Code SPINDIS: nuclear shell-model level densities at high energies with a simplified pairing interaction

D.K. Sunko
Physics Department,
Faculty of Science, University of Zagreb,
Zagreb, Croatia

Abstract

A comprehensive package for the calculation of shell-model level densities is presented. A basic combinatorial result, and a sophisticated configuration generator, allow one to obtain exact predictions of the shell model, including the spin distributions, in very large configuration spaces, going up to tens of MeV excitation in a heavy nucleus. A simplified pairing interaction, acting only within levels, is also taken into account. Utility programs include standardized generation of single-particle input, parametrized fitting of the calculated spin distributions, and various ways to produce graphical output.

1 Program Summary

- *Title of package:*
  SPINDIS

- *Licensing provisions:*
  none

- *Computer for which the program has been designed, and others for which it has been tested:*
  any Unix workstation or mainframe; test runs have been obtained on DEC Ultrix, DEC-Alpha, HP Apollo, SGI Indigo 2, IBM Sp2.

- *Operating system:*
  Unix

- *Programming language:*
  C

- *Number of lines in distributed program, including test data:*
  5253 lines of source code with comments, and 165,613 lines of test-run output. Compressed archive takes 1.1 Mb.

- *Keywords:*
  Level density, spin distributions, nuclear reactions, neutron resonances.

- *Nature of physical problem:*
  The estimation of the final-state level density is an important ingredient in all calculations of reaction rates. Typically, the level density can be measured directly only in narrow windows of energy and spin. The present program efficiently calculates the exact level densities predicted by the shell model up to very high excitation, even in heavy nuclei, and with a simplified pairing interaction taken into account. When fitted to known neutron resonance data, the program provides a benchmark estimate of the level densities and spin distributions for a given nucleus.

- *Method of solution:*

  The generating functions of spin distributions of a shell-model configuration with given seniority are used, along with an efficient method for the calculation of their coefficients. The distribution calculator is provided with configurations by the configuration generator, which uses auxiliary tables and C's pointer mechanism to avoid searching through configurations which are predictably too high in energy.

- *Restrictions on the complexity of the problem:*

  Spherical shell-model single-particle levels for both protons and neutrons are input to the program. Only the short-range part of the residual interaction is considered, and this in a simplified form, diagonal in seniority. There is no restriction in principle on the input itself, since the basic combinatorial algorithms are guaranteed to terminate. However, the running time can become impractical.

- *Typical running times:*

  Depend heavily on the number of single-particle levels involved. On a DEC Alpha AXP running at 233 MHz, from one second for ^{59}Ni up to 17 MeV excitation, to twenty-six hours for ^{244}Am up to 30 MeV excitation.

2 Long write-up

2.1 Introduction

The calculation of nuclear level densities is a problem of long standing in nuclear physics, with important ramifications in heavy-ion reactions, astrophysics, and reactor design. There are two basic items of information every such calculation must provide: first, the overall trend of increase of the number of states with energy; second, the spin distribution at a given energy. General arguments give the following answers to these questions: the increase is as the exponential of the square root of the energy [1], while the distribution of spin projections is a Gaussian [2].

The main problem in the estimation of level densities is how to obtain the actual parameters, governing these trends, which will provide concrete values for a given nucleus. In other words, in terms of trends all nuclei are the same, but for practical purposes they are not. A vast body of research has grown out of efforts to provide level densities with individual nuclear identity. For a brief but fairly complete overview of the literature, the reader is referred to [3].

Both of the above-mentioned general answers reflect the saddle-point assumption that the number of degrees of freedom available to the nucleus is infinite. This is much nearer to the truth for the energy distribution than for the spin distribution. Furthermore, the energies of orthogonalized orbitals may be added without restriction, while adding spin projections is subject to the requirement that the resulting wave function be antisymmetric. Thus not only does the Gaussian overestimate the number of spins in its tails, which extend to infinity, but also the connection between the quantum numbers of subshells participating in a spin distribution, and the parameters of its best Gaussian fit, is more complicated [4] than suggested by the standard 'central limit' argument [2]. In order to parametrize the spin distributions equally well at high and low spins, a third phenomenological parameter was introduced in [3], in addition to the two Gaussian ones.

In the present package, level densities are calculated beginning with the spherical shell model as input, and the number of states contributing at a given energy is directly counted by a combinatorial algorithm. This solves the problem of individuality, since each nucleus is uniquely specified by its shell-model ground state, but, as in all approaches beginning with free particles, opens the question of residual interactions. These can be of two main kinds, short-range or pairing, which tend to arrange nucleons in time-reversed pairs of orbitals, and long-range or quadrupole, which express the ability of the nucleus to deform. The importance of these stems not from their dominance at high energy, where one obviously expects the nucleus to be neither superconducting nor deformed, but in their specific action on states near the yrast line, through which a nucleus must pass while deexciting. Experience suggests that the bulk of the effect of residual interactions may be included in a shift of the origin of the energy [5].

Only the short-range part of the residual interaction is included in this work, and in a simplified form, diagonal in seniority:

$$H = \sum_i \epsilon_i n_i + G \sum_i S_+^{(i)} S_-^{(i)} \ , \tag{1}$$

Here n_i is the number operator for the i-th single-particle level, and $S_+^{(i)}$, $S_-^{(i)}$ are quasispin operators for the same level. They are diagonal in seniority; if it is specified for all levels, their contribution to the energy is [6]

$$-\frac{G}{4} \sum_i (n_i - s_i)(2j_i + 1 - n_i - s_i + 2) \ , \tag{2}$$

where s_i are the seniorities and j_i the single-particle spins of the various levels.

This force acts selectively near the yrast line, pulling down states with a large component of zero-coupled pairs, and thus also accounts in a natural way for the energy shift mentioned above. If G is used as a free parameter, fitted to the neutron resonance window [3], one also expects to take into account the part of the shift due to the long-range residual force. (It should be noted that phenomenological values of G in the model (1) are significantly higher than the $\sim 20/A$ MeV often quoted in the literature, owing to the inclusion of only the diagonal part of the pairing interaction.) The question which remains is the effect of the long-range force along the yrast line. Here a comparison with a boson model, which is of course dominated by the quadrupole force, is quite encouraging [7]. It appears that the main effect of the quadrupole force is to scatter states up and down in energy, keeping their count in a bin of any reasonable width more or less the same. This has been noticed before [8].

Traditionally, a serious disadvantage of the shell-model approach to the calculation of nuclear level densities has been the very large size of the combinatorial problem involved. This has two parts to it; first is the generation of configurations, second the calculation of spin distributions for a given configuration. This second problem has been completely solved [9] for the model (1), allowing for the first time an exact combinatorial calculation of level densities with a residual interaction, however simple, in large configuration spaces. Earlier treatments of pairing in a combinatorial context did not consistently take into account promoted pairs, thus the pairing contribution was taken to be linear in the number of pairs [10], or else their single-particle energy was simply added to the solution of a simplified BCS equation [11]. The crucial issue remaining for a successful implementation of the method was the design of an efficient configuration generator, to be described in the next section.

One attractive feature of an exact calculation is that it is well adapted to model investigations [12]. Another is that, despite all its simplifications, it can always serve as a well-defined benchmark for comparing different nuclei within the method, or the results of different methods. In order for this to be so, it is necessary to standardize the input to the method. One such standardization is included in the present program package. The utility program 'infile' will produce input for a nucleus by name and mass, using the Seeger single-particle levels, as given in Ref. [11], scaled by the cubic root of the mass number.

2.2 The working of the program

The present program is a new implementation of the FORTRAN package used to produce previously published work. It is completely rewritten in C, and contains significant algorithmic improvements. The distributed version of the program contains recalculated level densities corresponding to published results [3] obtained with the old code. The differences are insignificant at the level of the published figures. However, the new calculations did in every case find some states missed by the old version, which had included heuristics to make the code run faster. All such heuristics have been removed from the present C version.

The basic problem in the efficient generation of configurations is that, while the number of configurations up to some reasonable energy is only a small fraction of the total number, the elementary 'odometer' algorithm generates configurations in lexicographic order, without regard to their energy, and thus in fact spends most of its time in parts of configuration space above the energy of interest.

The way to avoid this problem is illustrated in Fig. 1. At the beginning of the run, auxiliary tables are generated, such that the minimum energy required to place n_i particles above the i-th level is known. The elementary move of the odometer is to move one particle from the highest occupied level to the first higher level. Before doing that, it checks the auxiliary table; if the energy involved is above the given cutoff, it will perform the other move available to it, moving all the particles down to the next lower occupied level, and rebuilding a configuration from there. Here C's pointer mechanism is very convenient, because it obviates the need to search for the lower occupied level. Instead, one has a linked list connecting occupied levels in the downward direction in energy, which is easily updated for each new configuration.

The actual implementation of this idea is more complicated, owing to the need to specify configurations by seniority. One has two loops, an outer one for uncoupled particles, and an inner one for pairs. The inner one works much as described above, except that the auxiliary tables for pairs need to be recalculated for each uncoupled-particle configuration. The outer one cannot work in that way, because of the blocking effect: for example, the lexicographically first configuration of uncoupled particles, with all of them in the lowest shell-model orbitals, is in fact quite high in energy, because when the pairs are filled in, the holes accompanying the uncoupled particles will appear deeply below the Fermi level. Thus pushing uncoupled particles upward in their single-particle level scheme may result in an overall lowering of the configuration energy. There is a special routine, logically part of the configuration generator, that will find the first admissible configuration, if any, of uncoupled particles, without having to cycle through all the forbidden configurations which precede it in lexicographic order. It is activated whenever the ground state of zero-coupled pairs for a given uncoupled-particle configuration turns out to have an energy above the cutoff.

All of the above considerations apply to the single-particle energy scheme. The effects of the pairing force are considerably more unpredictable, since (2) is not linear. The best policy found to treat them was to disregard them until the last moment. All that the configuration generators do to take the pairing force into account is to search for configurations up to energies higher than the desired cutoff. (The user specifies how much higher.) When a configuration is found, the effects of the pairing force are included, and if it is then below the cutoff, it is accepted.

There are two consequences of this policy. First, the actual excitation energy to which one can claim all levels of (1) have been found often turns out higher than the cutoff specified by the user. This is because the pairing force pulls some states to negative energies relative to the shell-model ground state, which increases the range up to the cutoff. Second, one cannot be sure, on the basis of a single run, that all the levels of (1) in the desired range have actually been found, until one checks that the additional 'buffer' energy has not been set too low. In practice, this means that the results up to some excitation should become insensitive to further changes in the cutoff, buffer, or both. In order to help the user decide whether to try another run, the program prints out the largest value of the pairing contribution encountered. If this is less than the value of the buffer, no additional run should be needed.

Generally speaking, the present version of the program is optimized under the constraint that not a single level of (1) should be lost. The only heuristic left in the searching for states, the buffer energy discussed above, is under the user's control.

Once a level has been found, it is passed to the distribution calculator, which uses the algorithm of [9] to find the spin distribution; of course, this is done only once for each distribution of uncoupled particles.

Since there is no interaction between neutrons and protons, the level densities of the two are computed separately, and then simply combined by the angular momentum coupling rules. The question arises here how to save the level-density information from the neutron and proton distributions. One could do it level by level, but the resulting files would be very large, and combining levels one by one would take an impractically long time. Instead, the levels found are binned, and only the bins are saved. Now in combining the bins, information is lost as to where the original levels were, introducing an uncertainty of twice the bin width in the combined density. In order to minimize this effect, the bins for the proton and neutron densities are by default a factor of five smaller than the bins for the combined result. The level densities (but not the cumulative level number) are additionally smoothed by a pairwise averaging of the number of levels in neighboring bins.

All possible combinations of the proton and neutron results are made, giving a combined density which goes up to the sum of the maximum proton and neutron energies. On the other hand, one is certain to have all states only up to the lesser of these two energies: for example, there is a single level at both the top of the proton and neutron densities, and combining these two gives another single level, at an energy where there are obviously many others. Rather than combine the levels only up to the 'safe' energy, the user is simply informed what this energy is.

2.3 Description of the program package

The directory tree is shown in Fig. 2. After compilation, the whole tree takes about 3.3 Mb of storage, most of which (2.3 Mb) is in the 'Am' directory, which contains a large sample run.

2.3.1 Directory 'spindis'

This is the root of the tree, containing a README file, a makefile, and a setup file for the enviroment variables. It is assumed that the user is using the C-shell.

2.3.2 Directory 'bin'

This is where the executable files are kept, along with the shell scripts that drive them. The package is designed to have a very simple interface, with little need to understand which executable does what. The commands which suffice for normal use are three:

- *infile:* executable, produces standard input files using the Seeger single-particle levels;
- *Start:* shell script, runs the other executables to produce the output;
- *table:* executable, tabulates various relationships from the output of Start, such as level density vs. energy, for input to the user's graphics packages.

2.3.3 Directory 'data'

A suggested organization of the program output, with subdirectories labeled by chemical symbols. In the distributed version, these contain three sample outputs, one of which (Ni) is small enough to run quickly on almost any machine, while the other two correspond to published work [3].

2.3.4 Directory 'graphics'

This contains scripts to produce level-density vs. energy graphics on Mathematica [13] and S-PLUS [14] platforms. There is also a fully commented configuration file, which is read by both these scripts.

2.3.5 Directory 'sources'

This consists of four subdirectories. Subdirectory 'c' contains the sources of the various subroutines, and subdirectory 'o' the object files. (The script driving the 'make' utility automatically puts the object and executable files where they belong.) The subdirectory 'main' contains the main program sources, while 'h' contains the various headers. One of these, 'switches.h', allows the main programs to be reconfigured, for instance to save levels one by one, instead of binning them 'on the fly' and saving the bin contents only. Another, 'constants.h', adjusts such things as the bin width.

2.4 Test run output

Here we give the transcript of the session which created the files in the directory 'Ni'. First use is assumed, so the setup and compilation steps are included. Comments are given in italics.

Set up enviroment variables, and compile the package. Options ('silent' in this case) get passed to the 'make' utility.

```
% cd spindis
% source Spinsetup
% Makespin -s
% rehash
% cd data/Ni
```

Create input files for nickel with A=59, with a cutoff of 10 MeV, pairing constant G=0.5 MeV, and buffer of 15 MeV, so that levels up to 25 MeV in the single particle scheme will be searched. Start the computation.

```
% infile Ni59 10 0.5 15
% Start Ni59
```

Now look at the files created. Those with suffix '.in' where created by 'infile'. Ni59.log is a log of the run. The '.out' files are barely human-readable.

```
% ls
Ni59.log            Ni59_n_table.out   Ni59_p_table.out
Ni59_n.in           Ni59_p.in          Ni59_spin.out
Ni59_n_spin.out     Ni59_p_spin.out    Ni59_table.out
```

The 'table' program works as an interface to the '.out' files. Here it is used to look at the level density ('d' option) up to the safe energy ('s' option). Note how the large value of the pairing force has produced a range of 17 MeV in the output, while a cutoff of only 10 MeV was specified in the input files. The densities are given in natural logarithms.

```
% table -ds Ni59
0.500  0.000
1.000  0.000
1.500  0.000
2.000  1.946
2.500  1.946
3.000  1.099
3.500  2.398
4.000  2.565
4.500  2.708
5.000  3.584
5.500  3.951
6.000  3.761
6.500  4.454
7.000  4.554
7.500  5.684
8.000  6.038
```

```
8.500  5.914
9.000  6.547
9.500  7.002
10.000  7.262
10.500  7.512
11.000  7.809
11.500  8.145
12.000  8.503
12.500  8.643
13.000  9.244
13.500  9.558
14.000  9.737
14.500  10.020
15.000  10.400
15.500  10.587
16.000  10.974
16.500  11.329
17.000  11.486
```

This gives the spin distribution at the energy of 12.2 MeV (i.e., in the bin from 12.0 to 12.5 MeV) as a function of $(J + 1/2)^2$*, again in natural logarithms.*

```
% table -e12.2 Ni59
1.000  4.997
4.000  5.624
9.000  5.905
16.000  6.026
25.000  6.031
36.000  5.932
49.000  5.746
64.000  5.472
81.000  5.112
```

```
100.000  4.654
121.000  4.078
144.000  3.296
169.000  2.303
196.000  0.693
```

This is a two-parameter (Gaussian) fit to the numbers above. Note how even in this small example, the Gaussian is a factor of two off at $(J+1/2)^2 = 144$*, or* $J = 23/2$*.*

```
% table -Fe12.2 Ni59
1.000  5.003
4.000  5.622
9.000  5.906
16.000  6.022
25.000  6.025
36.000  5.938
49.000  5.774
64.000  5.541
81.000  5.242
100.000  4.883
121.000  4.464
144.000  3.989
169.000  3.457
196.000  2.870
```

This is a three-parameter fit, as in Ref. [3].

```
% table -fe12.2 Ni59
1.000  5.003
4.000  5.622
9.000  5.905
16.000  6.021
25.000  6.021
```

```
36.000  5.928
49.000  5.748
64.000  5.483
81.000  5.124
100.000  4.659
121.000  4.062
144.000  3.300
169.000  2.321
196.000  1.051
```

Look at the log. The 'p' after 'Ni59' refers to the proton input. First comes a repetition of the input file, with explanations:

```
% cat Ni59.log
spiden Ni59 p
Number of j-shells : 31
Number of particles : 28

[  i ] : parity, 2*j+1 , energy :
[  1 ] :    1,      2,   0.000
[  2 ] :   -1,      4,   7.398
[  3 ] :   -1,      2,  10.557
[  4 ] :    1,      6,  14.719
[  5 ] :    1,      4,  18.906
[  6 ] :    1,      2,  19.497
[  7 ] :   -1,      8,  21.937 <-- last occupied orbital, with 8 particles
[  8 ] :   -1,      6,  26.869
[  9 ] :   -1,      4,  27.716
[ 10 ] :    1,     10,  29.052
[ 11 ] :   -1,      2,  29.848
[ 12 ] :    1,      8,  34.601
[ 13 ] :    1,      6,  35.628
[ 14 ] :   -1,     12,  36.090
```

```
[ 15 ] :    1,      4,  38.736
[ 16 ] :    1,      2,  39.173
[ 17 ] :   -1,     10,  42.153
[ 18 ] :    1,     14,  43.052
[ 19 ] :   -1,      8,  43.360
[ 20 ] :   -1,      6,  47.213
[ 21 ] :   -1,      4,  47.881
[ 22 ] :   -1,      2,  49.525
[ 23 ] :    1,     12,  49.576
[ 24 ] :   -1,     16,  49.962
[ 25 ] :    1,     10,  50.886
[ 26 ] :    1,      8,  55.407
[ 27 ] :    1,      6,  56.255
[ 28 ] :    1,     18,  56.794
[ 29 ] :   -1,     14,  56.871
[ 30 ] :   -1,     12,  58.284
[ 31 ] :    1,      4,  58.747
```

```
The energy range is 1187.714. Maximum excitation energy :   10.000
Pairing force :    0.500
Additional energy (buffer against pairing force) :   15.000
```

The active levels, and their maximal occupation, are chosen so as not to surpass 10+15=25 MeV.

```
Levels taken into account (with a total of 28 particles):


  __ parity
 |    __ 2j+1
 |   |      __ energy
 |   |     |       __ maximum no. of pairs
 |   |     |      |   __ max. no. of uncp. particles
 |   |     |      |  |
```

```
 1,  2,   0.000 [ 1, 1]
-1,  4,   7.398 [ 2, 1]
-1,  2,  10.557 [ 1, 1]
 1,  6,  14.719 [ 3, 3]
 1,  4,  18.906 [ 2, 2]
 1,  2,  19.497 [ 1, 1]
-1,  8,  21.937 [ 4, 4]
-1,  6,  26.869 [ 3, 3]
-1,  4,  27.716 [ 2, 2]
 1, 10,  29.052 [ 3, 3]
-1,  2,  29.848 [ 1, 1]
 1,  8,  34.601 [ 1, 1]
 1,  6,  35.628 [ 1, 1]
-1, 12,  36.090 [ 1, 1]
 1,  4,  38.736 [ 1, 1]
 1,  2,  39.173 [ 1, 1]
-1, 10,  42.153 [ 1, 1]
 1, 14,  43.052 [ 1, 1]
-1,  8,  43.360 [ 1, 1]
Energy of shell-model ground state:  429.134
With pairing included :  422.134
```

Here begins the actual run. The number of uncoupled-particle (seniority) configurations found during the run is given for each number of broken pairs. 757 levels were found up to 25 MeV, only 105 of which were below 10 MeV after the pairing force was included. (These levels are, of course, still hugely degenerate.) All seniority configurations found yielded at least one accepted level, so 0 were rejected. The maximal contribution of the pairing force was 12 MeV, indicating the value of 15 MeV for the buffer to have been more than ample.

```
 0 uncoupled particles :      1 seniority configurations.
```

```
 2 uncoupled particles :     39 seniority configurations.
 4 uncoupled particles :     31 seniority configurations.
 6 uncoupled particles :      1 seniority configurations.
 8 uncoupled particles :      0 seniority configurations.

757.0 levels found, with 105.0 accepted.
0 seniority configurations rejected.
Maximal pairing shift encountered :   12.000

Real   0.3
User   0.1
System 0.1
```

The density program would have been activated if the main program had been configured not to bin the levels, but to save them one by one.

```
density Ni59 p
The density program has not been activated.

Real   0.0
User   0.0
System 0.0
```

Now the same sort of thing follows for neutrons, omitted here. At the end, the proton and neutron distributions are combined. The restricted output program can be activated to produce level densities in some window of spins and parities, but a better way to do this is to use the 'table' program with the 'r' option.

```
combine Ni59

Real   0.6
User   0.5
System 0.0
```

```
restrict Ni59
The restricted output program has not been activated.

Real   0.0
User   0.0
System 0.0
```

3 Acknowledgements

The author would like to thank V. Paar for helpful suggestions and a careful reading of the manuscript. This work was supported by the Croatian Government under Project No. 1–03–235.

References

[1] H.A. Bethe, Rev. Mod. Phys. **9** (1937) 69.

[2] T. Ericson, Adv. Phys. **9** (1960) 425.

[3] V. Paar, D.K. Sunko, S. Brant, M.G. Mustafa, and R.G. Lanier, Z. Phys. **A345** (1993) 343.

[4] D.K. Sunko, Phys. Lett. **B201** (1988) 7.

[5] T. von Egidy, H.H. Schmidt, and A.N. Bekhami, Nucl. Phys. **A454** (1986) 109; **A481** (1988) 189.

[6] P. Ring and P. Schuck, *The Nuclear Many-Body Problem*, Springer, Berlin, Heidelberg, New York, 1980.

[7] S. Brant *et al.*, to be published.

[8] K. Sato, Y. Takahashi and S. Yoshida, Z. Phys. **A339**, (1991) 129.

[9] D.K. Sunko, Phys. Rev. **C35** (1987) 1936.

[10] F.C. Williams, Nucl. Phys. **A133** (1969) 33.

[11] M. Hillman and J.R Grover, Phys. Rev. **185** (1969) 1303.

[12] D.K. Sunko, Ann. Phys. (NY) **199** (1990) 227.

[13] Mathematica, Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237, USA.

[14] S-PLUS, Statistical Sciences, Inc., 1700 Westlake Avenue North, Suite 500, Seattle, WA 98109, USA.

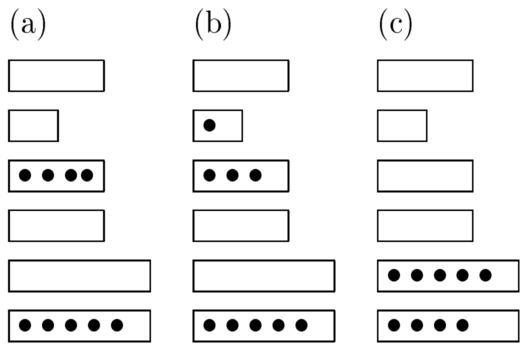

Figure 1: The operation of the configuration generator. If (b) is too high in energy, one can instead go directly from (a) to (c), in this example saving 11 steps.

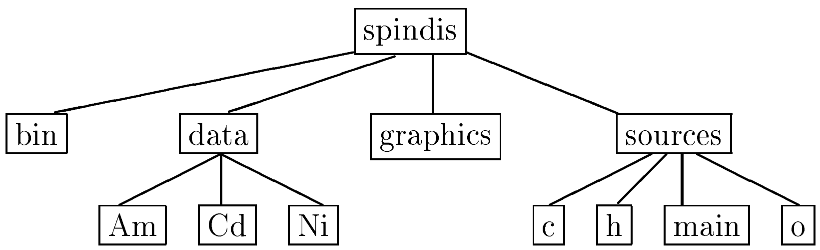

Figure 2: The directory tree of the distributed version of the program.