Lecture 4.5:
Interpolation and Splines

D. Jason Koskinen
koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics
Feb - Apr 2021

Photo by Howard Jackman

Continuous Data/Description

- Inherently, data and Monte Carlo simulation provide discrete units of information. Often what analyzers want is a continuous description which can be achieved using interpolation and extrapolation techniques
  - Interpolation - Describing data in-between known data points, where ‘known data’ also includes finite Monte Carlo. Can smooth out artifacts of the simulation process or discrete data taking
  - Extrapolation - Estimating beyond the range of known data. A dodgy practice when the behavior beyond the data range is relatively unknown. Without a specific well described scenario there is no great method for extrapolation

Interpolation

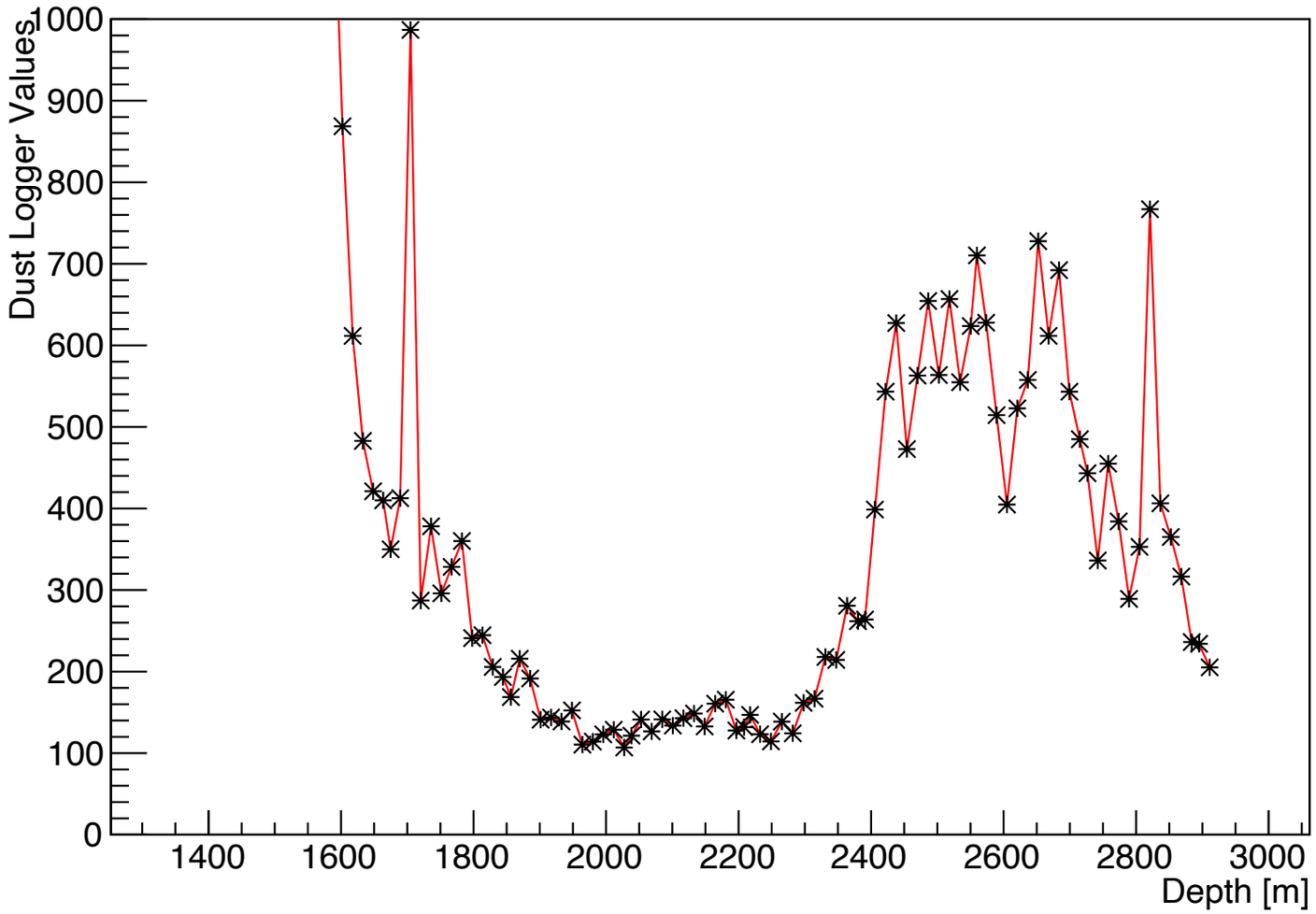
- Numerically/Mathematically provide information about the data between known points
- Has numerous advantages when you have data that is unlikely to, or just cannot, be reasonably modeled or fit with any combination of analytic functions
- A simple interpolator is a line (linear) between each data point

Interpolation

*http://dx.doi.org/10.3189/2013JoG13J068

**Full disclosure, I do not fully know what the “Dust Logger” values are, besides ‘optical intensity’

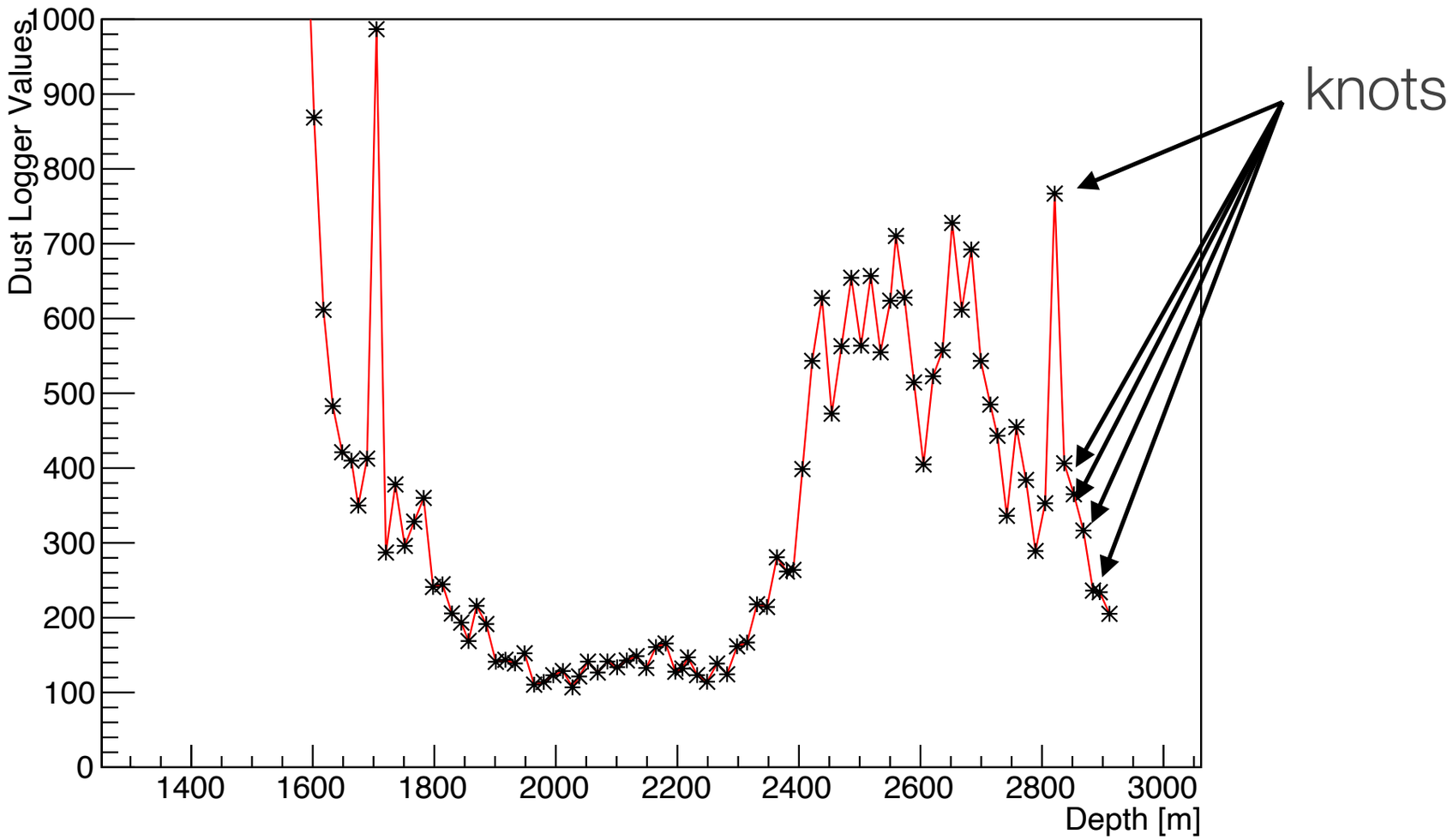
- A simple interpolator is a line (linear) between each data point
- Dust logger data taken from sample in Antarctica. Glacial dust across millennia has an incredible array of features



Interpolation

*http://dx.doi.org/10.3189/2013JoG13J068

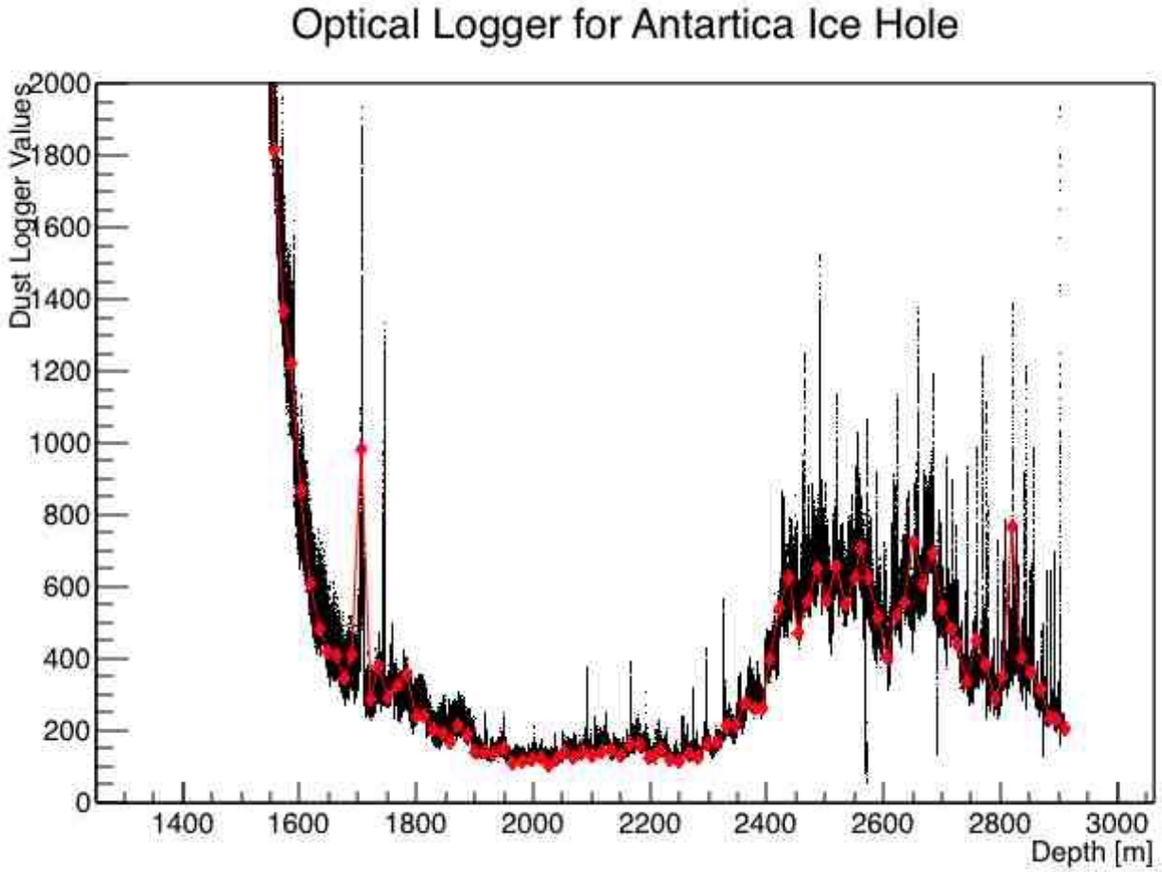
- Each data point included in the spline creation is known as a ‘knot’

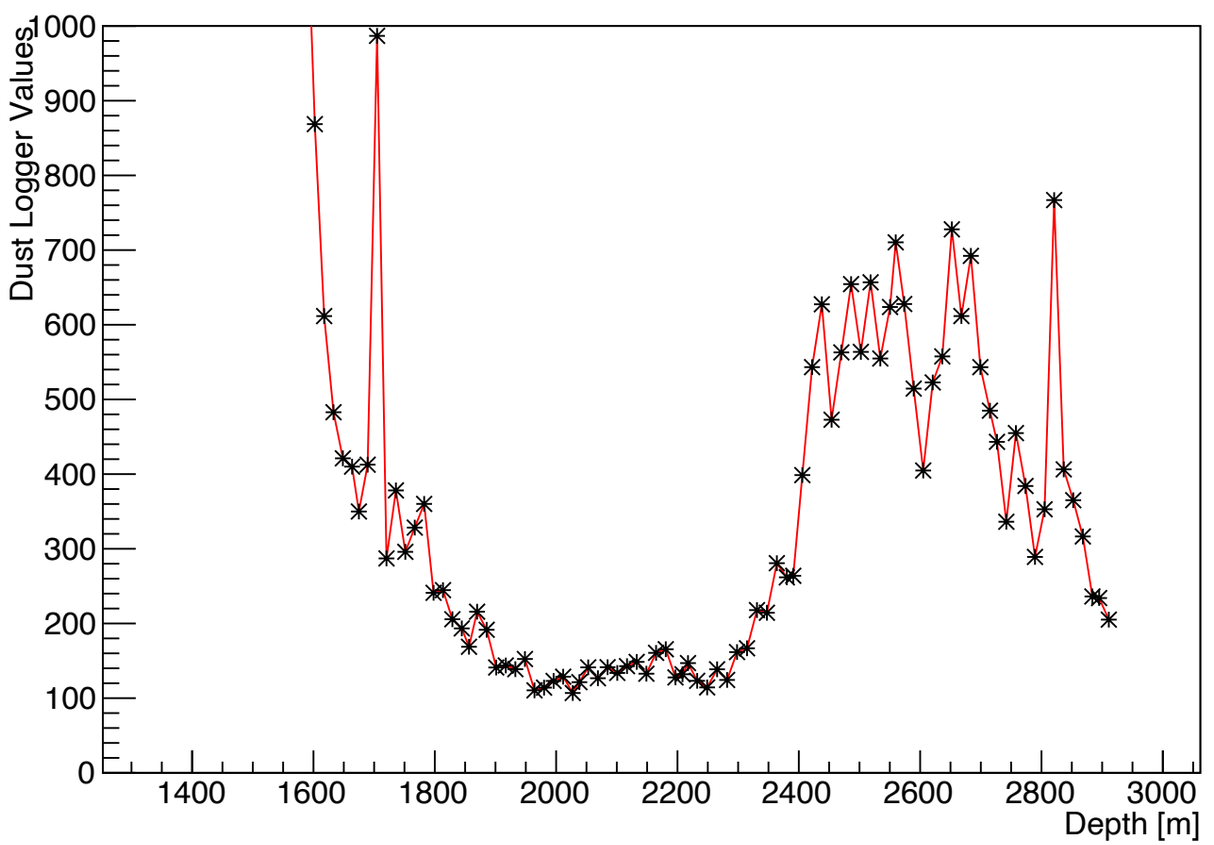


Interpolation

- The actual data has ~1.28M entries (left plot), while the linear spline only uses every 10,000th entry, i.e. 128 knots

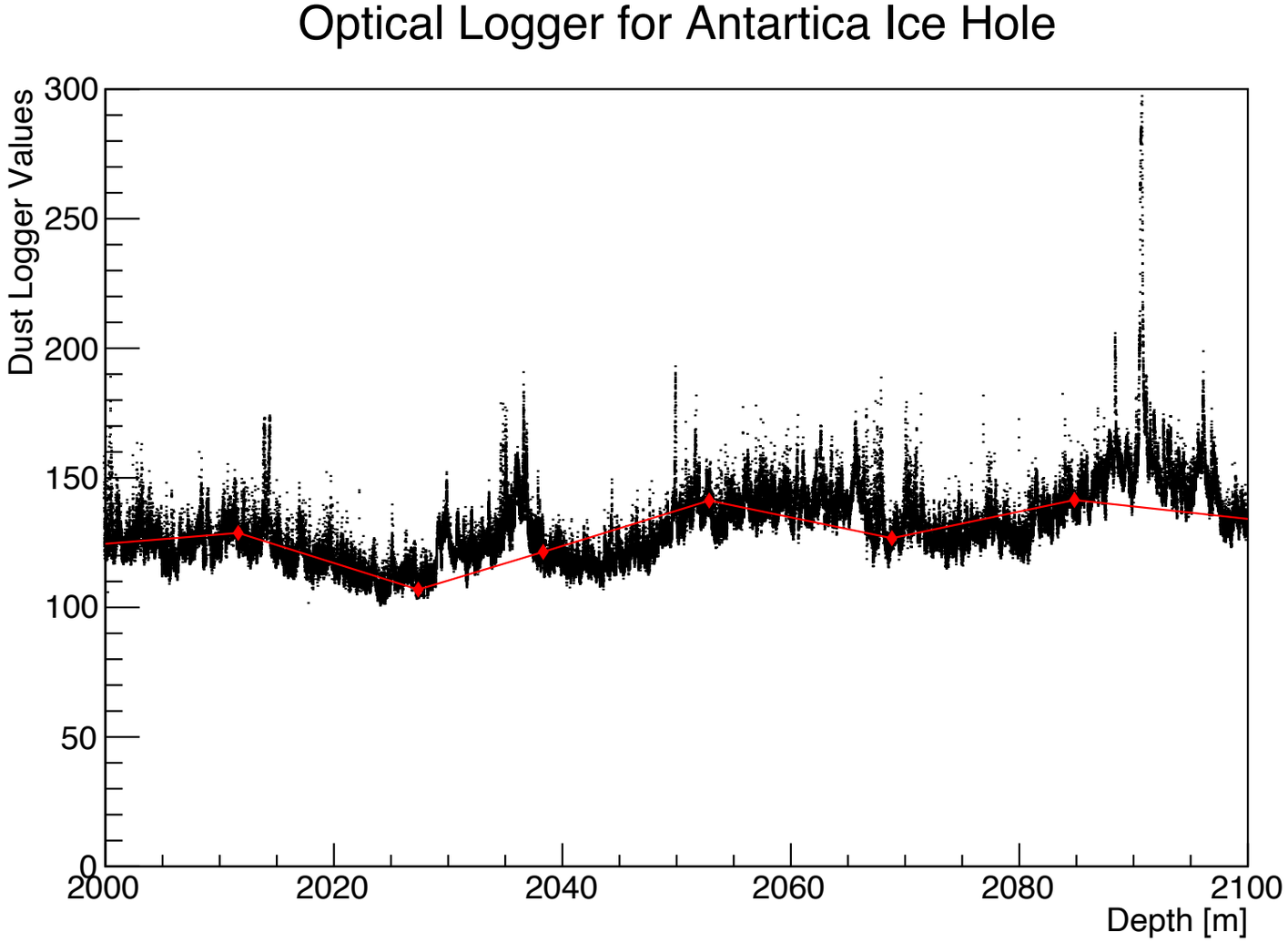
Image is ‘grainy’ because PDF images don’t like being 1+ million points without crashing viewers





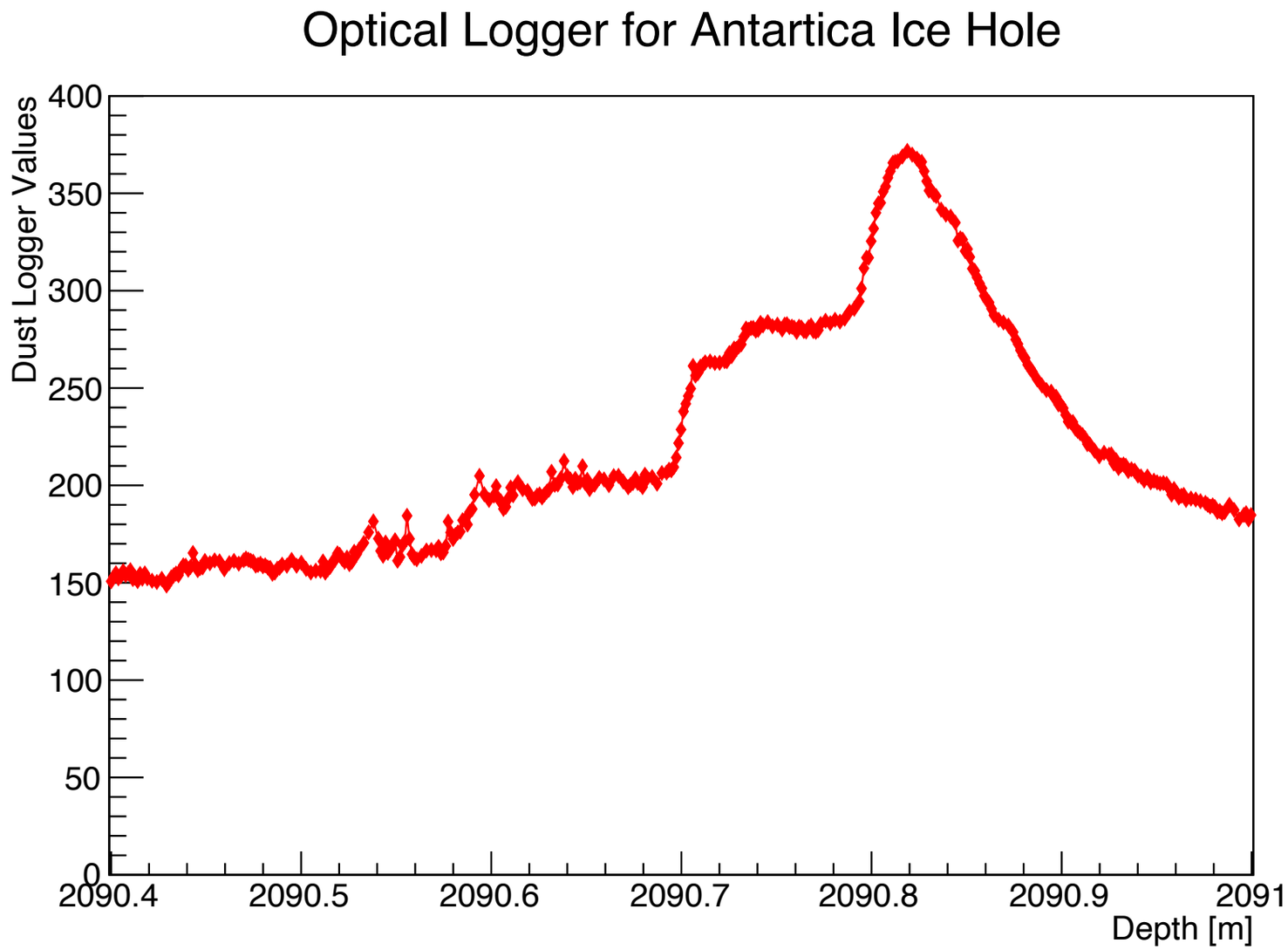
Close Look

- Zoomed in we see that the 128-knot spline gets most of the features right, but since it's not created using all of data there are some missing features.
- Depending on the knot sample method, the spline results can change. Normally not an issue because interpolation will include **every** data point as knots



Really Close Look

- There exist very real features at the centimeter scale in depth for over 2 km of measurements

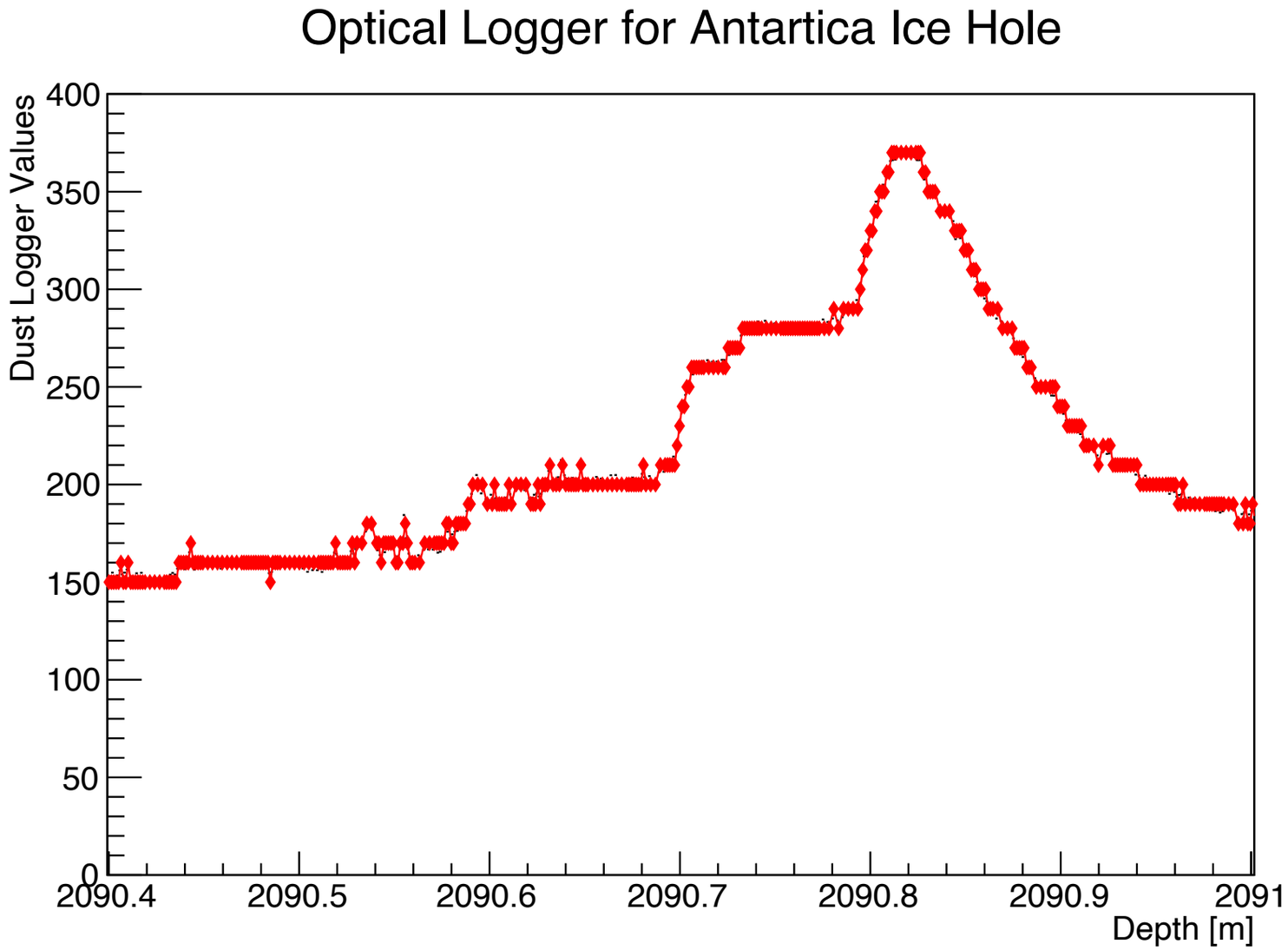


Spline/Interpolation Use

- Where do we want to use splines?
- Computer aided drawing and graphics
- Creating continuous functions from discrete data
- Creating smooth functions from jagged or irregular data

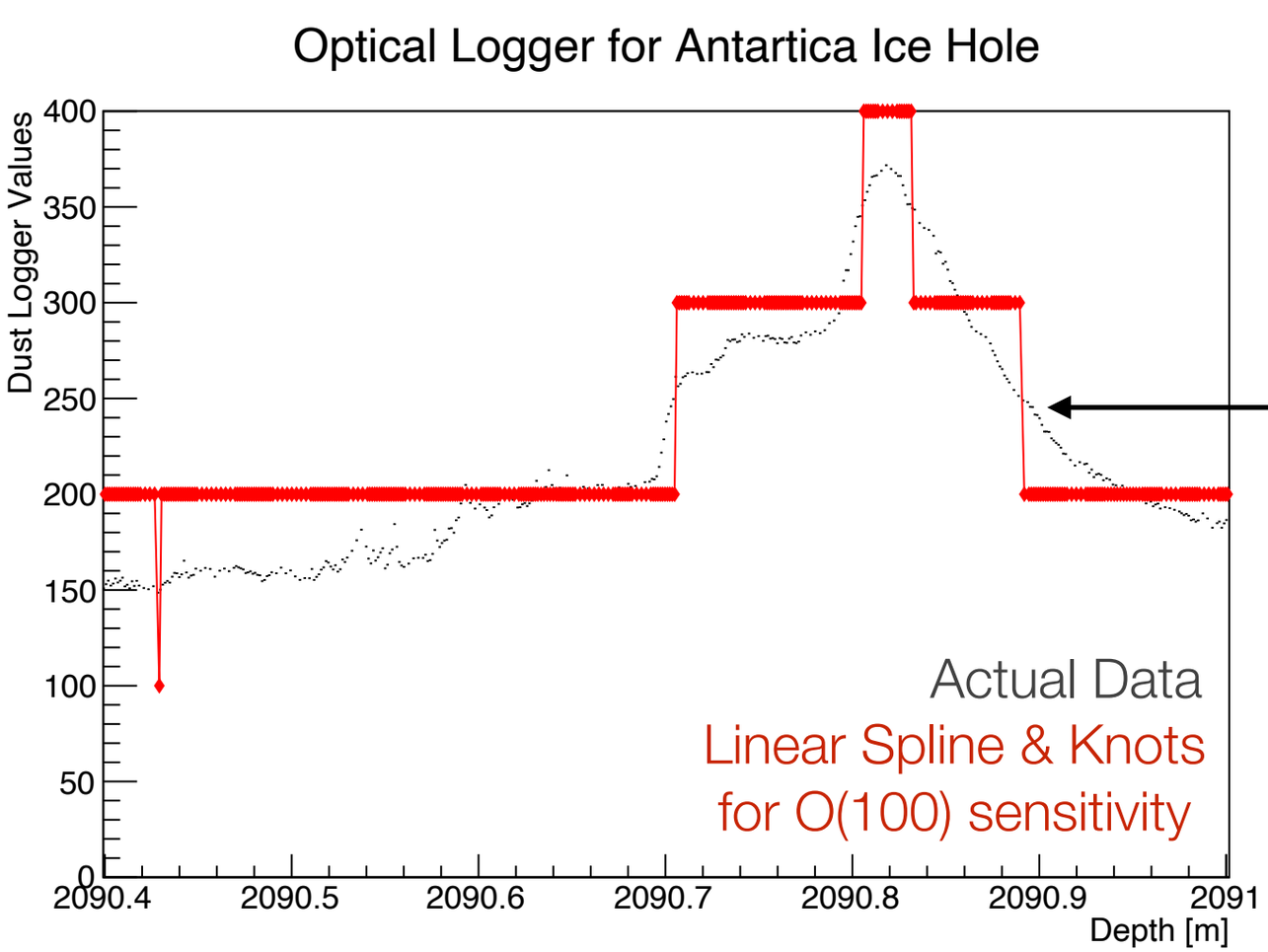
Modified Scenario

- Now let's imagine that the optical instrument is only sensitive to values of $\mathcal{O}(10)$ m in optical intensity, e.g. 220, 230, and 240 instead of 220.213, 232.66, and 244.391. But, we *know* that there are features.



Modified Scenario

- Or even worse, being sensitive at $\mathcal{O}(100)$ m instead of $\mathcal{O}(10)$ m
- Now we have a lot of duplicates, and effectively discrete jumps in Dust Logger Values over very, very, short changes in depth

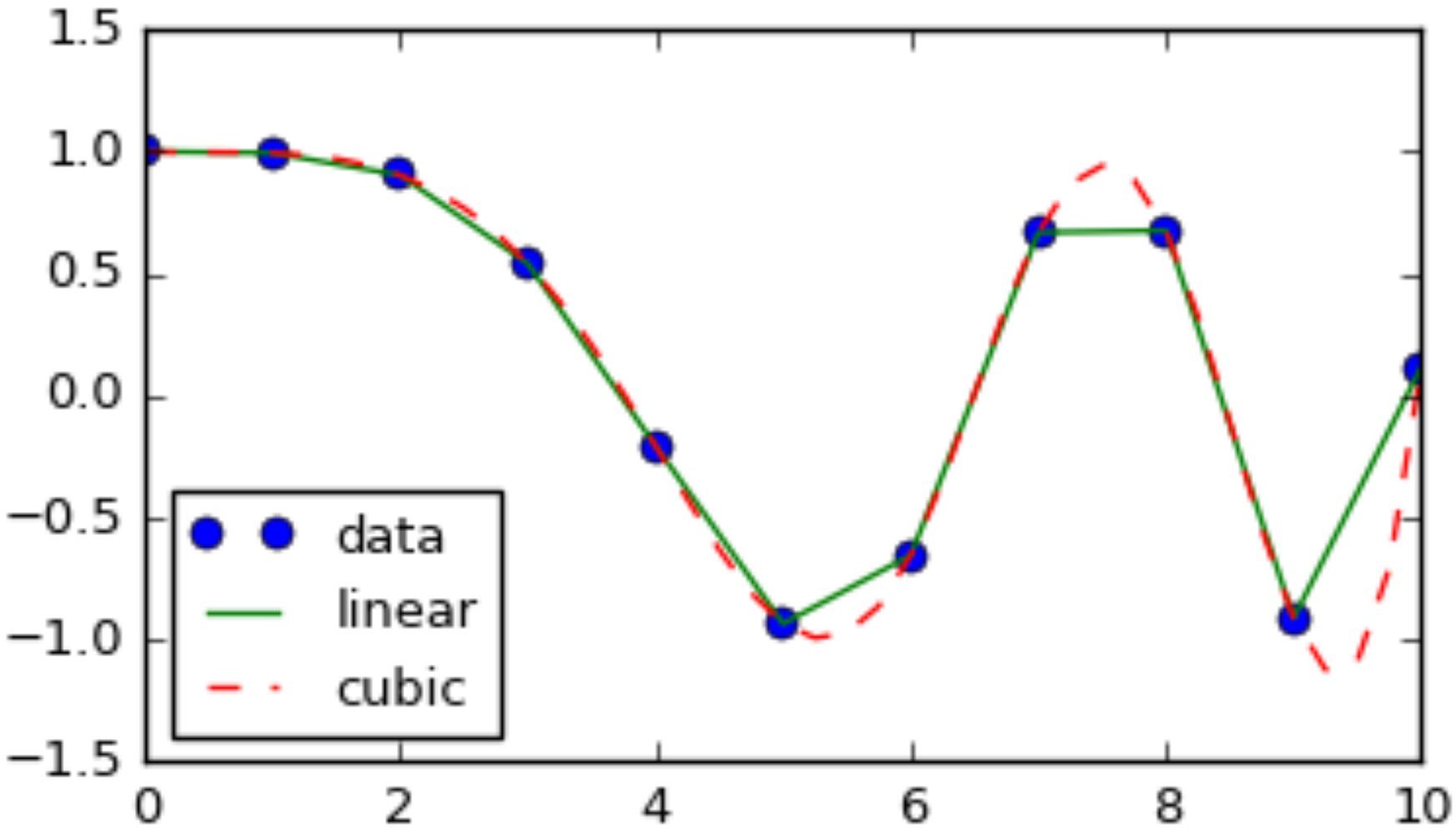


Separated by mm in depth, yet logger would produce drastically different values. If we ‘know’ that there are smoothly varying features we can proceed to smooth the data.

Cubic Splines

*Scipy interpolate

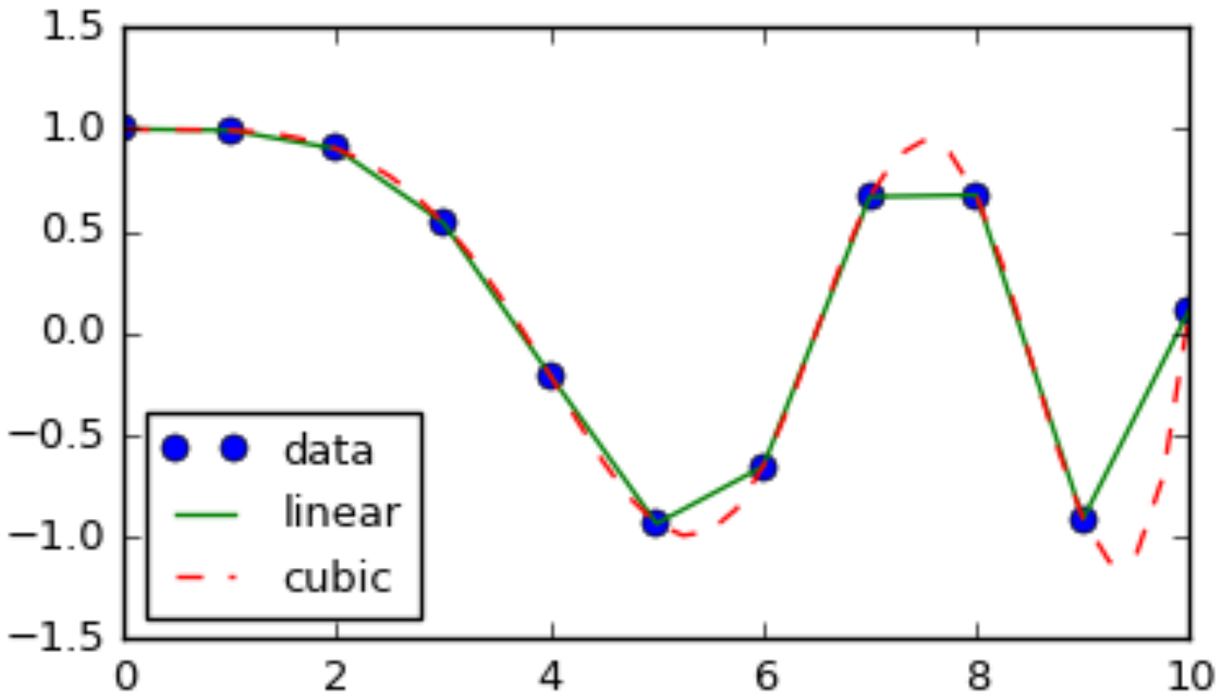
- If we know that the data smoothly changes, it is a possibility to use a cubic spline which ‘smoothes’ the interpolation



Common Spline Types

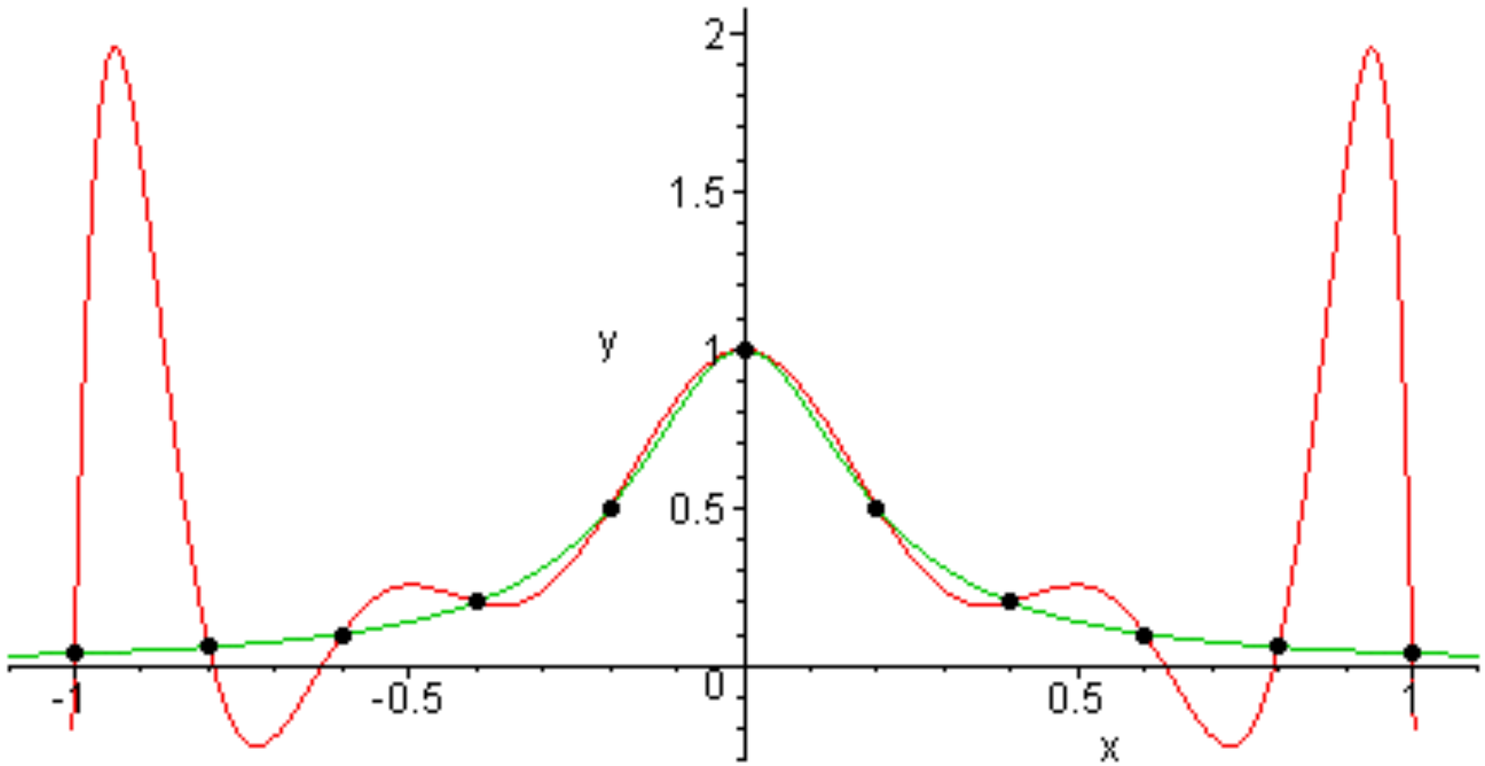
*Scipy interpolate

- Linear splines are continuous across the data points, but do not match the 1st or 2nd derivative at the knots
- Quadratic splines (not shown) match the 1st derivative but not necessarily the 2nd
- Cubic splines are continuous and match the 1st and 2nd derivative at the knots



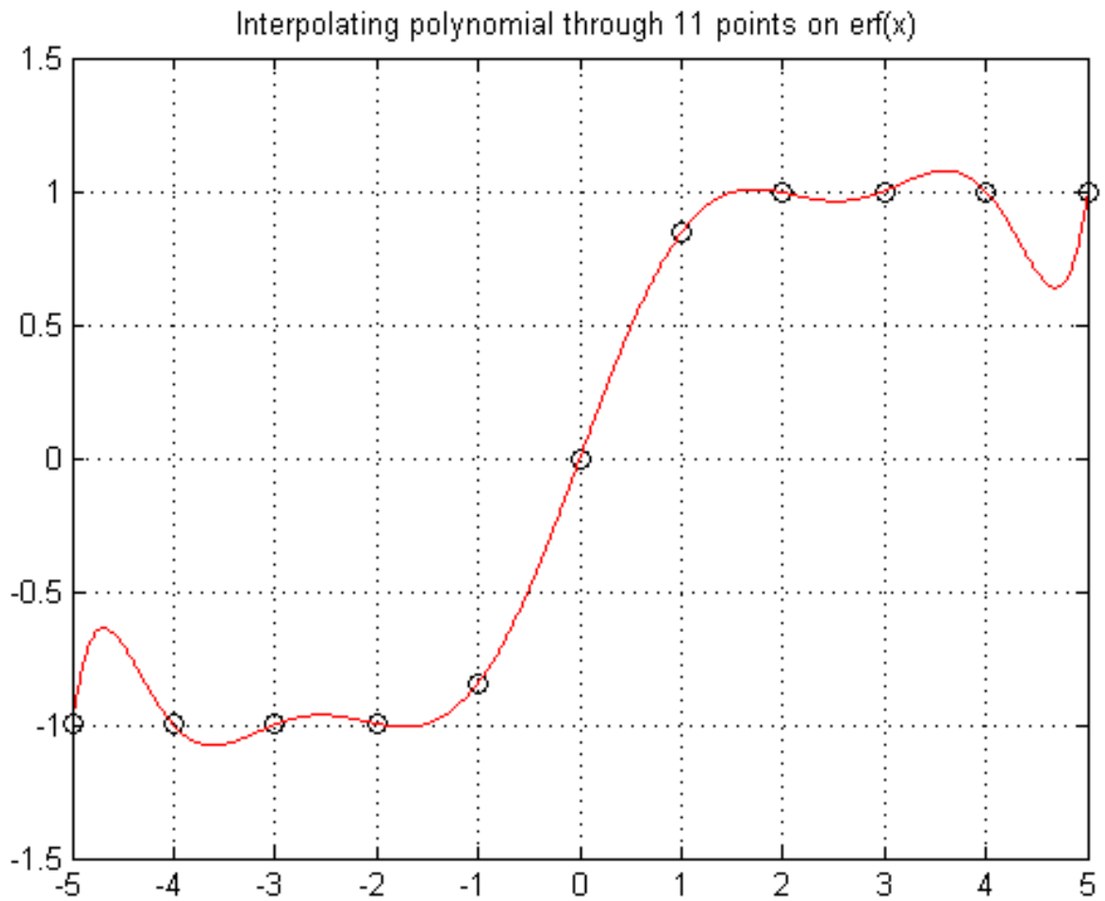
Polynomial Interpolation

- For n data points, an approx. n-order polynomial will go through all the data points smoothly
- Often though, the interpolation behavior near the edges is problematic
  - Imagine fitting a 1.28M order polynomial



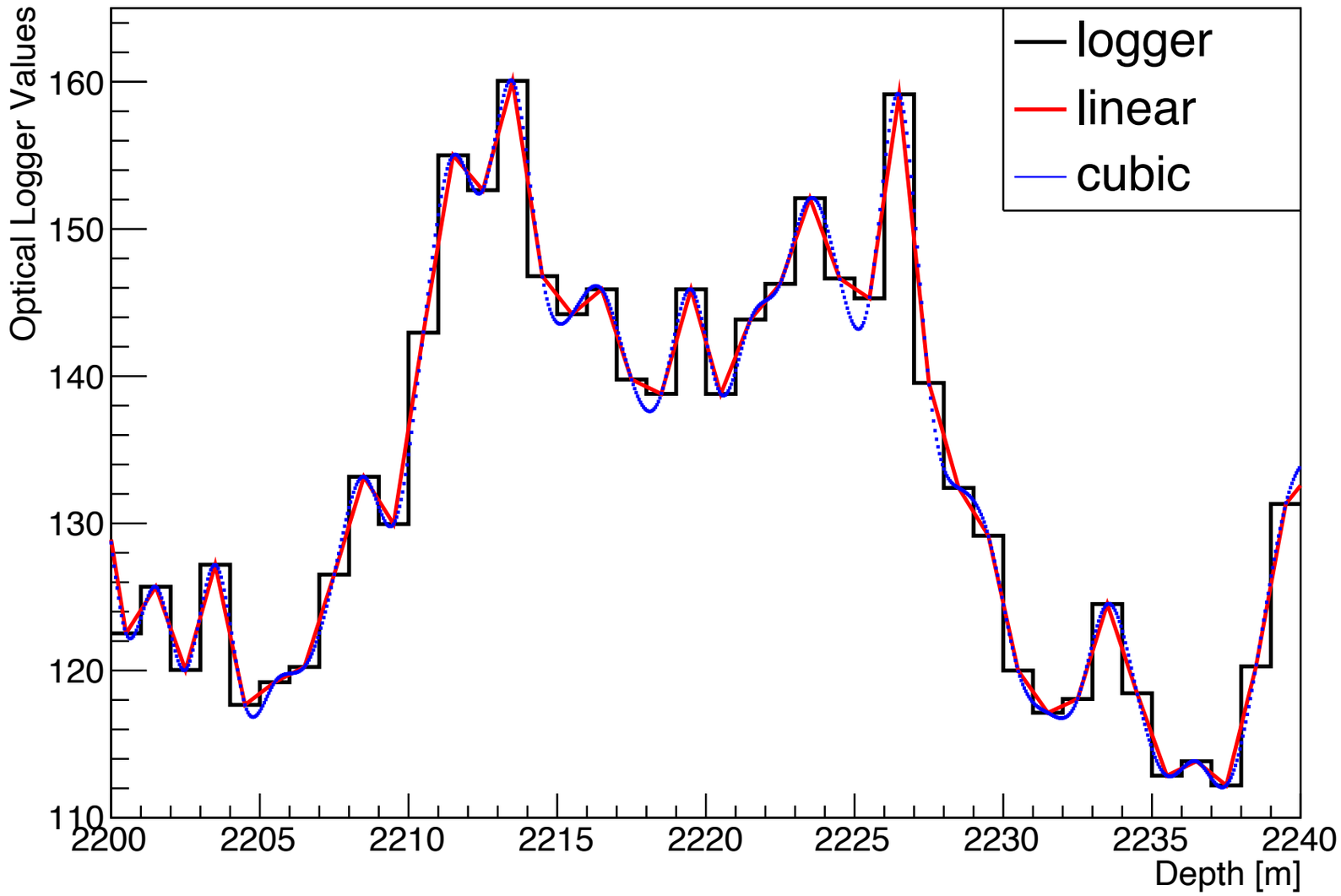
Polynomial Interpolation

- A problem referred to as ‘ringing’ is also pronounced in polynomial interpolations. In transitions to flat data, the polynomial will carry some influence from the matching between the 1st and/or 2nd derivative from the previous knot(s) to the next knot(s). Here, the interpolation should be zero between the first few knots, as well as the between the last few knots.



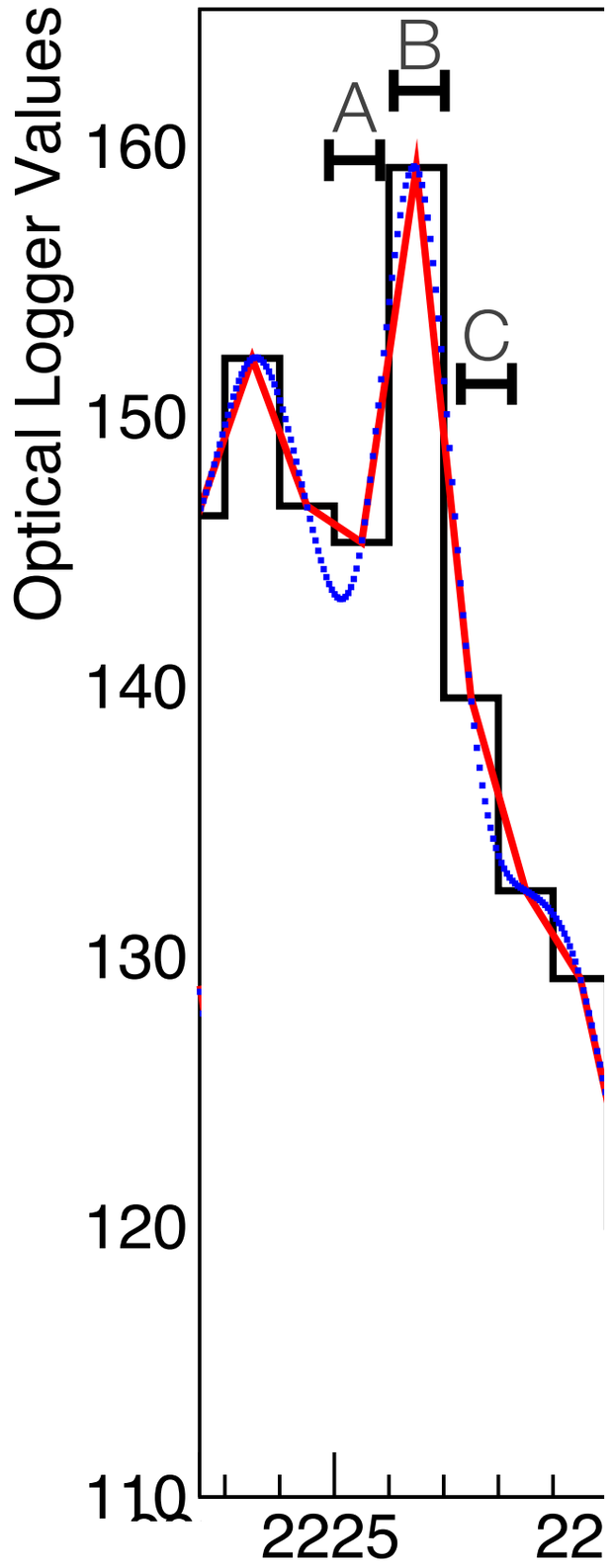
Cubic Spline Comments

- While it is nice to have smooth interpolations, cubic splines have some drawbacks, especially in ranges where the knots go from increasing in value to decreasing, and vice versa. The more substantial the change, i.e. magnitude of the 2nd derivative, the more likely a cubic spline will be problematic and over/under shoot.



Spline Under/Over Shoot

- The following linear and cubic splines can not match the data of the underlying histogram, i.e. the summation/ integral over the same depth produces different values
- At the right there are three regions (A, B, C) each representing a certain region in depth. Compared to the data (black histogram):
  - In region A, the linear spline integral ‘over predicts’ the data, whereas the cubic spline integral mostly matches the value from data (~146)
  - In region B, both the linear and cubic ‘under predict’ the data (~159)
  - In region C, both the linear and cubic spline look to match the value from data (~140)



1D versus Multi-Dimensional

- There are some nice tools for doing 2D interpolation and spline fits (scipy.interpolate.interp2d, scipy.interpolate.griddata)
  - In a pinch, you can create many 1D splines to map out the multi-dimensional space
  - We will be sticking with 1D splines and interpolation

Spline Exercise #1

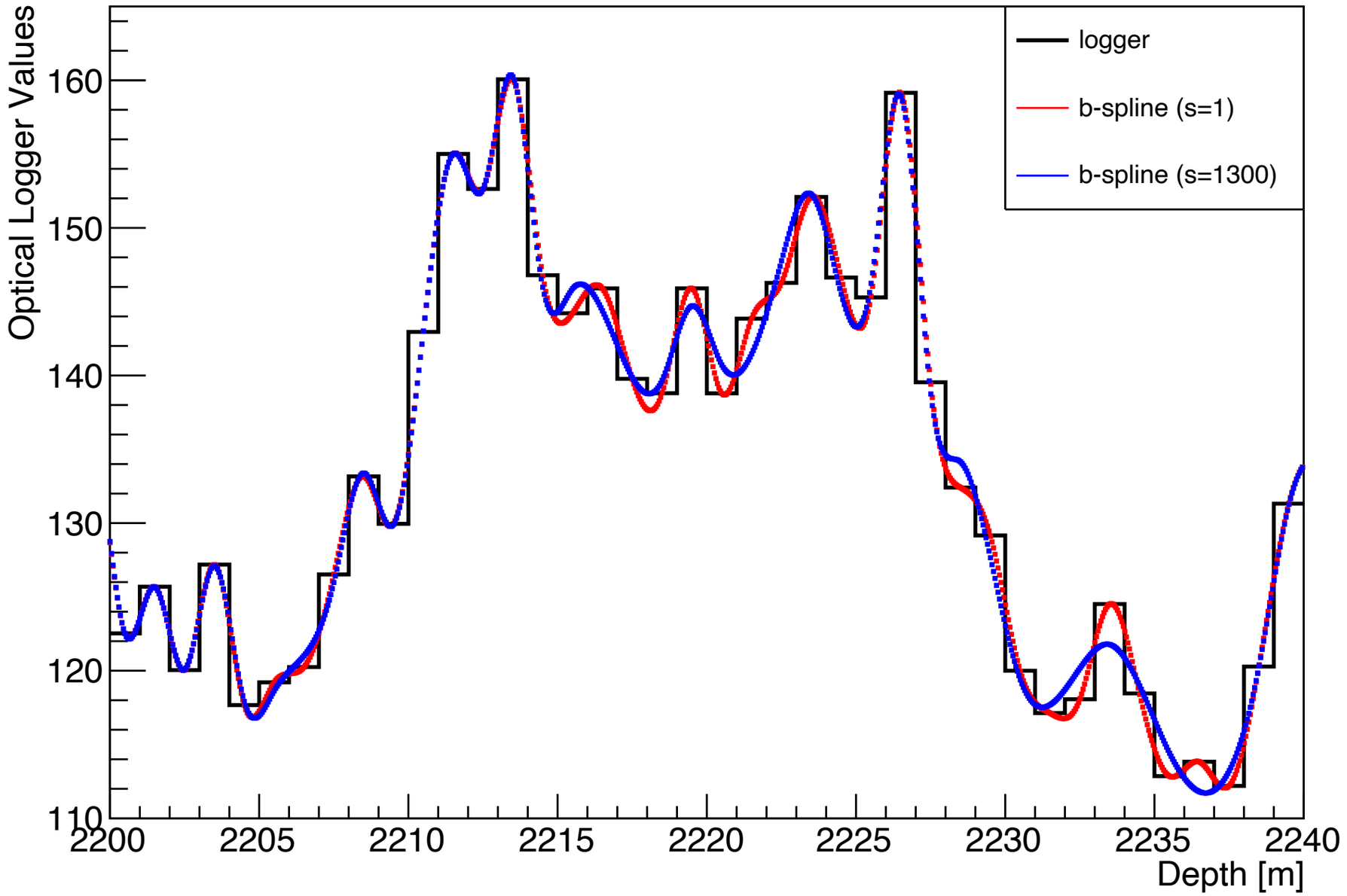
- There is a optical/dust logger file
  - http://www.nbi.dk/~koskinen/Teaching/data/DustLog_forClass.dat
- Construct a 1D interpolation for the data:
  - Linear spline
  - Cubic spline
    - Use default options if possible. Don't worry about other options, namely 'smoothness' or 'weights'
- Plot the splines and the data on the same graph

b-splines and Smoothing

- Basis splines (b-splines) are probably what you used to create the cubic splines. They are piecewise polynomials of order k (k=3 for cubic), where the interpolated value and most often the 1st derivative and 2nd derivative match the adjacent piece-wise polynomials at the knots.
- There is a parameter ‘smoothness’ which can regulate the behavior of the spline
  - Large smoothness means a cubic spline is more smooth (less bumpy), but also not constrained to go through the knots
  - Small smoothness means the splines are constrained to be close to the knots.

b-splines and Smoothing

- General metric is that if you want to smooth an odd-order spline with n data points use $s = n \pm \sqrt{2n}$



Exercise #2

- I have sampled from an ‘unknown’ function in one-dimension at regular intervals from $1x10^{-5}$ to 1 which is going to be used as a PDF over the same range
  - The x and y values are on the class webpage at SplineCubic.txt
  - There is some small ‘noise’ contamination introduced too
- For a linear, quadratic, and cubic spline to the same data, make a comparison between the integral from $1x10^{-5}$ to 0.01 between the three splines
  - Is there any reason to use a ‘smoothed’ spline?
  - Repeat the same comparison for integrals between 0.03 and 0.1

Exercise #3

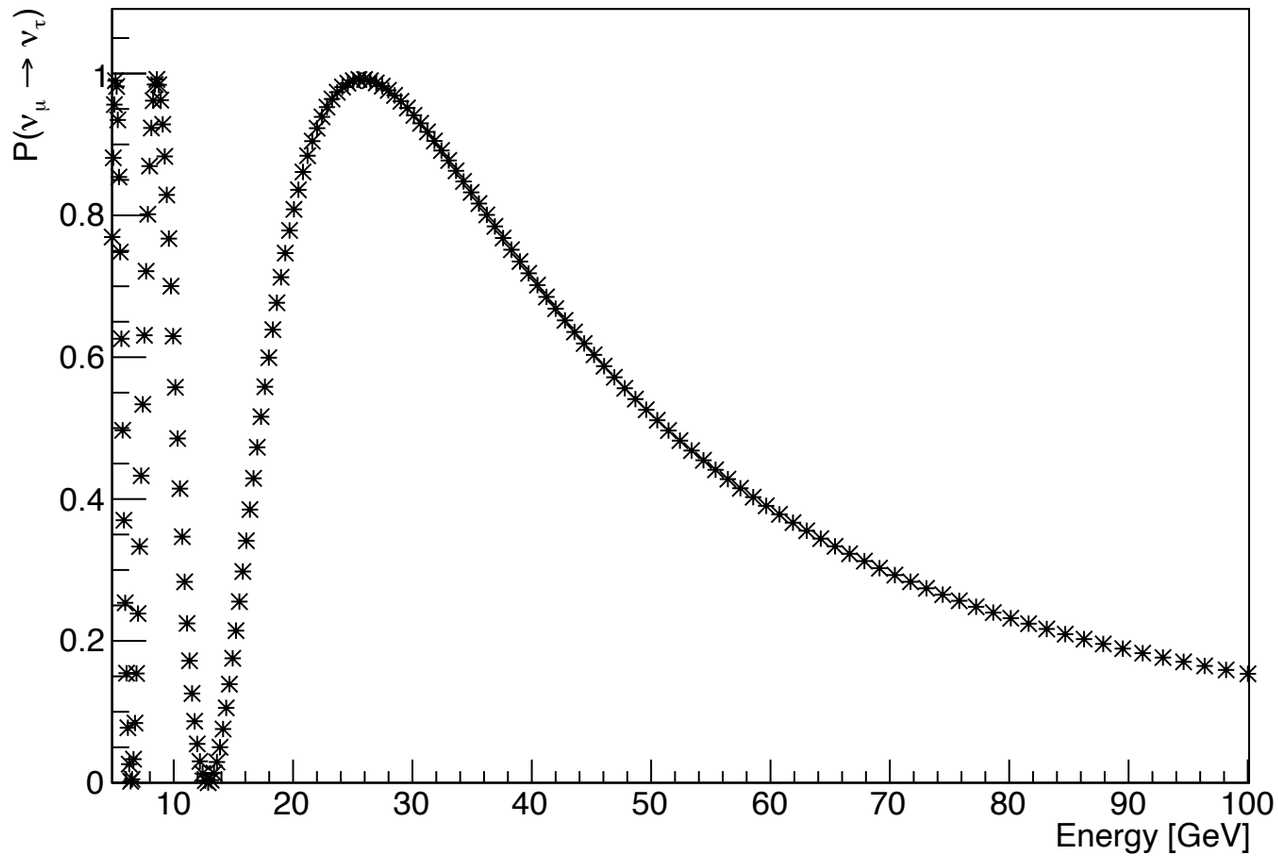
- Neutrino oscillation is a physical phenomena whereby a neutrino changes flavor during its travel. A simplified expression for the probability of oscillation, i.e. flavor change,

$$P(\nu_\mu \to \nu_\tau) \propto \sin^2(2\theta_{23}) \sin^2\left(\frac{1.27 \cdot L \cdot \Delta m^2_{32}}{E}\right)$$

- On the class webpage there is a file SplineOsc1.txt, which has nominal values of the oscillation probability
  - The parameters θ_{23} and Δm^2_{32} are set by global experiments and here L is a neutrino travel distance of 13000 km
  - Column 1 is the energy (E) and 2 is the oscillation probability (P)

Exercise #3

- At very small values of energy, the oscillation goes into ‘rapid oscillation’, where experimentally the probability is best approximated by 0.5
  - If you spline the data in the file, at what energy is the sampling rate of the data too sparse to accurately reflect the true oscillation probability?
  - Can you use the ‘smoothness’ spline feature to produce a spline function which matches the osc. prob. where the sampling is okay, and then averages to 0.5 when the sampling rate stops being sinusoidal at lower energy?



Exercise Extra

- For the very brave, there is a file uploaded to http://www.nbi.dk/~koskinen/Teaching/data/honda2012_spl_solmin.d, which is the atmospheric neutrino flux in binned groupings of energy, zenith angle, azimuth angle, and neutrino flavor type
- See if you can parse the data to make a 1D spline in energy of the flux value for a single azimuth bin, a single zenith angle bin, and a single neutrino flavor
- Can you do the spline in 2D now, including the zenith angle?
- The energy bins have integrals which are important to be preserved even after the spline creation. How different is the integral of the 2D spline function compared to the binned integral over the same range(s)?

Extra Extra

- Write an integral preserving interpolation for the 3D case of energy, zenith angle, and azimuth angle in the atmospheric neutrino flux file
  - Only for energies < 5 GeV