GNDS resonance format proposals

November 10, 2020: EG-GNDS

C.M. Mattoon
Nuclear Data and Theory group, PLS/NACS

LLNL-PRES-816458
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Two competing format proposals seek to simplify background xsc in resonance region.

- <resonancesWithBackground> node appears as a form inside <crossSection>. Full cross section is obtained by reconstructing resonances and adding to background.
- Proposals are incompatible – we need to pick one
- Neither proposal changes contents of <resonances> node, but future format proposal will likely expand resonance options

resonancesWithBackground in GNDS-1.9:

Critique #1: lots of nesting

Critique #2: fastRegion is not a background cross section

```
<crossSection>
  <resonancesWithBackground label="eval">
    <resonances href="/reactionSuite/resonances"/>
    <background>
      <resolvedRegion>
        <XYs1d>...</XYs1d>
      </resolvedRegion>
      <unresolvedRegion>
        <XYs1d>...</XYs1d>
      </unresolvedRegion>
      <fastRegion>
        <XYs1d interpolation="lin-log">...</XYs1d>
      </fastRegion>
    </background>
    <uncertainty>
      <covariance href="$covariances#..."/>
    </uncertainty>
  </resonancesWithBackground>
```

Note: resolved and unresolved are optional (parameters may have been provided only for self-shielding)

Proposal #1 moves fastRegion up one level

- <fastRegion> moves up from <background> to <resonancesWithBackground>

```
<crossSection>
  <resonancesWithBackground label="eval">
    <resonances href="/reactionSuite/resonances"/>
    <background>
      <resolvedRegion>
        <XYs1d>...</XYs1d>
      </resolvedRegion>
      <unresolvedRegion>
        <XYs1d>...</XYs1d>
      </unresolvedRegion>
    </background>
    <fastRegion>
      <XYs1d interpolation="lin-log">...</XYs1d>
    </fastRegion>
    <uncertainty>
      <covariance href="$covariances#..."/>
    </uncertainty>
  </resonancesWithBackground>
```

New critique: not as clear what <uncertainty> node relates to.
Just fastRegion? Just background? Both?

Proposal #2 does away with <background> in favor of <regions1d> or <XYs1d>

- Relationship to <uncertainty> is more clear
- No explicit markup for resolved, unresolved, fast regions

```
<crossSection>
  <resonancesWithBackground label="eval">
    <resonances href="/reactionSuite/resonances"/>
    <regions1d>
      <axes>...</axes>
      <XYs1d index="0">...</XYs1d>
      <XYs1d index="1">...</XYs1d>
      <XYs1d index="2">...</XYs1d>
      ...
    </regions1d>
    <uncertainty>
      <covariance href="$covariances#..."/>
    </uncertainty>
  </resonancesWithBackground>
```

Some thoughts…

- Proposal #1 is more explicit and enforces having clear boundaries between resolved/unresolved/fast regions in the background cross section.
  - Note that non-zero background cross sections are less common in modern evaluations

- Proposal #2 is simpler, only requires previously defined basic data containers

Discussion

Anticipated format proposals for the next GNDS update (2.0 and after)

November 10, 2020: EG-GNDS

C.M. Mattoon
Nuclear Data and Theory group, PLS/NACS

LLNL-PRES-816458
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

GNDS-2.0 already includes several revisions, but other ideas are still under development

- This is my current list of GNDS ideas that I expect/hope will soon become format proposals
- List is based on a combination of in-person discussion and open issues in gitlab
  - If you have an idea (even if not yet fully-formed), please turn it into an issue tracker on gitlab so we can better keep track of it!
- Some of these may still make it into v2.0, others are likely further off

Some cleanup of specifications doc, should be possible before v2.0

Make better use of enumerations in specifications (and enumerate more options)

- Along with a few other enumerated types, the allowed list of ‘interpolation’ values is defined in section 3.5.

Type name: `interpolation`
Allowed values: [‘flat’, ‘charged-particle’, ‘lin-lin’, ‘lin-log’, ‘log-lin’, ‘log-log’]

- Later in the document the ‘interpolation’ appears as an attribute on functional containers but does not link back to section 3.5. We should link enumerated types to their definition

Correct links!

- Some links in the specifications document point to the wrong spot.
- Example: the <energy> node inside an uncorrelated distribution should point to outgoing energy distributions but instead points to the excited state energy in PoPs
- Fix is to use namespace-aware code to generate links

Format changes that could be complete for v2.0

Remove ‘index’ attribute from the <axis> (or make it optional)?

- Current layout:

```
<axes>
  <axis index="2" label="energy_in" unit="eV"/>
  <axis index="1" label="mu" unit=""/>
  <axis index="0" label="P(mu|energy_in)" unit=""/></axes>
```

- Index allows making a unique link to a particular axis… but the labels are also unique and can be used to construct a link
- Order of <axis> indices has been source of confusion
- If we go this route, the axis ‘label’ should be required (currently optional according to specifications)

Changes to <values> node to support hybrid ascii/binary storage

- Problem: ascii forms like XML and JSON are slow to load, mostly due to cost of converting *lots* of ascii strings to doubles
- Binary forms like HDF5 can load faster (and reduce disk space), but not if the full GNDS hierarchy is translated
- Proposed solution: store hierarchy in XML/JSON, values in HDF5. Add optional href to <values> node to connect the two:

```
<values> 3.48172 0.172839 4.21334 0.82731 … </values>
  could become
<values href="HDF#/array1"/>
   or
<values href="HDF#/array1" indices="592,198072"/>
```

Reasons for looking at hybrid storage:

- LLNL has processed ENDF/B-VIII GNDS files with both Monte Carlo and deterministic data at 23 temperatures.
- Files are quite large:
  - n + Fe56: 1.3 Gb in XML, 1.2 Gb in pure HDF
  - n + U235: 1.6 Gb in XML, 1.8 Gb in pure HDF
- Hybrid storage with numeric data in HDF does better:
  - n + Fe56: 450 Mb
  - n + U235: 545 Mb
- Still working on optimizing load times, but currently the hybrid method is fastest to load

Add template axes section

- Units and axes are repeated frequently in GNDS
- Could define each type of <axes> in a central place and link to those definitions from elsewhere
- For example,

```
<templateAxes>
  <axes label="crossSection">
    <axis index="1" label="energy_in" unit="eV"/>
    <axis index="0" label="cross section" unit="b"/>
  </axes>
  <axes label="multiplicity">...</axes>
  <axes label="angularDistribution">...</axes>
  ...
</templateAxes
```

Changes that are likely post-2.0

Resolved resonance region

- Frequent discussions between resonance region evaluators. Wish list (courtesy of I. Thompson) includes
  - Allowing for zero damping factors
  - Relativistic kinematics
  - Brune parameterization
  - Charged-particle elastic scattering
  - Level-matrix method

- Bulk of the work here will be in processing codes, but will require (at least) adding some flags to GNDS to indicate various options

Thermal Neutron Scattering Law

- Currently handle coherent *and*/*or* incoherent elastic, plus incoherent inelastic
  - Evaluations with both coherent and incoherent inelastic (per proposal from M. Zerkle) is simple, requires no specifications change

- Relaxing incoherent approximation for inelastic?
  - On NCSU wish list, likely requires GNDS extension

- What about LEAPR input (or equivalent for other models)?
  - One option in GNDS-2.0 is to store these in *documentation/computerCode/inputDecks* nodes

  - Or we could make TNSL-specific markup for these model inputs

Expand PoPs to support more ENSDF-like data types?

- Adam Hayes gave an overview of the ENSDF modernization effort.
- That effort will likely expand particle data well beyond current capabilities in PoPs
- Our task: revisit PoPs to ensure it can handle all nuclear structure data relevant to GNDS evaluations
  - Don't necessarily need to handle all of ENSDF, but likely need to expand beyond current capabilities

Others?