the variables in Fortran namelist of the binary "kshell". The numbers in parenthesis are default values.

- mtot   Doubled J_z quantum number of the system.
- n_eigen (1)   Number of eigenstates to be computed.
- n_block (0)   Block size for block Lanczos method.
  0: the Lanczos method without block algorithm
- n_restart_vec (10)   Number of vectors to restart in Thick-restart Lanczos.
- max_lanc_vec (100)   A maximum number of vectors in Thick-restart Lanczos.
- maxiter (300)   Maximum number of iterations of the restart process in the thick-restart Lanczos method.
- hw_type (1)   Empirical formula to determine harmonic oscillator $\hbar\omega$.
  $1 : \hbar\omega = 41A^{-1/3}$ $2 : \hbar\omega = 45A^{-1/3} - 25A^{-2/3}$

- is_double_j (.false.) J-projection ($2J$=mtot) during Lanczos process.
- is_load_snapshot (.false.) Load snapshot file which stores the status of the computation (tmp_*)
- fn_save_wave File name for output wave function.
- fn_load_wave (random) File name for initial vector in Lanczos method.
- beta_cm (0.0) β_{cm} parameter for Lawson method. (beta_cm= $\beta_{CM}\hbar\omega/A$, the same definition as Nushell/OXBASH).

$$H = H_{\text{int}} + \beta_{\text{CM}}(H_{\text{CM}} - \frac{3}{2}\hbar\omega) \tag{5}$$

- eff_charge (1.0, 0.0) Effective charge for E2 transition and quadrupole moment.
- gl (1.0, 0.0), gs (5.586, -3.826) g factors for orbit (gl) and spin (gs) to compute M1 transition and magnetic moment.
- tol (0.000001) Convergence condition for Lanczos method. The iteration stops if the deviation of energy at 1 iteration is smaller than this value.
- mode_lv_hdd (0) Lanczos vectors are stored on memory (0) or on hard disk (1).

5.4 Partition file

As an example, the partition file Mg24_w_p.ptn is shown in the following for a positive parity state, and a 2-particle excitation is allowed from the $0d_{5/2}$ orbit.

```
# comment line with '#'
# particle-hole truncation orbit(s) : min., max.
#      [2, 5] :  6 8
# partition file of w.snt  Z=4  N=4  parity=+1
 4 4 1   # # of active protons, neutrons, parity
# num. of  proton partition, neutron partition
 6 6     # line number of following proton, neutron partitions
# proton partition
     1     0  2  2  # index, occupation numbers of proton d3/2, d5/2, s1/2
     2     0  3  1
     3     0  4  0
     4     1  2  1
     5     1  3  0
     6     2  2  0
# neutron partition
     1     0  2  2  # index, occupation numbers of neutron d3/2, d5/2, s1/2
     2     0  3  1
     3     0  4  0
     4     1  2  1
     5     1  3  0
     6     2  2  0
```

```
# partition of proton and neutron
15             # combination of proton and neutron (# of the following lines)
    1     3    # proton 1 (0,2,2) - neutron 3 (0,4,0)
    2     2
    2     3
    2     5
    3     1
    3     2
    3     3
    3     4
    3     5
    3     6
    4     3
    5     2
    5     3
    5     5
    6     3
```

Note that partition files for positive parity and negative parity should be prepared individually.

5.5 Restrictions of this software

If your Fortran compiler does not support integer(16), such as with the Intel Fortran compiler, the maximum number of single-particle states of protons or neutrons is 62. In the case where gfortran and kmbit = 16 in constant.f90, the maximum is 126.

6 Frequently asked questions

1. Compiler error with "popcnt, poppar is not available".

   Remove the "#" in the head of the following line of Makefile

   ```
   # FFLAGS := $(FFLAGS) -DNO_POPCNT  # avoid using popcnt, poppar
   ```

2. Compiler error with "Error: There is no specific subroutine for the generic mpi_file_read_at_all".

   "-DSPARC" option in Makefle may solve this problem.

3. The program stops with "Segmentation Fault".

   One of the probable causes is the shortage of stack size. It is worth trying `ulimit -s unlimited` , or using the "-heap-arrays" option for Intel Fortran compiler. Another possible cause is the lack of the stack size of the OpenMP thread. In the batch file, check the following line

   ```
     export OMP_STACKSIZE=1g
   ```

   If it does not exist, try adding this line. Or, try incraseing the number.

4. The program stops with "ERROR: increase max_ptnn".

   Increase the max_ptnn

   ```
     integer, parameter :: max_idx=20, max_ptnn=300000
   ```

   in bridge_partitions.F90. The same prescription is valid for other variables.

5. How to count M-scheme / J-scheme dimension.

   Prepare .snt file and partition file (.ptn), and execute "kshell/bin/count_dim.exe foo.snt bar.ptn".

6. Insufficient memory.

   At least two Lanczos vectors should be stored on the memory for the Lanczos method (mode_lv_hdd = 0). $D \times 16$ Bytes main memory is needed where D is the M-scheme dimension with double-precision real numbers. For example, $D = 10^9$ M-scheme dimension in pf-shell ^{56}Ni demands 16GB memory at least. The M-scheme dimension is shown in log file as

   ```
   total m-scheme dimension      1087455228  = 10** 9.04
   ```

   The required memory size for storing Lanczos vectors in 1 node is also shown as

   ```
   Memory / process is:     4.051 GB x     10 =     40.511 GB
   ```

   For some additional memory usage, the memory needed is 1.2 times this value.

   (a) To store old Lanczos vectors during computation
       set mode_lv_hdd = 1 at kshell_ui.py. (0 for default)

   (b) Store the Lanczos vector in a single-precision real number.
       This halves the required memory size. Set kwf=4 in constant.f90 as in the following.

       ```
       !  integer, parameter :: kwf = 8
         integer, parameter :: kwf = 4
       ```

       Note that in a few cases with a small M-scheme dimension, this change might cause a bad convergence of the Lanczos method with J-projection. (Double Lanczos method.)

7. In MS-Windows Cygwin or Mac OSX, the program stops with "libgomp: Thread creation failed: Resource temporarily unavailable".

   Memory size is insufficient for starting OpenMP threads. Set `export OMP_STACKSIZE=1g` in the script. In Makefile, add `-Wl,--stack,40000000` to FFLAGS. If it fails, try increasing the numbers, or decreasing the number of threads (e.g. `set export OMP_NUM_THREADS=4` in the script). If these trials fail in vain, remove the OpenMP support in `export OMP_NUM_THREADS=1`. Instead of the OpenMP support, you can try flat MPI parallel to use the cores efficiently.

8. In the block Lanczos method, "mode_lv_hdd=1" cannot be used. Please specify "mode_lv_hdd=0".

7 Acknowledgments

This code was partly supported by KAKENHI grants (17K05433, 25870168, 15K05094) from JSPS, the HPCI Strategic Program Field 5, Priority issue 9 to be tackled by using Post K Computer from MEXT and JICFuS, and the CNS-RIKEN joint project for large-scale nuclear structure calculations. The numerical calculation was performed partly on the FX10 supercomputer at the University of Tokyo, the K computer at AICS (hp190160,hp180179, hp170230, hp160211, hp150224, hp140210, hp130024), and the Oakforest-PACS system (hp200130) for Multidisciplinary Computational Sciences Project of Tsukuba University (xg18i035).

We acknowledge T. Abe, Y. Futamura, M. Honma, T. Ichikawa, Y. Iwata, C. W. Johnson, H. Matsufuru, T. Mizusaki, T. Miyagi, J. E. Midtbø, T. Otsuka, C. Qi, T. Sakurai, T. Togashi, N. Tsunoda, Y. Tsunoda, Y. Utsuno, S. Yoshida, T. Yoshida, and C. Yuan for their feedbacks, valuable discussions, contributions and/or tests of the KSHELL code developments.

The user interface of OXBASH and Nushell, and the design of MSHELL and MSHELL64 were helpful for the code developments.