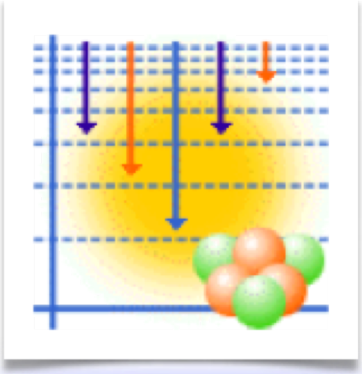
ENSDF Analysis & Utility Codes

F.G. Kondev
Physics Division, Argonne National Laboratory
kondev@anl.gov



Joint ICTP-IAEA Workshop on Nuclear Structure and Decay Data:
Theory, Experiment and Evaluation, Trieste IT 2018

https://www-nds.iaea.org/public/ensdf_pgm/

Utility Codes:

- ✓ fmtchk
- ✓ Java-NDS
- ✓ ensdat
- ✓ AveTools (V.AveLib)
- ✓ xls2ens

Analysis Codes:

- ✓ alphad
- ✓ gabs
- ✓ **gtol**
- ✓ **logft**
- ✓ **pandora**
- ✓ jgamut
- ✓ **ruler**
- ✓ radlist
- ✓ BrIcc
- ✓ BrIccMixing

fmtchk

This program analyzes the format of an ENSDF formatted file to verify that it conforms to "EVALUATED NUCLEAR STRUCTURE DATA FILE. A Manual for Preparation of Data Sets" by J.K. Tuli, Brookhaven National Laboratory Report BNL-NCS-51655-01/02-Rev (February 2001) and subsequent memos.

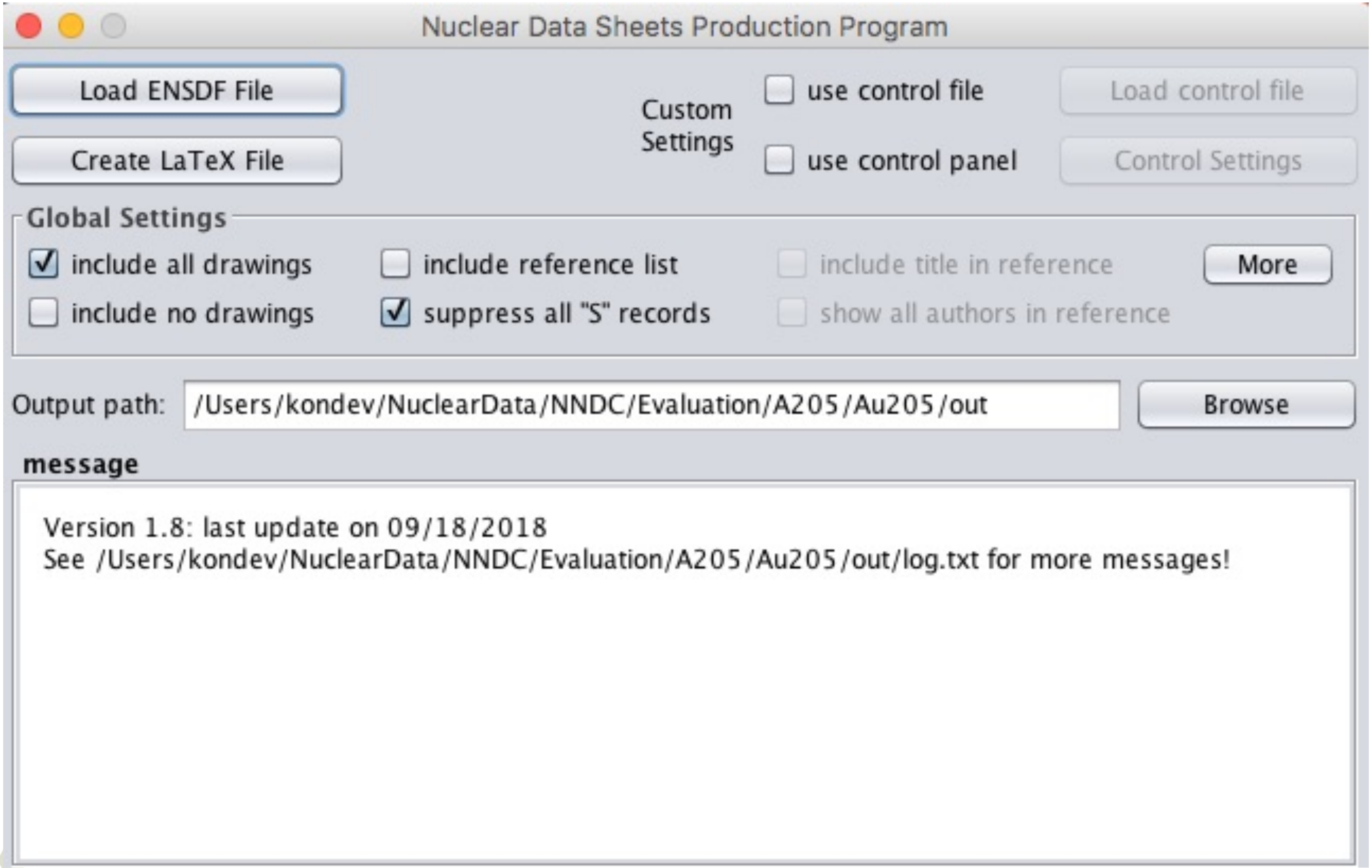
Input file:
(Sample input file: fmtchk.inp)

An ENSDF-formatted file.

Output file:
(Sample output file: fmtchk.rpt)

A report file indicating possible errors or warnings is generated Brief explanations of the fatal error (prefix <F>), error (prefix <E>), warning (prefix <W>), and informational (prefix <I>) messages are listed below.

java-nds


Nuclear Data Sheets Production Program
Load ENSDF File
Create LaTeX File
Custom Settings
use control file
use control panel
Load control file
Control Settings
Global Settings
include all drawings
include no drawings
include reference list
suppress all "S" records
include title in reference
show all authors in reference
More
Output path:
/Users/kondev/NuclearData/NNDC/Evaluation/A205/Au205/out
Browse
message
Version 1.8: last update on 09/18/2018
See /Users/kondev/NuclearData/NNDC/Evaluation/A205/Au205/out/log.txt for more messages!

java-nds: what you need?

--- 8G RAM memory for running the program smoothly for large ENSDF mass-chain files. If RAM is less than 8G, code will be quite slow, for example about 2-3 minutes with 8GB, 10 min with 6 GB, and >15 min for 4 GB.

--- latest version of Java:

JRE 8 or above, which can be downloaded at http://java.com/en/download/

--- LaTeX compiler:

for Windows, **MiKTEX,** free to download at http://miktex.org/download

Download MiKTEX 64-bit (or 32-bit) version appropriate to your computer.

for Linux and MacOS, a LaTex compiler should come with the system.

There is a detailed manual distributed with the program

avetools

developed by T. Kibedi (ANU)

VisualAveragingLibrary (avelib)

developed by M. Birch (McMaster U)

xls2ens

developed by J. Chen (ANL/MSU)

1. **Running requirements**:

   python 2.7.6 (not working with python 3.0.x), xlrd package (version 0.9.4)

2. **The input excel file:**

   This file must at least have two sheets named "**Header**" and "**Data**".

ENSDF Analysis Codes

BrICC

BrICCMixing

gabs

radlist

Why do we need log*ft*, I(β+), EAV, CK, *etc.*?

Log *ft* Systematics (see Jπ guidelines and review by 1998Si17):

• may enable Jπ assignments

(if log *ft* ~3.5, < 5.9, <12.8 or log $f^{1U}t$<8.5, *etc.)*

• may allow us to set useful limit on branch to g.s., *e.g.*,

• TI < … if log*ft* > 5.9 (for a 1st-forbidden branch), or

• TI < … if log$f^{1u}t$ > 8.5 for a 1st-forbidden unique branch

I(β+):

• theoretical capture to positron ratio enables decay scheme normalization when Iβ^+(annihilation) has been measured

EAV, CK, CL…:

• Information needed by RADLST

(comparison of I(x-ray) from RADLIST with measured I(x-ray) can provide a useful check on the decay scheme)

courtesy to C. Baglin

LOGFT

Calculates vital β^- decay and $\beta^+ + \varepsilon$ decay information (allowed, 1U, 2U transitions only).

Input (also uses information on N and P records):

```
 81SR   B            3.0                                                   1U
169YB   E                                               0.88   22
```

1st forbidden unique: col 78-9

Iε+Iβ+: col 65

Iβ-: col 22

Output:

```
81SR    B            3.0              8.5                     1U
81SRS   B EAV=3127 15
169YB   E            0.013 3   0.87 22    8.84   11 0.88   22
169YBS E EAV=409.3 14$CK=0.8147 2$CL=0.13091 4$CM+=0.03
953 1
```

Log *ft*

I(β+)

I(ε)

Warning: output is from allowed calculation for <u>all but</u> 1U and 2U cases!

courtesy to C. Baglin

ruler

calculates g-ray transition probabilities when the level $T_{1/2}$,BR, Iγ and other quantaties (MR, ICC, etc.) are known

have issues when non-symmetric quantities are used

γ-ray decay

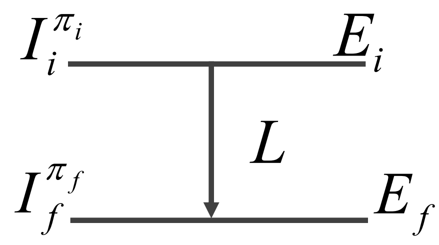
$$|I_i - I_f| \le L \le |I_i + I_f|$$

$$\Delta\pi(EL) = (-1)^L$$

electric multipole

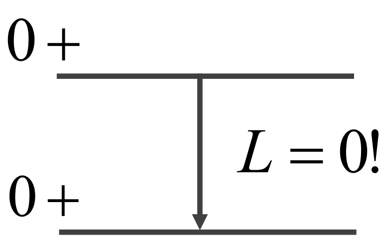
$$\Delta\pi(ML) = (-1)^{L+1}$$

magnetic multipole

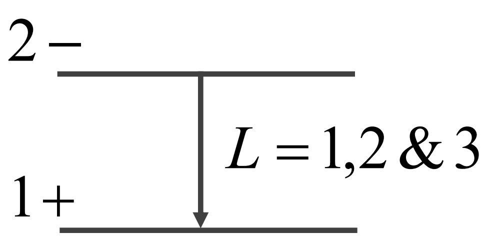


$$E_\gamma = E_i - E_f$$

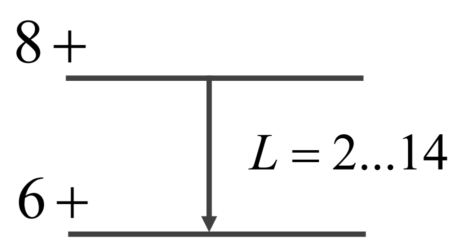
dipole	quadrupole	octupole	hexadecapole	
E1:L=1,yes	E2:L=2,no	E3:L=3,yes	E4:L=4,no	E5:L=5,yes
M1:L=1,no	M2:L=2,yes	M3:L=3,no	M4:L=4,yes	M5:L=5,no



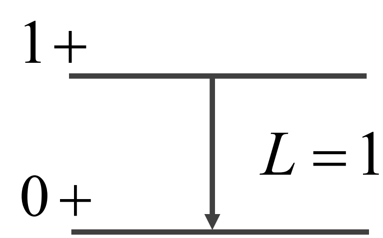
$E0$



$E1(M2, E3)$



$E2(M3, E4...)$

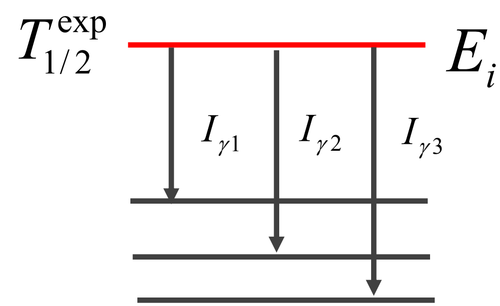


$M1$

Partial lifetime & Transition Probability

$$T_{1/2}^{\gamma} = T_{1/2}^{\exp} / BR_{\gamma} = T_{1/2}^{\exp} \times \frac{\sum I_{\gamma_i} \times (1+\alpha_{Ti})}{I_{\gamma}}$$

partial half-life



$$P_{\gamma}(XL : I_i \rightarrow I_f) = \frac{\ln 2}{T_{1/2}^{\gamma}} = \frac{8\pi(L+1)}{L[(2L+1)!!]^2} \left(\frac{E_{\gamma}}{\hbar c}\right)^{2L+1} B(XL : I_i \rightarrow I_f)$$

Partial γ–ray Transition Probability

Reduced Transition Probability

$$B(XL : I_i \rightarrow I_f) = \frac{\left|\langle I_i | M(XL) | I_f \rangle\right|^2}{2I_i + 1}$$

contains the nuclear structure information

Hindrance Factor in γ-ray decay

$$F_{W(N)} = \frac{B(XL)_{Theory}}{B(XL)_{Exp}} = \frac{T^{\gamma}_{1/2}(XL)_{Exp}}{T^{\gamma}_{1/2}(XL)_{Theory}}$$

… usually an upper limit, but …

Hindrance Factor: Weisskopf (W): based on spherical shell model potential

Nilsson (N): based on deformed Nilsson model potential

EL	$B(EL)_W, e^2 fm^{2L}$	$T^{\gamma}_{1/2}(EL)_W$, sec	ML	$B(ML)_W, \mu_N^2 fm^{2L-2}$	$T^{\gamma}_{1/2}(ML)_W$, sec
E1	$0.06446A^{2/3}$	$6.762A^{-2/3}E_\gamma^{-3}\times10^{-15}$	M1	1.7905	$2.202E_\gamma^{-3}\times10^{-14}$
E2	$0.0594A^{4/3}$	$9.523A^{-4/3}E_\gamma^{-5}\times10^{-9}$	M2	$1.6501A^{2/3}$	$3.100A^{-2/3}E_\gamma^{-5}\times10^{-8}$
E3	$0.0594A^{2}$	$2.044A^{-2}E_\gamma^{-7}\times10^{-2}$	M3	$1.6501A^{4/3}$	$6.655A^{-4/3}E_\gamma^{-7}\times10^{-2}$
E4	$0.06285A^{8/3}$	$6.499A^{-8/3}E_\gamma^{-9}\times10^{4}$	M4	$1.7458A^{2}$	$2.116A^{-2}E_\gamma^{-9}\times10^{5}$
E5	$0.06929A^{10/3}$	$2.893A^{-10/3}E_\gamma^{-11}\times10^{11}$	M5	$1.9247A^{8/3}$	$9.419A^{-8/3}E_\gamma^{-11}\times10^{11}$

What about uncertainties?

http://pythonhosted.org/uncertainties/

uncertainties

Overview User Guide Uncertainties in arrays Technical Guide

```
>>> x = ufloat(0.20, 0.01)  # x = 0.20+/-0.01
```

```
>>> from uncertainties import ufloat_fromstr
>>> x = ufloat_fromstr("0.20+/-0.01")
>>> x = ufloat_fromstr("(2+/-0.1)e-01")  # Factored exponent
>>> x = ufloat_fromstr("0.20(1)")  # Short-hand notation
>>> x = ufloat_fromstr("20(1)e-2")  # Exponent notation
>>> x = ufloat_fromstr(u"0.20±0.01")  # Pretty-print form
>>> x = ufloat_fromstr("0.20")  # Automatic uncertainty of +/-1 on last digit
```