Lawrence Livermore National Laboratory

Progress with Processing GND

Caleb Mattoon,
Bret Beck and Gerry Hedstrom

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

Outline:

- Resonance reconstruction: comparison with RECENT
- Processing for deterministic transport
  - Comparing results with NJOY
- Processing for Monte Carlo transport
- Storing and sharing processed data with GND
- Future work

‘fudgeReconstructResonances’

- Now built into Fudge, reconstructs cross sections from res. parameters. Includes multiprocessing for speed
- Supports ENDF LRF 1,2,3,7 for RRR; LRF 1,2 for URR. No support yet for reconstructing angular distributions
- Building R-Matrix elements (for LRF 3 and 7) written in C for better performance
- How to use:

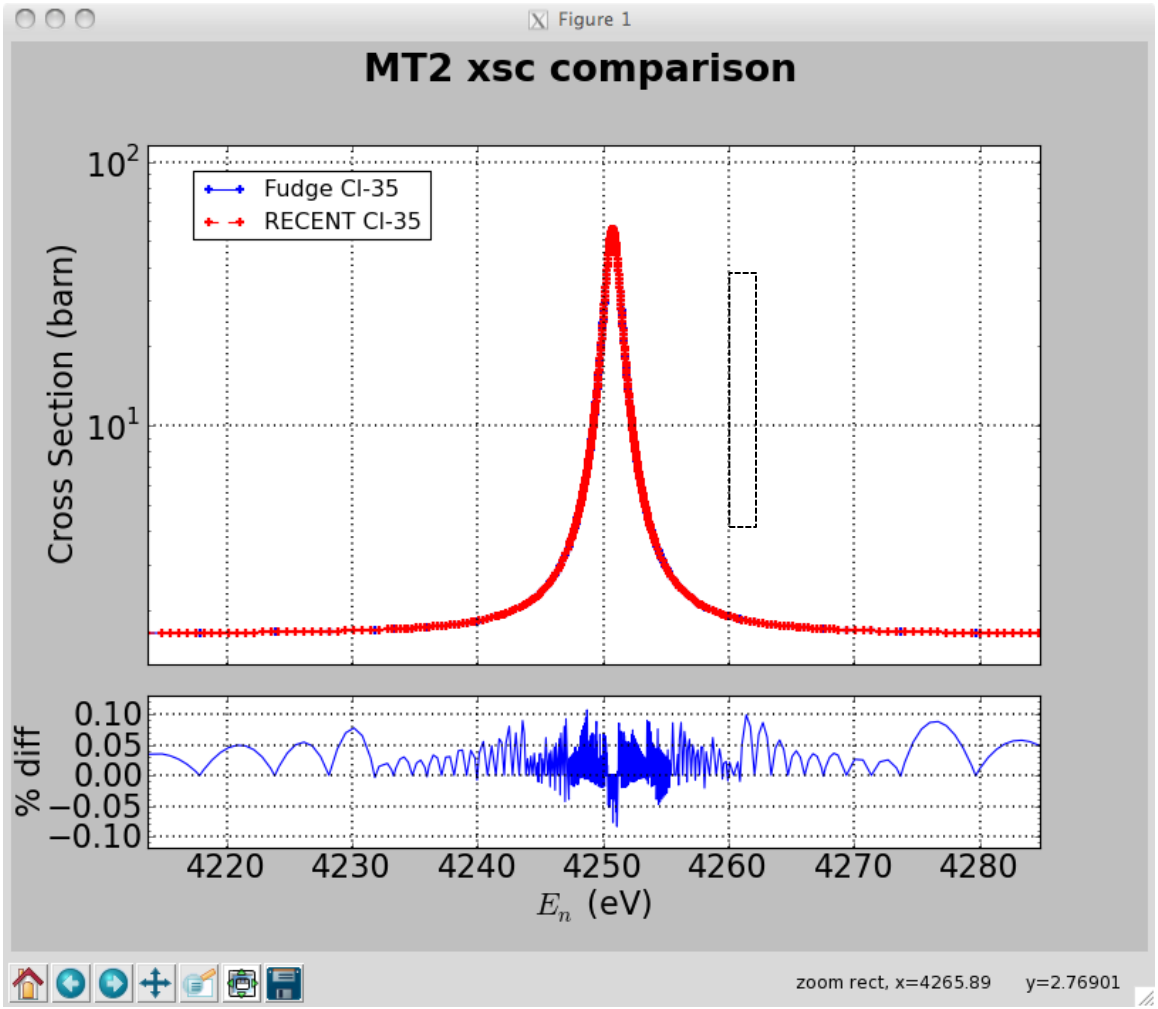
```
>>> from fudge.gnd import reactionSuite
>>> r = reactionSuite.readXML( “neutrons/n-092_U_238.xml” )
>>> r.reconstructResonances( accuracy=0.001, verbose=True )
```

Comparing results to RECENT

- Test: reconstruct same file with both RECENT and fudgeReconstructResonances, to same accuracy, compare results

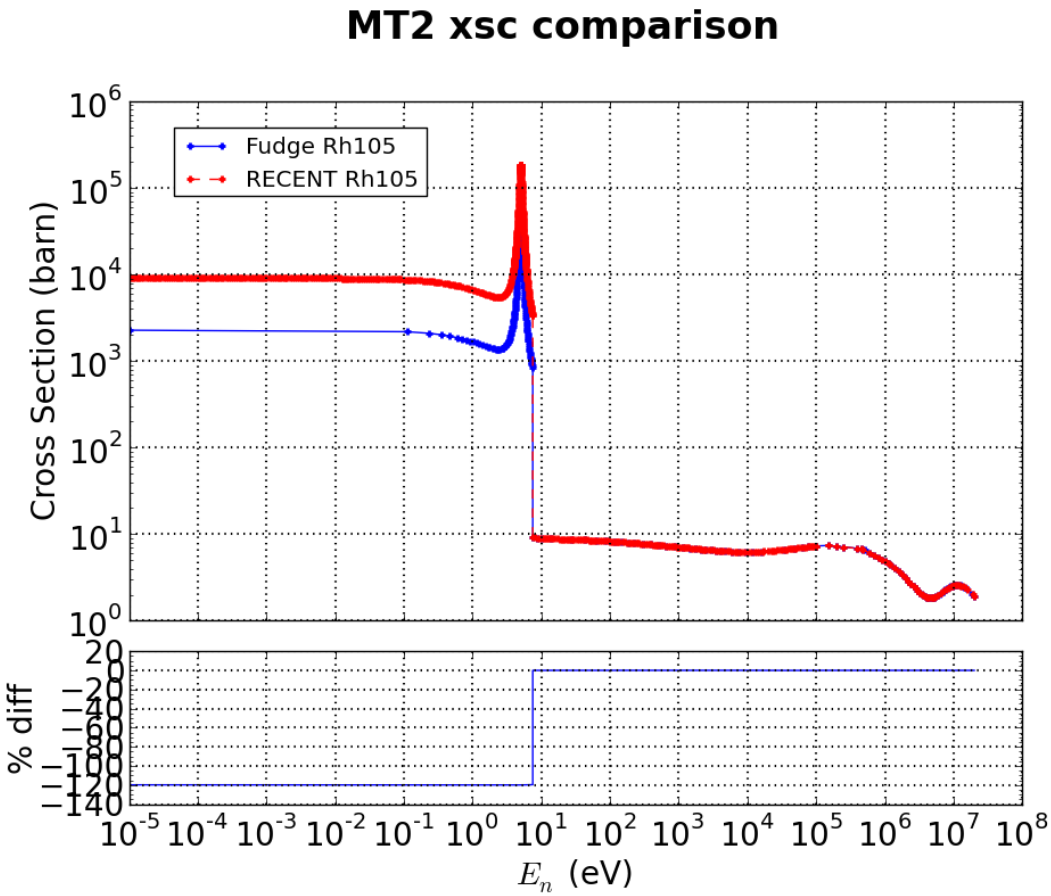
- New tool compareCrossSections (similar to COMPLOT) helps with comparison

Comparing results to RECENT



Comparing results to RECENT

- Trouble in Rh-105
- Cause: conflicting spin assignments in the evaluation!



Comparing results to RECENT

- Rh-105 evaluation originally taken from JENDL-3.3. Spin assignments for RRR and URR differ:

```
 4.510500+4  1.040000+2          0          0          1          04531 2151
 4.510500+4  1.000000+0          0          0          2          04531 2151
 1.000000-5  7.500000+0          1          1          0          04531 2151
 5.000000-1  6.200000-1          0          0          1          04531 2151
 1.040050+2  0.000000+0          0          0         12          24531 2151
-5.000000+0  1.000000+0  1.610000+0  1.450000+0  1.600000-1  0.000000+04531 2151
 5.000000+0  1.000000+0  4.900000-1  3.300000-1  1.600000-1  0.000000+04531 2151
 7.500000+0  1.000000+5          2          2          0          04531 2151
 3.500000+0  6.207500-1          0          0          3          04531 2151
 1.040050+2  0.000000+0          0          0          2          04531 2151
 3.000000+0  0.000000+0          5          0        174         284531 2151
 0.000000+0  0.000000+0  0.000000+0  1.000000+0  0.000000+0  0.000000+04531 2151
 7.500000+0  3.410700+1  0.000000+0  1.500700-3  1.500000-1  0.000000+04531 2151
```

- ENSDF agrees with 7/2 assignment
- Evaluation re-done in JENDL-4, with yet another spin assignment (3/2). Should we adopt that?

Other resonance region issues

- Common problem in evaluations: invalid interpolations!
- Sample from U236, trying to use log-log interpolation in URR:

# energy	competitive	fission
1500.0	0.0	0.00034
		...
45000.0	0.0	0.00034
50000.0	2.03e-05	0.00034
60000.0	0.000333	0.00033
		...

- Other stuff?

Processing for deterministic transport

- Boltzmann equation:

$$\frac{1}{v}\partial_t\phi(E,\Omega)+\Omega\cdot\Delta\phi(E,\Omega)+\rho\sigma_t\phi(E,\Omega)=\frac{\rho}{4\pi}\int_{\Omega'}d\Omega'\int_0^\infty dE'K(E,\Omega\cdot\Omega'|E')\phi(E',\Omega')$$

- Kernel ‘K’: probability that incident particle with incident energy and direction E’,Ω’ produces outgoing particle at E,Ω

- For each Legendre order, produce transfer matrix:

$$J_{gh,\ell,r}=\frac{\int_{\varepsilon'_g}dE'\sigma_r(E')M_r(E')\phi_\ell(E')\int_{\varepsilon_h}dE\int_\mu d\mu P_\ell(\mu)\pi_r(E,\mu|E')}{\int_{\varepsilon'_g}dE'\phi_\ell(E')}$$

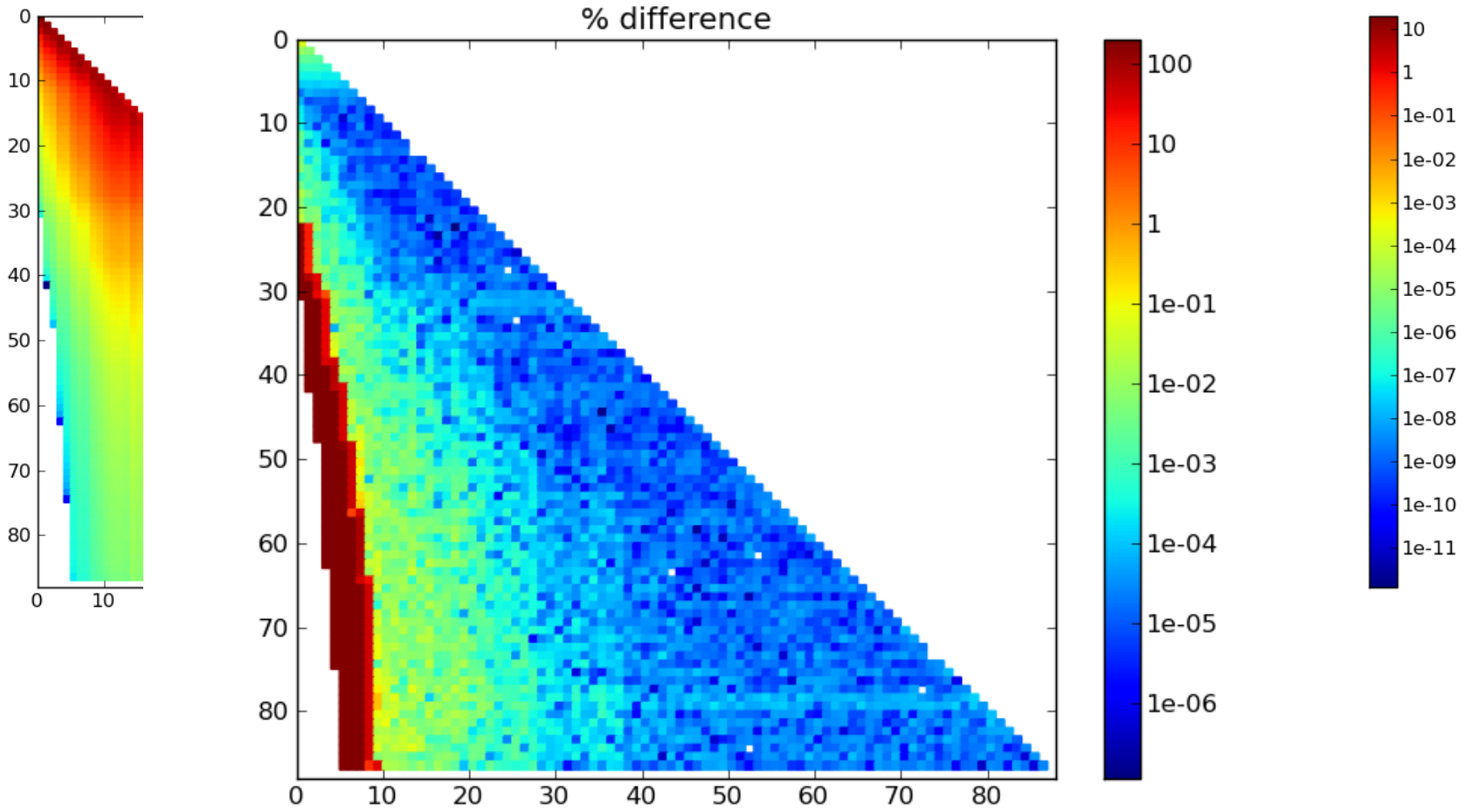
Generating transfer matrices from GND:

- LLNL's new code *get_transfer*: produce transfer matrices from nearly any distribution supported by ENDF-6*
  - c++ code, primarily written by G.Hedstrom
  - integrated into Fudge, permits
- Now testing *get_transfer* by comparing to results from NJOY's GROUPR module

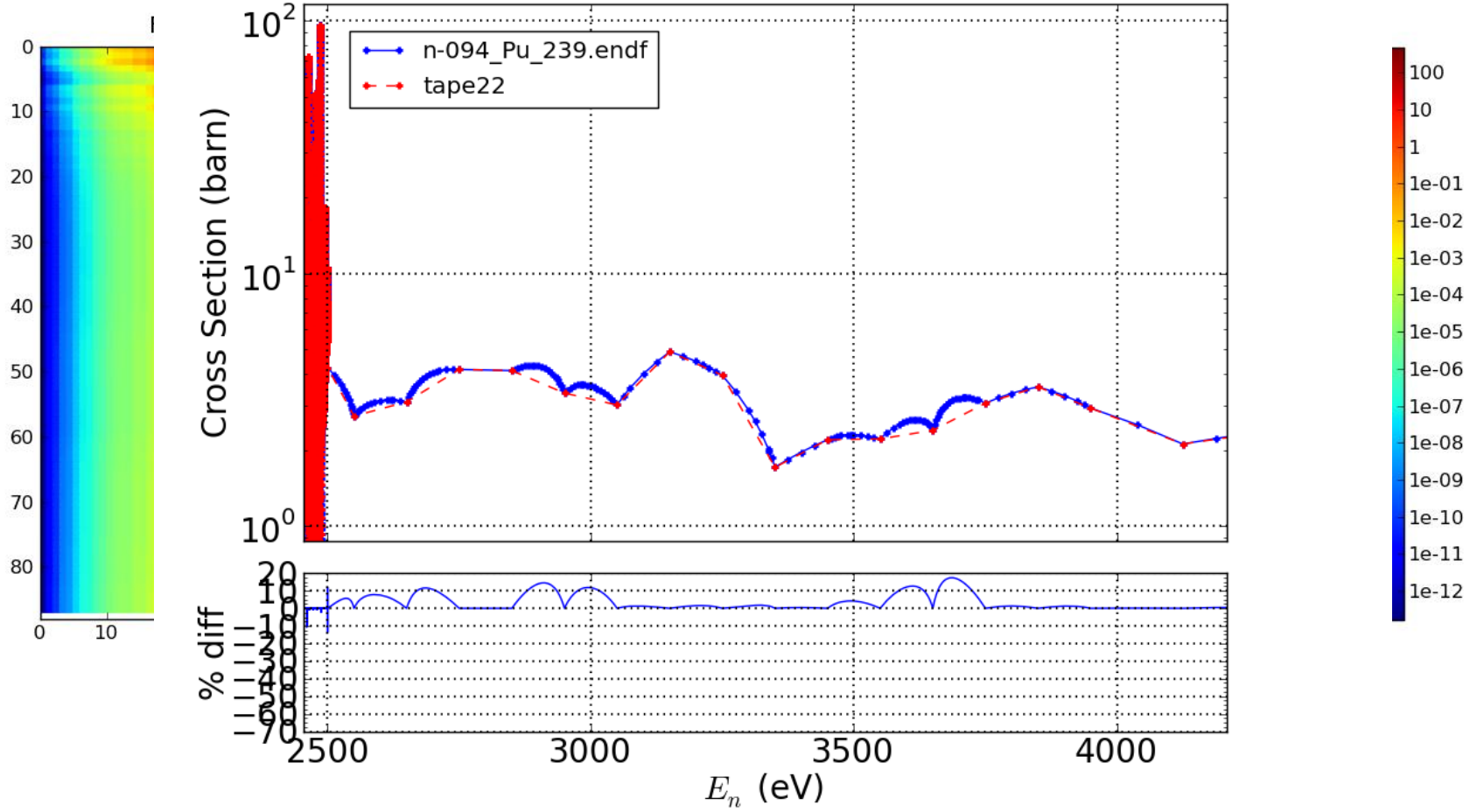
*exception: Madland-Nix parameterization

Transfer matrix comparison:

H1 (n, elastic) L=0 transfer matrices:

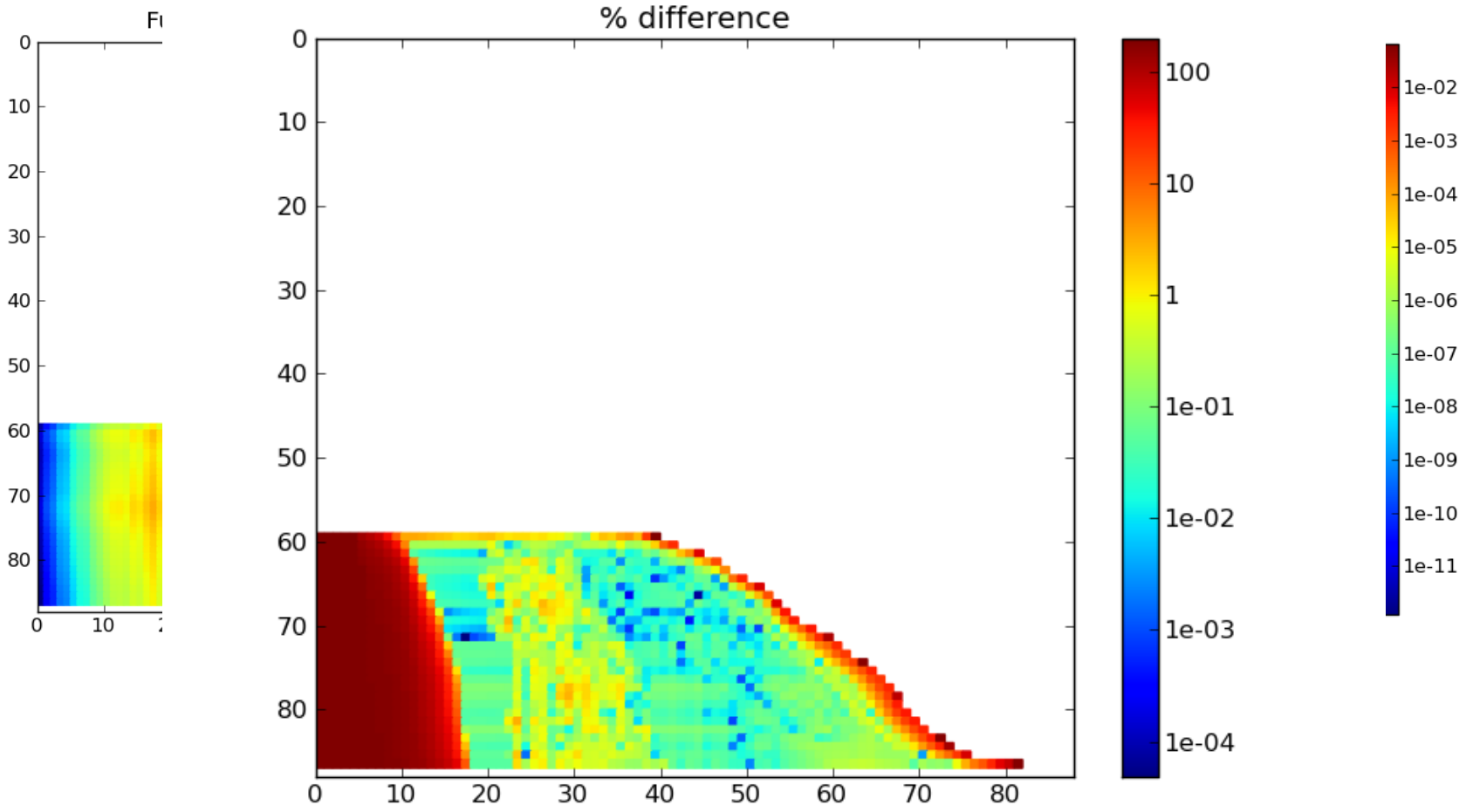


Comparing processing capability:



Comparing processing capability (Kalbach-Mann):

Pu239 (n, 2n) prompt neutron L=0 transfer matrices:



Processing for Monte Carlo codes:

- Two ways of handling MC data:
  - at LLNL, we are moving towards ‘minimal’ MC processing, with most of the work done by access routines
  - legacy codes, however, will continue to need pre-processed data

‘Minimal processing’

- Tasks when processing and using nuclear data:
  - Doppler broadening, generating transfer matrices (expensive)
  - grouping, equiprobable binning, cdf-ing, reaction-specific transfer matrix summing (cheap)

- To give users more freedom, trying to move ‘cheap’ tasks to happen at run-time, through access routines

Future work and legacy support:

- Heating cross sections: have code, need to link it into Fudge
- Continued testing against other processing codes: outgoing gammas, higher L-orders, various temps, comparing MC data...
- Translating to other formats: ACE, LLNL's legacy MCF/NDF, etc.

Sharing processed data

- GND could enable better sharing of data
- Common processed format, perhaps to be translated into site-specific (binary) forms?
- Format sample:

```
<grouped xData="matrix" size="87,87">
        <axes>
          <axis index="0" label="energy_in" unit="eV" interpolation="linear,flat" frame="lab"/>
          <axis index="1" label="energy_out" unit="eV" interpolation="linear,flat" frame="lab"/>
          <axis index="2" label="C_l(energy_in,energy_out)" unit="b" frame="lab"/></axes>
        <l value="0">
          <matrix rows="87" columns="87" form="sparse_asymmetric">
            0 0 1 8.04765353
            1 0 2 0.0128888641 7.59496411
            2 1 2 0.0385045077 8.95273144
            3 2 2 0.0592808172 11.3596173
            4 3 2 0.109015267 9.60703412
            5 4 2 0.0470416286 8.87828991
            ...
```

Conclusions:

- Processing is an essential requirement before GND can fill the vital role of ENDF in nuclear data
- Much progress has already been made on processing GND files; more is needed especially to support legacy codes
- We will continue to compare to NJOY, AMPX, etc. as we implement new processing codes
- Hopefully other codes will be able to handle GND as well!

87-group structure (abbreviated):

group #	lower bound (eV)
0	1.3068e-3
10	102.45
20	4233.9
30	98.909e+3
40	632.470e+3
50	2.5299e+6
60	6.0425e+6
70	10.12e+6
80	14.407e+6
87	18.134e+6