Cross Section Covariances at LNLL: Tools and Codes

Kyle Wendt

November 30, 2020

LLNL-PRES-817121
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

FUDGE and Evaluated Means and Uncertainties (EMU)

- **FUDGE** is our workhorse for reading and manipulating GNDS files.
  - Robust suite of tools for working with cross sections, distributions, etc
  - Limited support for reading and writing GNDS covariance files.
    - Used to crash on some correctly formatted files derived from ENDF.
  - Little to no support for working with covariances beyond accessing individual channel.
  - Support recently added for converting ENDL extended covariances to GNDS

- **EMU** is our new toolkit for working with GNDS formatted covariances.

EMU is our new toolkit for working with covariances.

- Original goal was a GNDS aware replacement for KIWI.
  - KIWI only support native LLNL ENDL format
  - KIWI had limited treatment of physics constraints (i.e., summed channels)
  - KIWI did not support cross channel covariances
- EMU is built on top of FUDGE for reading and writing GNDS files
  - Library agnostic
  - all UQ and covariance logic is contained in EMU (for time being)
  - partial rewrite of FUDGE covariance suites IO to fully support ENDF
- Alpha version recently released to limited testing group.
  - Planned to be released as opensource in near (~1 year) future.
- EMU will be the eventual home of most of our covariance infrastructure.

User facing EMU currently focuses on sampling

Current pre-release UX is focused on supporting users performing MCMC style UQ in three forms: script interface, CLI, and structured map format

Map file generation ensures that (GIDI) users are always reading in the correct set of sampled GNDS files, no need for user to keep track of all file locations.

This will enable reproducible UQ studies.

```
<emu format="0.1">
    <map path="$DATA_ROOT/ENDF-VIII/all.map" />
    <variations destination="$DATA_ROOT/generated/{library}/variations 1">
        <!-- Everything in a variations block will be varied together. -->
        <protare projectile="n" target="Cr50"/>
        <protare projectile="n" target=”O16"/>
    </variations>
</emu>
```

```
<map library="ENDFB-VIII.0/variations 1/0001" format="0.1">
    <protare projectile="n"  target="Cr50". evaluation="ENDF/B-8.0”
path="$DATA_ROOT/generated/ENDFB-VIII.0/variations 1/neutrons/n-
024_Cr_050.var0001.xml"/>
    <protare projectile="n"  target="O16"  evaluation="ENDF/B-8.0"
path="$DATA_ROOT/generated/ENDFB-VIII.0/variations 1/neutrons/n-
008_O_016.var0001.xml"/>
    <import path="$DATA_ROOT/ENDF-VIII/all.map"/>
</map>
```

Our current covariance testing toolset is limited

- FUDGE
  - Only validates that the covariance file is formatted correctly
- EMU
  - Has a small suite of sanity checks on structure of covariance suites (not in alpha release)
    - Domain checks, relative vs absolute, filtering of negative eigenvalues (full cross-channel covariance matrix or each channel individually), existence of “summed loops”, etc
  - More in-depth tools will be implemented/integrated once immediate user needs are sated.
- Various custom scripts from various developers and users

Most tests are essentially visualization checks against EXFOR data

- Many disparate user-made tools to examine diagonal elements
- Rough “intuition” checks for cross channels (i.e. elastic vs inelastic)

There are many planned/needed expansions for testing

- Expansion of work at LLNL to use integral benchmarks as a testbed for covariances (R. Casperson)
  - Forward UQ problem can give a signal if your means and covariances are reasonable
  - Inverse UQ problem can help highlight where you need to improve evaluations
  - How to track/combine different derived covariances?
- Looking for a statistical test of some form
  - Most naïve choices are great for comparing models, but not for rating a single “model”
  - Gaussian Processes may give some insight on what to do.
- How do we build a metric that is robust to bad/no measured data?
  - Generative machine learning may provide a pathway (backup slides if time)
- We are adding tools to EMU to help facilitate work in these directions.
- Stronger focus on how to develop better covariances for future evaluations.

What to do when *useful* covariances are missing?

- Find and yell at the evaluator:
  - “Why is this covariance such #%##$&%&^”

- Request/fund a re-evaluation with a full quantification of uncertainties and systematic errors
  - Expensive without investments in automating the evaluation process.
- Construct simple “ad hoc” covariances based on
  - Differences between existing evaluation libraries.
  - Comparison of mean values with spreads of experimental data
  - Model-dependence between channels
    - Clone covariance pattern in library for neighbors in this nuclear region.
    - For example, elastic and inelastic are commonly anti-correlated.
- Use low-fidelity (“Low-Fi”) covariances described by Little *et al* (2008):
  - Can also take from TENDL
- Use “machine learning” like approaches to generate covariances

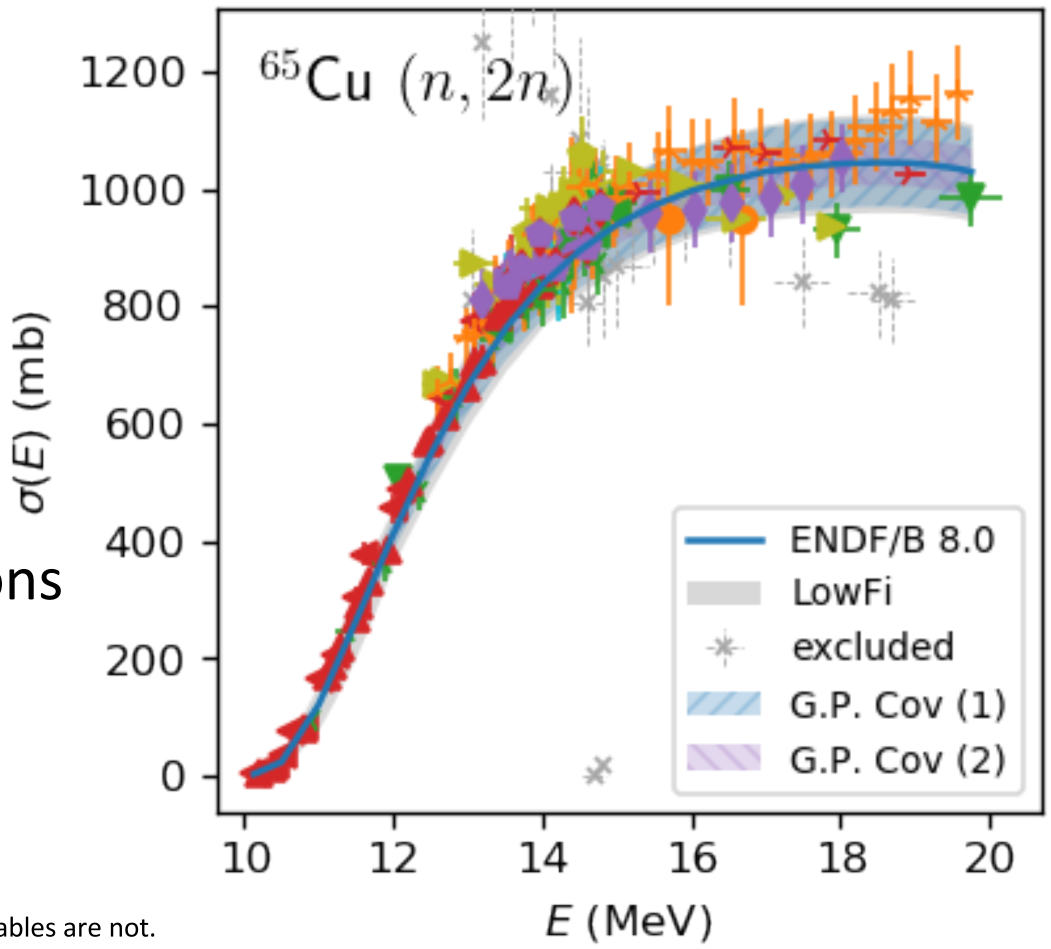
Towards data driven "Low-Fi" covariances

Abstractly, an evaluation with a covariance matrix represents a way to sample a set of (nearly) continuous functions that are distributed pointwise as a multivariate Gaussian*

— **i.e. A Gaussian Process**

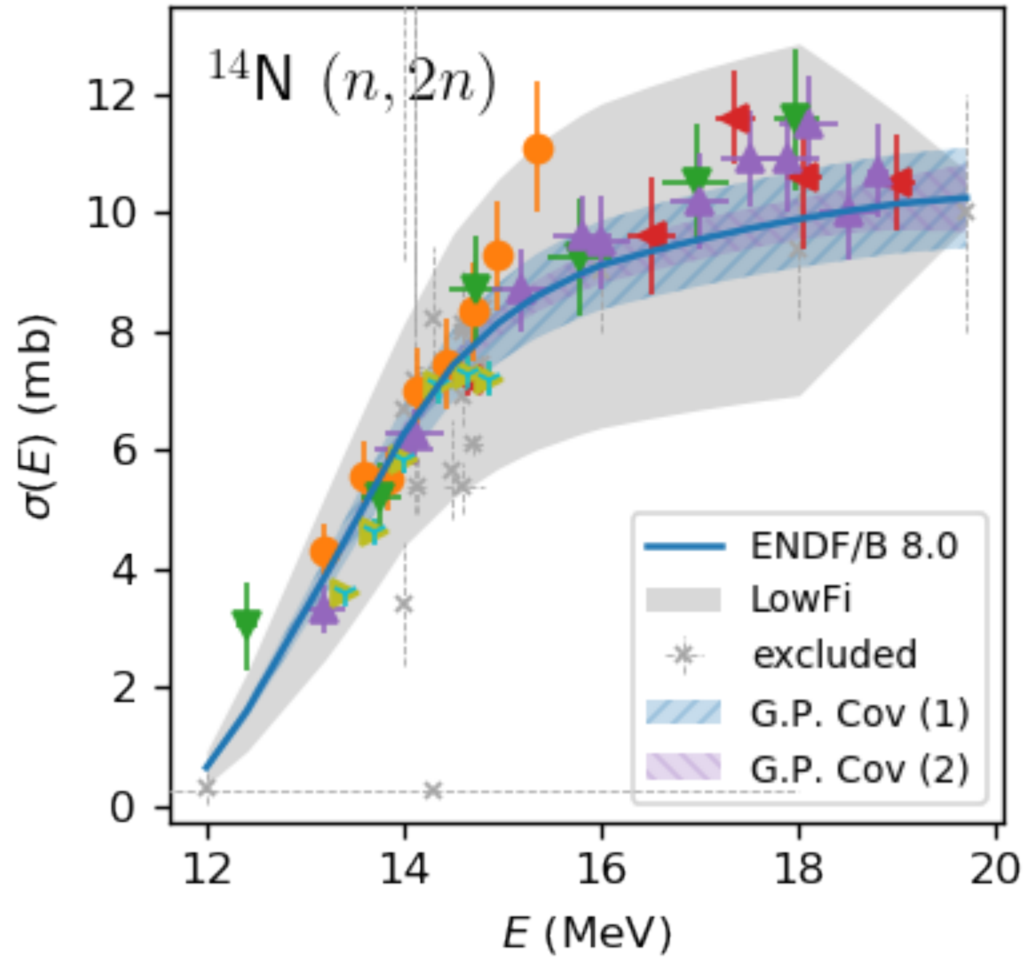
$$F(x) \sim GP\big(\mu(x), K(x', x)\big)$$

- $\mu(x)$ is the average or mean function
  - The evaluation + interpolation rules
- $K(x', x)$ is the covariance kernel of the functions
  - The covariance suite + interpolation rules
- $\mu(x), K(x', x)$ are often parameterized



*This is only approximately true, physical cross sections and distributions are positive definite, Gaussian stochastic variables are not.

What does it mean to be data driven?



$$K(x', x) = \sigma^2 f(x) f(x') e^{-\frac{(x-x')^2}{2l^2}}$$

- σ and l are hyperparameters trained on data.

$$\log p(y_d|M) = -\frac{1}{2}\ \tilde{y}_d \widetilde{K}^{-1} \tilde{y}_d - \frac{1}{2}\log|\widetilde{K}| + C$$

$$\widetilde{K} = K + \boldsymbol{\Sigma_d},\ \tilde{y}_d = \boldsymbol{y_d} - \mu(\boldsymbol{x_d})$$

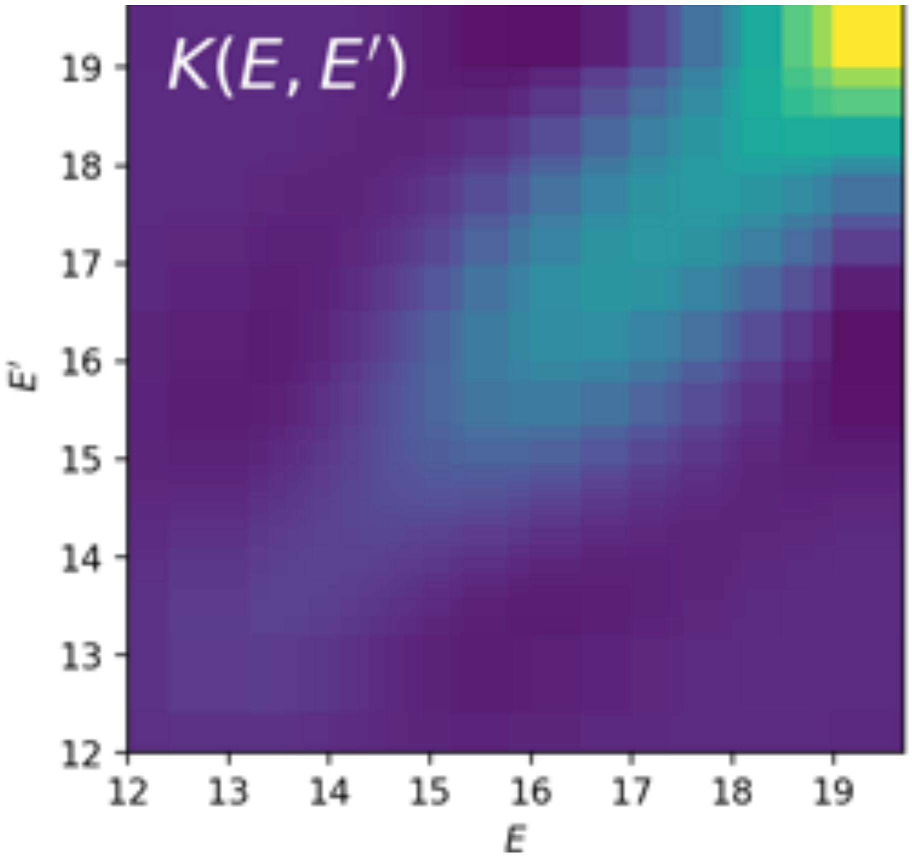
- Predicted effective mean and covariance are determined by MVN conditioned on training data

$$\bar{\mu}(x) = \mu(x) + K(x, \boldsymbol{x_d}) \widetilde{K}(\boldsymbol{x_d}, \boldsymbol{x'_d})^{-1} (\boldsymbol{y_d} - \mu(x_d))$$

$$\overline{K}(x, x') = K(x, x')\ -\ K(x, \boldsymbol{x_d}) \widetilde{K}(\boldsymbol{x_d}, \boldsymbol{x'_d})^{-1} K(\boldsymbol{x_d}, x)$$

The surrogate covariance function is determined fully by data (x_d, y_d) and the predicted mean of the evaluation ($\mu(x)$), instead of parameter variations.

What does it mean to be data driven?



$$K(x', x) = \sigma^2 f(x) f(x') e^{-\frac{(x-x')^2}{2l^2}}$$

- σ and l are hyperparameters trained on data.

$$\log p(y_d|M) = -\frac{1}{2}\ \tilde{y}_d \widetilde{K}^{-1} \tilde{y}_d - \frac{1}{2}\log|\widetilde{K}| + C$$

$$\widetilde{K} = K + \boldsymbol{\Sigma_d},\ \ \tilde{y}_d = \boldsymbol{y_d} - \mu(\boldsymbol{x_d})$$

- Predicted effective mean and covariance are determined by MVN conditioned on training data

$$\bar{\mu}(x) = \mu(x) + K(x, \boldsymbol{x_d}) \widetilde{K}(\boldsymbol{x_d}, \boldsymbol{x'_d})^{-1} (\boldsymbol{y_d} - \mu(x_d))$$

$$\overline{K}(x, x') = K(x, x') \ - \ K(x, \boldsymbol{x_d}) \widetilde{K}(\boldsymbol{x_d}, \boldsymbol{x'_d})^{-1} K(\boldsymbol{x_d}, x)$$

The surrogate covariance function is determined fully by data (x_d, y_d) and the predicted mean of the evaluation ($\mu(x)$), instead of parameter variations.

Questions and conclusions (and more questions)

1. What codes are available at LLNL
   - Various scattered tool and script for working with covariances
   - EMU will be our future workhorse for working with covariances
2. What are you testing?
   - Integral benchmarks and sanity checks
3. What do you do with questionable covariances or filling in the gaps?
   - GPR gives us a recipe to fill in gaps. (soon to be integrated into EMU)
4. How much time would you need for beta-testing of covariances with current codes?
   - Integral benchmark testing takes few months.
5. What developments are needed for testing that satisfies your users?

Questions and conclusions (and more questions)

5. What developments are needed for testing that satisfies your users?
   - Enforcement of consistently across benchmarks
     - e.g., latest ENDF moves spectral indices about ~4 sigma from where they should be
   - Better treatment of group invariant covariances
     - Short range self scaling terms are only well behaved over a narrow range of group sizes and do not make much sense mathematically. Further they do not make sense in applications.
   - How to correctly handle negative cross section realizations?
     - Artifact of treating nuclear uncertainties as Gaussian.
     - Summed channels are especially suspectable to this (elastic = total - rest); always rejects!
   - All data and all data massaging for each evaluation and standard should be included in the published libraries.
     - Without this, most statistical tests do not make sense.
   - What should we do when there is no data?

Lawrence Livermore
National Laboratory