Lawrence Livermore National Laboratory

Introduction to the Generalized Nuclear Data Structure (GND)

Caleb Mattoon

Bret Beck, Eugene Brooks, Dave Brown*, Nidhi Patel and Neil Summers

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

Outline:

- Overview of a GND file:
  - <reactionSuite>
  - <covarianceSuite>
- xData: generalized data containers for GND
- xml schema

The GND format:

- The goal: make one unified structure for all forms of nuclear data: evaluated, MC, deterministic and experimental
- New data hierarchy must be easily human-readable, and representative of underlying physics
- Define a *structure*, xml is one implementation

Latest version of GND now available to CSEWG on GForge and on the Green Data Oasis:
https://ndclx4.bnl.gov/gf/project/gnd

ftp gdo-nuclear.usllnl.org

Sample GND file (n_094_Pu_239 translated from ENDF):

```
<?xml version="1.0" encoding="UTF-8"?>
<reactionSuite projectile="n1" target="Pu239" format="gnd version 1.0"
   temperature="0. K" xmlns:xlink="http://www.w3.org/1999/xlink">
  <styles>
    <style name="evaluated" version="7.5.5" library="ENDF/B-VII"></style>
   </styles>
  <documentation name="endfDoc"><![CDATA[...]]></documentation>
  <particles>...</particles>
  <resonances reconstructCrossSection="true">...</resonances>
  <reaction label="0" outputChannel="n1 + Pu239" date="2006-09-01"
   ENDF_MT="2">...</reaction>
  ...
  <reaction label="38" outputChannel="n1[multiplicity:'2'] + Pu238"
   date="2006-09-01" ENDF_MT="16">...</reaction>
  ...
  <reaction label="45" outputChannel="n1[multiplicity:'energyDependent']
   [firstChance fission]">...</reaction>
  ...
  <reaction label="71" outputChannel="Pu240 + gamma" date="2006-09-01"
   ENDF_MT="102">...</reaction>
</reactionSuite>
```

<particles>:

```
<particles>
  <particle name="gamma" genre="photon" transportable="true" mass="0 amu"/>
  <particle name="n1" genre="nucleus" transportable="true" mass="1.00866491574 amu"/>
  <particle name="Pu239" genre="nucleus" Jpi="1/2" mass="239.052172899498 amu">
    <level name="Pu239_e0" index="0" energy="0 eV"/>
    <level name="Pu239_e1" index="1" energy="7861 eV">
      <gamma energy="7861 eV" finalLevel="Pu239_e0" probability="1.0"/></level>
    <level name="Pu239_e2" index="2" energy="57276 eV"/>
    ...
    <level name="Pu239_e40" index="40" energy="3.909e6 eV"/>
    <level name="Pu239_c" index="c" energy="u:6.34e5 eV"/></particle>
  <particle name="Pu240" genre="nucleus" mass="240.053813545 amu"/></particles>
```

- List particles, with basic level scheme. Contains any particles, levels and gammas that appear in the ENDF file
- In future, may be stored in separate nuclear structure database. See talks by Tim Johnson and Nidhi Patel for more details!

Sample GND file (n_094_Pu_239 translated from ENDF):

```
<?xml version="1.0" encoding="UTF-8"?>
<reactionSuite projectile="n1" target="Pu239" format="gnd version 1.0"
   temperature="0. K" xmlns:xlink="http://www.w3.org/1999/xlink">
  <styles>
    <style name="evaluated" version="7.5.5" library="ENDF/B-VII"></style>
   </styles>
  <documentation name="endfDoc"><![CDATA[...]]></documentation>
  <particles>...</particles>
  <resonances reconstructCrossSection="true">...</resonances>
  <reaction label="0" outputChannel="n1 + Pu239" date="2006-09-01"
   ENDF_MT="2">...</reaction>
  ...
  <reaction label="38" outputChannel="n1[multiplicity:'2'] + Pu238"
   date="2006-09-01" ENDF_MT="16">...</reaction>
  ...
  <reaction label="45" outputChannel="n1[multiplicity:'energyDependent']
   [firstChance fission]">...</reaction>
  ...
  <reaction label="71" outputChannel="Pu240 + gamma" date="2006-09-01"
   ENDF_MT="102">...</reaction>
</reactionSuite>
```

<resonances>:

```
<resonances reconstructCrossSection="true">
  <resolved lowerBound="1e-5 eV" upperBound="1.25e5 eV" nativeData="Reich-Moore">
    <Reich-Moore scatteringRadius="5.365 fm" computeAngularDistribution="true" ...>
      <resonanceParameters>
        <table dimensions="187,6">...</table></resonanceParameters>
    </Reich-Moore>
  </resolved>
  <unresolved lowerBound="1.25e5 eV" upperBound="1e6 eV" nativeData="tabulatedWidths">
    <tabulatedWidths scatteringRadius="5.606283 fm" forSelfShieldingOnly="true" ...>
      <L_section L="0">...</L_section>
  </unresolved>
</resonances>
```

- All ENDF-6 resonance formats are supported
- fudge2.0 includes a new resonance reconstruction tool

Sample GND file (n_094_Pu_239 translated from ENDF):

```
<?xml version="1.0" encoding="UTF-8"?>
<reactionSuite projectile="n1" target="Pu239" format="gnd version 1.0"
   temperature="0. K" xmlns:xlink="http://www.w3.org/1999/xlink">
  <styles>
    <style name="evaluated" version="7.5.5" library="ENDF/B-VII"></style>
   </styles>
  <documentation name="endfDoc"><![CDATA[...]]></documentation>
  <particles>...</particles>
  <resonances reconstructCrossSection="true">...</resonances>
  <reaction label="0" outputChannel="n1 + Pu239" date="2006-09-01"
   ENDF_MT="2">...</reaction>
  ...
  <reaction label="38" outputChannel="n1[multiplicity:'2'] + Pu238"
   date="2006-09-01" ENDF_MT="16">...</reaction>
  ...
  <reaction label="45" outputChannel="n1[multiplicity:'energyDependent']
   [firstChance fission]">...</reaction>
  ...
  <reaction label="71" outputChannel="Pu240 + gamma" date="2006-09-01"
   ENDF_MT="102">...</reaction>
</reactionSuite>
```

<reaction> element:

```
<reaction label="1" outputChannel="n1 + Pu239_e1" date="2006-09-01" ENDF_MT="51">
  <crossSection nativeData="pointwise">
    <pointwise xData="XYs" length="123" accuracy="0.001">
      <axes>...</axes>
      <data>...</data>
    </pointwise></crossSection>
  <outputChannel genre="twoBody" Q="-7861 eV">
    <product name="n1" label="n1" multiplicity="1">
      <distributions nativeData="angular">...</distributions>
      <energyDeposition>...</energyDeposition></product>
    <product name="Pu239_e1" label="Pu239_e1" multiplicity="1">
      <distributions nativeData="none"></distributions>
      <decayChannel>...</decayChannel></product></outputChannel>
</reaction>
```

- <documentation> also allowed inside each reaction
- 'nativeData': permits multiple forms of the same data
- Using <decayChannel>, we can explicitly describe two-step break-up, etc.

Definition of a GND reaction:

- A GND reaction is generally defined by the list of outgoing products:

  n1 + Pu239 → n1 + Pu239 ← this is MT=2

  n1 + Pu239 → n1 + Pu239_e1 ← MT=51

- An extra reaction id can also be added, so evaluators can be much more specific:

  <reaction outputChannel="n1 + Pu239 [shape elastic]">…

  <reaction outputChannel="n1 + Pu239 [compound elastic]">…

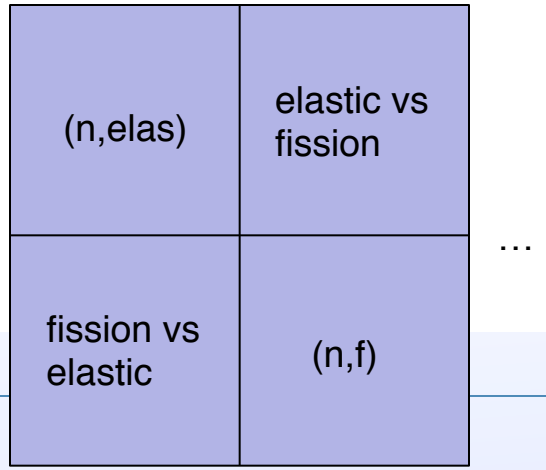
Sample GND file (n_094_Pu_239 translated from ENDF):

```
<?xml version="1.0" encoding="UTF-8"?>
<reactionSuite projectile="n1" target="Pu239" format="gnd version 1.0"
   temperature="0. K" xmlns:xlink="http://www.w3.org/1999/xlink">
  <styles>
    <style name="evaluated" version="7.5.5" library="ENDF/B-VII"></style>
   </styles>
  <documentation name="endfDoc"><![CDATA[...]]></documentation>
  <particles>...</particles>
  <resonances reconstructCrossSection="true">...</resonances>
  <reaction label="0" outputChannel="n1 + Pu239" date="2006-09-01"
   ENDF_MT="2">...</reaction>
  ...
  <reaction label="38" outputChannel="n1[multiplicity:'2'] + Pu238"
   date="2006-09-01" ENDF_MT="16">...</reaction>
  ...
  <reaction label="45" outputChannel="n1[multiplicity:'energyDependent']
   [firstChance fission]">...</reaction>
  ...
  <reaction label="71" outputChannel="Pu240 + gamma" date="2006-09-01"
   ENDF_MT="102">...</reaction>
</reactionSuite>
```

<covarianceSuite>:

```
<?xml version="1.0" encoding="UTF-8"?>
<covarianceSuite projectile="n1" target="Pu239" version="gnd version 1.0”>
  <reactionSums>
    <reactionSum id="total" ENDF_MFMT="33,1">...</reactionSum></reactionSums>
  <section id=”... [total nubar]" nativeData="covarianceMatrix">
    <rowData xlink:type="simple" xlink:href="/reactionSuite/.../multiplicity”/>
    <covarianceMatrix>...</covarianceMatrix></section>
  <section id=”... [delayed nubar]" nativeData="covarianceMatrix">
    <rowData xlink:type="simple" xlink:href="/reactionSuite/.../multiplicity/>
    <covarianceMatrix>...</covarianceMatrix></section>
  ...
</covarianceSuite>
```

- Covariances are stored in a separate file, and use links to associate them to the correct section.
- Each <section> corresponds to one chunk of the ‘full’ covariance matrix:



Current status in GND:

```
<section id="n1[multiplicity:'2'] + Mn54 + gamma" nativeData="covarianceMatrix">
  <rowData xlink:type="simple" xlink:href="/reactionSuite/reaction[@label='41']/crossSection" ENDF_MFMT="33,16"/>
  <covarianceMatrix>
    <axes>
      <axis index="0" label="row_energy_bounds" unit="eV" interpolation="linear,flat" length="5"> 8971600
            1.7901e7 2.8371e7 4.4965e7 6e7</axis>
      <axis index="1" label="column_energy_bounds" unit="eV" interpolation="linear,flat"
           mirror_row_energy_bounds="true"/>
      <axis index="2" label="matrix_elements" unit=""/></axes>
    <matrix dimensions="4,4" type="relative" form="symmetric" precision="6">
       9.009000e-01
       0.000000e+00  2.980270e-02
       0.000000e+00  2.745890e-02  2.757980e-02
       0.000000e+00  2.554860e-02  2.535970e-02  1.231670e-02</matrix></covarianceMatrix></section>
```

- Also translated: ‘summed’ covariances (LB=0), diagonal matrices, rectangular matrices.
- Not currently handled: MF=32, LB=8 and 9, cross-material matrices

xLink elements

- What if we need to link between different data?
- Xlink is used to link xml elements:

```
<distributions nativeData="reference">
  <reference xlink:href="/reactionSuite/reaction[@label='43']/
      outputChannel/product[@label='n1']/distributions"/></distributions>
```

- May also link to external file:

```
<rowData xlink:href="externalFile.gnd#/reactionSuite/
        reaction[@label='0']/crossSection"/>
```

xData: basic GND data containers

- ENDF format includes several general-purpose data containers (TAB1, TAB2, LIST). GND also has general-purpose data storage: xData

- xData must handle:
  - 2d (cross section, multiplicity)
  - 3d (angular and energy distributions)
  - 4d (double differential distributions)
  - piecewise interpolations

- In fudge2, 2-dimensional xData is implemented in c, with a python wrapper

2d xData example:

```
<pointwise xData="XYs" length="123" accuracy="0.001">
  <axes>
    <axis index="0" label="energy_in" unit="eV" interpolation="linear,linear" frame="lab"/>
    <axis index="1" label="crossSection" unit="b" frame="lab"/></axes>
  <data> 7894.169 0 8e3 0.03555347 ... 1.95e7 0.07446407 2e7 0.0731713</data></pointwise>
```

- Could represent cross section, multiplicity, or other 2d data
- ‘accuracy’ attribute: evaluator attests that data can be interpolated with this accuracy
- ‘frame’ required for all axes, ‘interpolation’ required only on the independent axis

Additional comments on xData

- xData elements are meant to be flexible data containers
- However, the format is designed to be more restrictive than ENDF in specific cases:
  - ENDF permits repeated x-values in a single interpolation region:

```
2.505500+4 5.446610+1          0          0          0          02525 3  2     1
0.000000+0 0.000000+0          0          0          1         842525 3  2     2
        84          2                                               2525 3  2     3
1.000000-5 0.000000+0 1.250000+5 0.000000+0 1.250000+5 5.729550+02525 3  2     4
```

  - In GND, these must be split into multiple regions

XML schema for GND

- Defines the structure of a GND file, and may be used to validate that GND files are properly formatted

- Validation is simple, may be done using online schema:

```
>xmllint -schema http://.../gnd.xsd gnd_sample.xml
```

- Validating catches many format errors:
  - Missing elements or attributes
  - Extra, unrecognized elements or attributes
  - Invalid data in text fields:

    <data>1.0e-5 3.8e+3 2.5e-2 2.3 e+3 …</data>
  - Incorrect type in attributes (missing units, etc)

Conclusions:

- First full release of GND is available! Features:
  - simple hierarchy for storing nuclear reaction data, both in python classes and in xml
  - visualization tools (require gnuplot and/or matplotlib)
  - documentation, xml schema, sample files

- Available via:

  >ftp gdo-nuclear.uclnl.org

  https://ndclx4.bnl.gov/gf

Conclusions:

- Easy to use!

```
To convert an ENDF file to the new format, just do:
>fudge/bin/rePrint.py n-094_Pu_239.endf

or, to convert a random ENDF file and compare results, do:
>fudge/bin/rePrintSample.py

Either way, GND files (in xml) are created, plus the results
are translated back to ENDF
```

- Please try it out and give us feedback!

- Extra slides:

2d piecewise xData:

```
<piecewise xData="regionsXYs">
  <axes>
    <axis index="0" label="energy_in" unit="eV" interpolation="byRegion,byRegion" .../>
    <axis index="1" label="crossSection" unit="b" frame="lab"/></axes>
  <region index="0" length="2" accuracy="0.001">
    <interpolationAxes index="0" interpolation="log,log"/>
    <data> 1e-5 3.4 1e3 0</data></region>
  <region index="1" length="12" accuracy="0.001">
    <interpolationAxes index="0" interpolation="linear,linear"/>
    <data> 1e3 0 1050 2.3 ... 1950 0 2e3 0</data></region></piecewise>
```

- Very similar to pointwise data, with addition of the ‘interpolationAxes’ element

3d xData example:

```
<angular nativeData="LegendrePointwise">
  <LegendrePointwise xData="W_XYs_LegendreSeries">
    <axes>
      <axis index="0" label="energy_in" unit="eV" interpolation="linear,linear" frame="lab"/>
      <axis index="1" label="mu" unit="" interpolation="linear,linear" frame="centerOfMass"/>
      <axis index="2" label="P(energy_in|mu)" unit="" frame="centerOfMass"/></axes>
    <energy_in value="1e-05" index="0" length="3"> 1 0 0</energy_in>
    <energy_in value="1000.0" index="1" length="5"> 1 1.746e-3 ... -7.352e-7</energy_in>
    ...
    <energy_in value="18000000.0" index="31" length="21"> 1 0.9313 ... 8.89e-4</energy_in>
    <energy_in value="20000000.0" index="32" length="21"> 1 0.9421 ... 2.493e-3</energy_in>
  </LegendrePointwise>
</angular>
```

- ‘W_XYs’: a list of XY values for each W (incident energy)
- Again, ‘interpolation’ required only on the independent axes

<table> element (used within resonances):

```
<table dimensions="187,6">
  <columns>
    <column index="0" name="energy" units="eV"/>
    <column index="1" name="L"/>
    <column index="2" name="J"/>
    <column index="3" name="channelSpin"/>
    <column index="4" name="neutronWidth" units="eV"/>
    <column index="5" name="captureWidth" units="eV"/></columns>
  <data>
<!--  energy | L |   J | channelSpin | neutronWidth | captureWidth  -->
    -110000.0   0   2.0           2.0       26286.24            1.0
    -100000.0   0   3.0           3.0       65984.88            1.0
    -759.8076   0   2.0           2.0       33.22168      0.4531165
     340.7875   0   2.0           2.0       23.67655      0.4096677
     ...
  </data></table>
```

- XML comments (in green) help out with readability

Status of ENDF-β5 conversion:

- Bulk of ENDF-BVII.1-β5 can be converted. Some sections aren't yet translated:
  - MF32 and 34 covariances
  - MT460 (delayed gamma production)
  - Redundant cross sections (ie, if MT51-91 present, MT4 is ignored)
  - Total (MT1): just store the cross section?

- All but one file from ENDF neutron sub-library can be translated (Ni59 has unusual interpolation, not yet implemented)

Lawrence Livermore National Laboratory

Future plans for GND

Caleb Mattoon

Bret Beck, Eugene Brooks, Dave Brown*, Nidhi Patel and Neil Summers

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

GND version 1.0 is now available. What's next?

- Additions to the format
- Access routines
- Improved checking and visualization
- Documentation, Evaluator toolkit, …?

Format additions:

- Some of ENDF still must be translated:
  - MF32 and MF34 covariances
  - Delayed gammas (MT460)
  - Total (MT=1), and any other reactions that don't specify product info
- Optional elements in GND: energy/momentum deposition, thermal scattering and S(α,β), transfer matrices…
- Please give us format suggestions soon!

Access routines:

- Structure of GND should be stable soon
- We are starting on access routines (in c) to allow LLNL users to start testing GND
- Also need a wrapper, to allow old LLNL codes to still use old access routines
- API (the ‘Generalized Interaction Data Interface’ GIDI) will also be released

Improved checking and visualization:

- Fudge+GND offers a big advantage to evaluators: checking and visualization are built in!

- 2-d plotting already works, 3-d and 4-d on their way. Evaluator will be able to visually inspect their file right away!

- Checking/fixing codes will be ported from legacy fudge. Simple to use:

```
>from fudge.gnd import reactionSuite
>r = reactionSuite.readXml("newEvaluation.xml")
>warnings = r.check()
# and, possibly:
>r.fix()
```

Other work:

- Documentation must be brought up-to-date.
- ‘Evaluator API’, to simplify creating a new GND evaluation.
- Processing:
  - Processing of ENDF outgoing particle data types (evaporation models, etc.) nearly complete. c++ code needs to be hooked back into fudge
  - For MC codes, grouping of data should be done by access routines (allows user to choose any group structure)