

## LA-UR-18-30509

Approved for public release; distribution is unlimited.

Title: Fitting Nuclear Data with Chebyshev Polynomials

Author(s): Josey, Colin James  
Brown, Forrest B.

Intended for: 2018 American Nuclear Society Winter Meeting and Nuclear Technology  
Expo, 2018-11-11/2018-11-15 (Orlando, Florida, United States)

Issued: 2018-11-02

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Fitting Nuclear Data with Chebyshev Polynomials



**Colin Josey, Forrest B. Brown**

2018 ANS Winter Meeting  
November 13, 2018



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

# Introduction

In Monte Carlo, nuclear data often dominates:

- the memory footprint
- the runtime performance

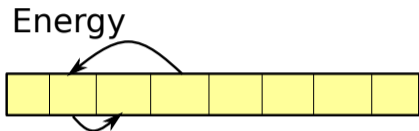
of a simulation.

These effects can be compounded with large numbers of nuclides or many temperatures.

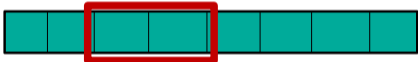
Most codes store nuclear data linear-linear pointwise.

In this presentation, we experiment with using piecewise Chebyshev polynomials.

# Data Lookup with Linear-Linear



Reaction 1



Reaction 2



1. Binary Search  
 $O(\log(n))$  Operations
2. Read Pair  
(No Sequential Access)
3. Interpolate

# Linear-Linear Pros and Cons

## Advantages

- Easy to generate
- Easy to plot
- Good compression

## Disadvantages

- Binary search yields many cache misses
- Each reaction loaded separately

Performance could be improved with some focus on cache management.

## Other Attempts to Reduce Costs

Many people focus on reducing the cost of the binary search:

- Hashing  
Perform  $\mathcal{O}(1)$  operation to reduce domain to search
- Fractional Cascading  
Amortize cost of current search through previous searches on other nuclides
- Unionization  
Put all nuclides on the same grid so only one search is done

Or completely eliminating the search, as is done in the windowed pole representation

# Chebyshev Format

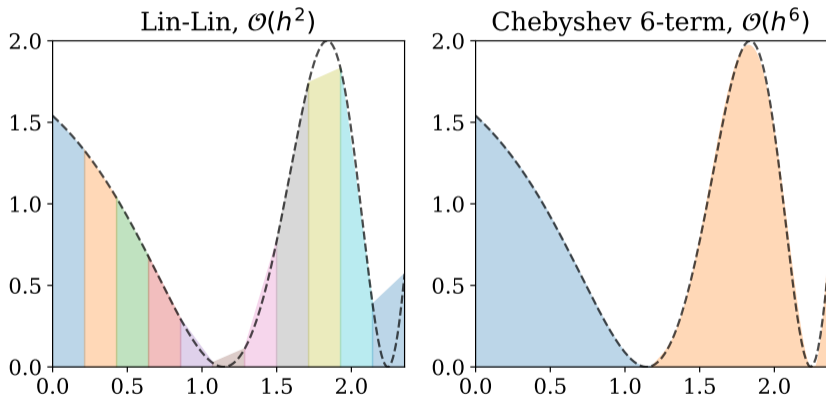
Our approach is to make the following changes:

- Replace lin-lin with piecewise Chebyshev  
Higher order → fewer regions needed to achieve an error → quicker binary search.
- Adjust the storage format to keep data at a given energy together  
Fewer cache misses reading multiple reactions.

How this was done will be discussed in the following slides.

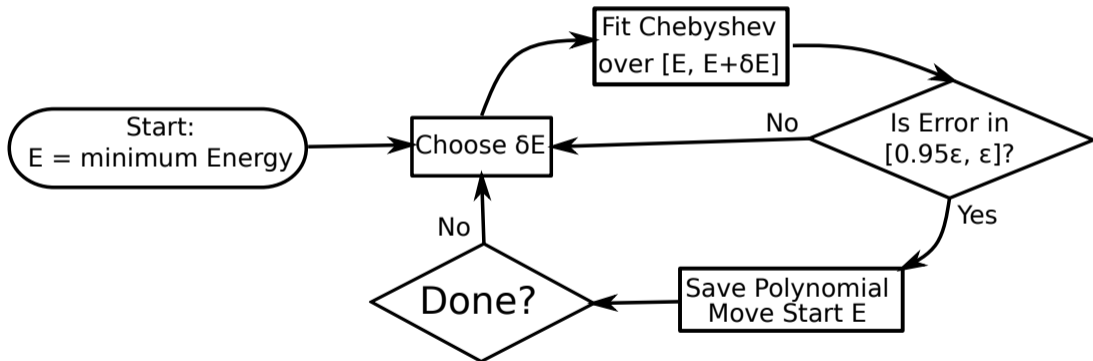


# Chebyshev Example



Same memory usage, but Chebyshev has 1/4 the points in the energy grid.

# Library Construction



# Fitting Chebyshev Polynomials

Stage 1: Form a Chebyshev Interpolant  $a$  with  $N$  terms

$$a_n \approx \frac{2 - \delta_{0n}}{N} \sum_{k=0}^{N-1} \cos \left( \frac{n\pi \left(k + \frac{1}{2}\right)}{N} \right) f(x_k)$$
$$x_k = \cos \left( \frac{\pi \left(k + \frac{1}{2}\right)}{N} \right)$$

Here,  $f(x)$  is the Doppler broadened cross section and  $x$  is energy.

For speed, the cosine sum can be performed with a discrete cosine transform.

Stage 2: Truncate  $a$  to first  $M$  terms,  $M \ll N$ .

For large  $N$ , this generates a good approximation to the “minimax” polynomial.

# Checking Error

To check error, we'd want a global maximum on some user-defined error.

**Option 1:** perform global optimization.

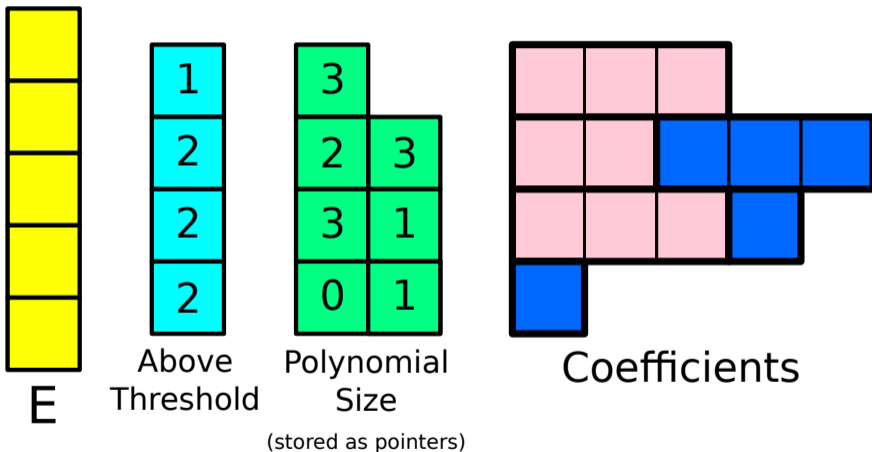
Requires numerous evaluations of  $f(x)$  over the domain.

**Option 2:** compute error for the  $x_k, f(x_k)$  pairs we already have.

As  $N \rightarrow \infty$ , this gives the global maximum.

To avoid missing features, local minima and maxima of the unbroadened cross section should be pre-broadened and added to the test points.

# Library Construction



All coefficients for an energy range are stored together.

# Choosing Nuclides to Test

**Best Case:** very smooth, complex data

The ideal nuclide has continuous ( $C^\infty$ ) data in which high order derivatives have a meaningful contribution. Resolved resonances with no File 3 data is ideal.

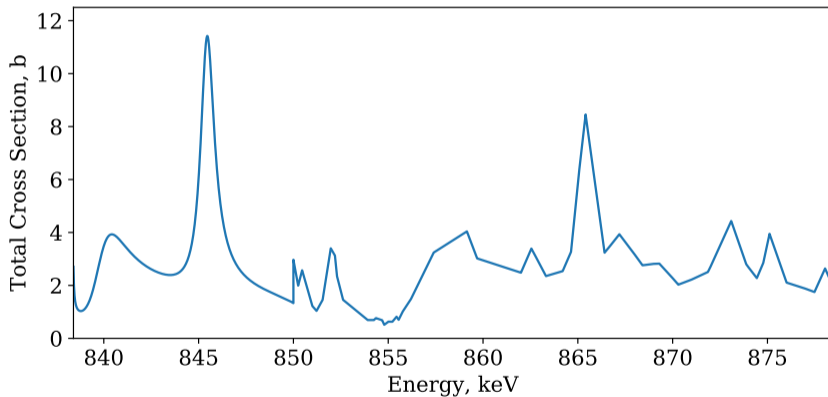
$^{235}\text{U}$  used for testing. Only has occasional discontinuities.

**Worst Case:** jagged or discontinuous data

This appears frequently in File 3 contributions.

$^{56}\text{Fe}$  used for testing.

# $^{56}\text{Fe}$ Data, ENDF/B-VII.1



Discontinuous at 850 keV,  $C^1$  discontinuous beyond.

# Testing Process

Libraries were:

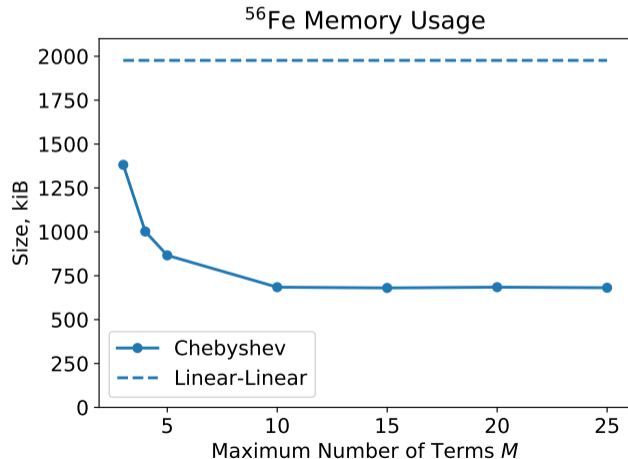
- Generated at 293.6K
- to an accuracy of 0.1%
- with a variety of terms  $M$

Performance was calculated:

- by timing 50 million evaluations (with random uniform lethargy)
- with 500 libraries in memory randomly selected for each evaluation (to eliminate cache effects)

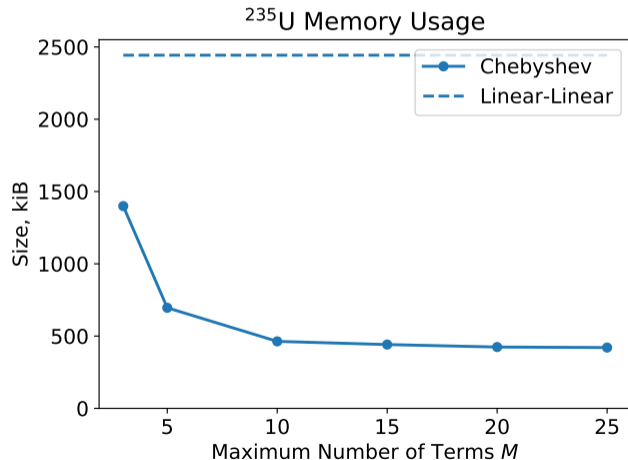


# $^{56}\text{Fe}$ Memory Usage



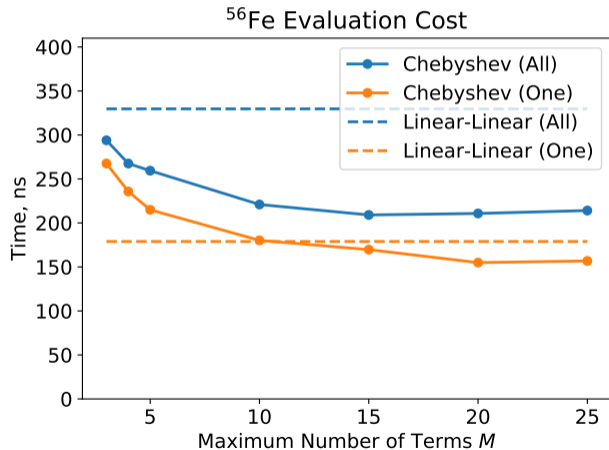
- Best size reduction: 2.9
- Nearly asymptotic at  $M = 10$

# $^{235}\text{U}$ Memory Usage



- Best size reduction: 5.8
- Smooth data compresses much better

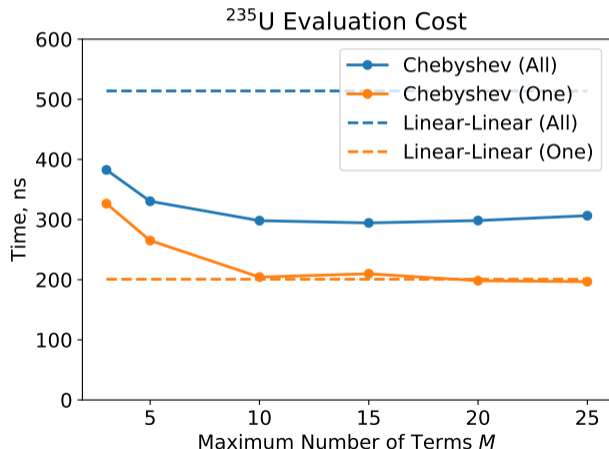
# $^{56}\text{Fe}$ Performance



13% speedup - one reaction

36% speedup - all reactions

# $^{235}\text{U}$ Performance



$\approx 0\%$  speedup - one reaction

42% speedup - all reactions

# Summary

With regard to memory:

- The smoother the data, the more compressible.
- Even fairly rough data compressed by a factor of 2.9.

With regard to speed:

- Single reaction lookup was approximately the same.
- Looking up all reactions at once resulted in a 35-40% speedup.