

This section contains a walk through of the method to back diffuse a given ice core depth series to attempt to restore as much of the original signal as possible. For now, the method has been developed for sections that has been dated to be between the two volcanic eruptions Laki and Tambora, as these are very well dated, and thus it is possible to find the optimal diffusion length to back diffuse with as the actual number of annual layers is exactly known for this data series. The method is easily modified for any other dated depth series, as what is needed is just the number of annual layers in the given data series.

Figure 0.1 shows a flowchart of the method used to estimate the diffusion length of a depth series with a preliminary guess of number of annual layers. In the following sections each of the steps in the method will be discussed more thoroughly and examples will be given, all based on the Greenlandic ice core drilled at Site A near Crete(REFERENCES!!!). The method is built such that it takes two inputs - the isotopic depth series, and the specifications for the particular ice core - and uses these for the first preliminary computations needed to make a first, naïve guess of the diffusion length,  $\sigma_0$ . This diffusion length is then used to deconvolute the data and give a first estimate of the number of peaks in the data series. If this number is different from the already specified number of annual layers - in this case 32 - then the diffusion length will be updated accordingly: If the counted number is higher(lower) than the actual number, the diffusion length is adjusted downwards(upwards) with  $\Delta\sigma_2$  and the deconvolution and peak counting is performed again. On the other hand, if the counted number is equal to the actual number, then the diffusion length is optimized to find the largest diffusion length which still gives the actual number of counted peaks. When this  $\sigma_{\text{final}}$  is reached, the algorithm stops and returns the final diffusion length estimate along with the associated back diffused depth series.

### 0.0.1 Input

To compute the final diffusion length and depth series, two inputs are needed: the measured isotopic depth series and the specifications of the examined core. Through this section all examples have been carried out using the core Site A.

In Figure 0.2 the depth series between the eruptions Laki and Tambora can be seen along with the entire ice core in the background. This is the diffused, measured raw data from Site A. Dating of the ice cores has been carried out by matching Electrical Conductivity Measurements (ECM, Section ??) with

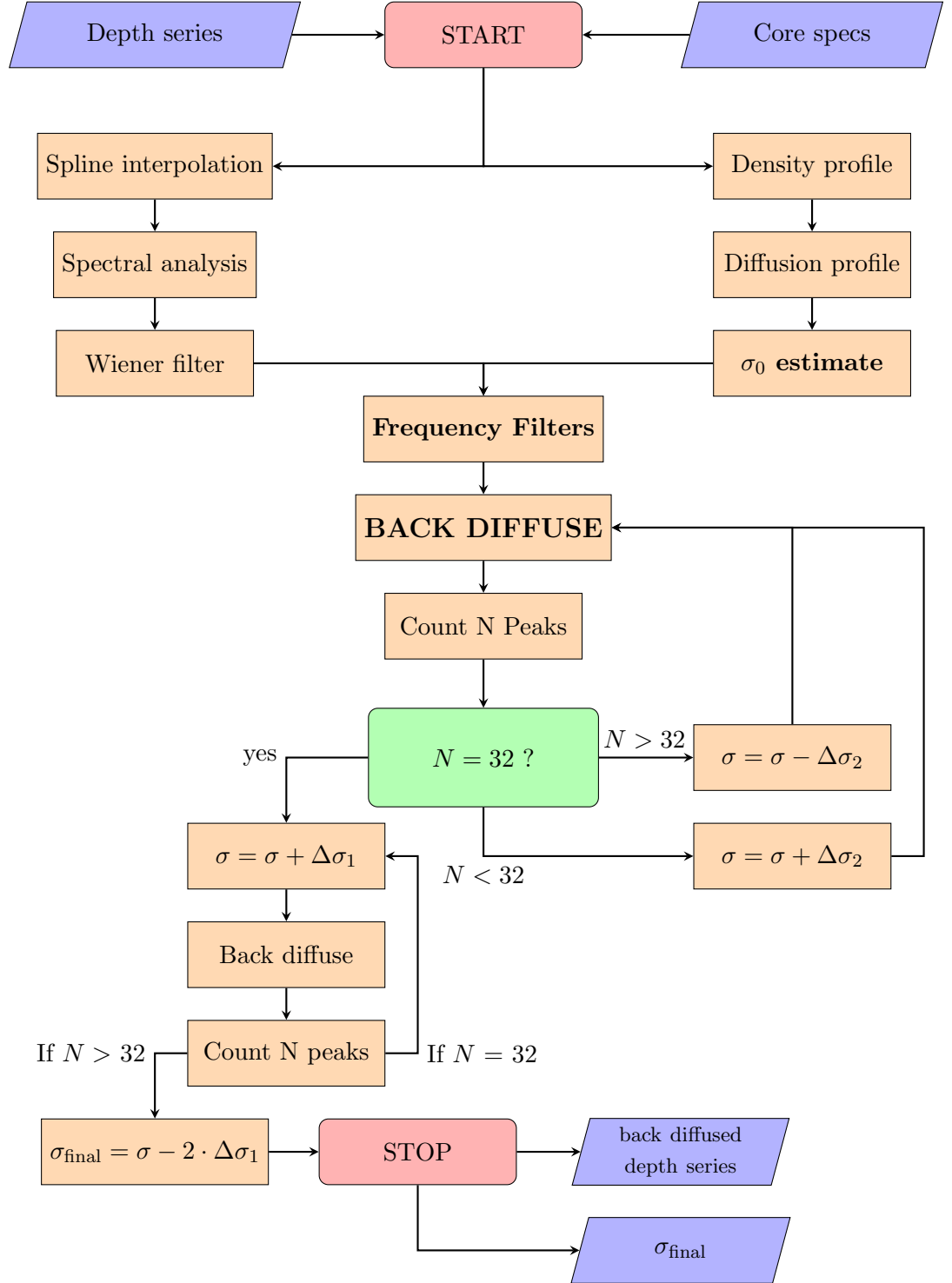


Figure 0.1: Flowchart of method for diffusion length computation.

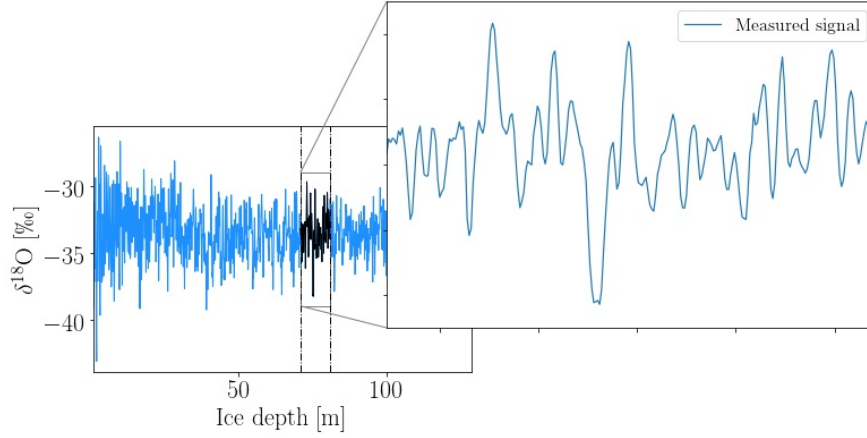


Figure 0.2: The entire ice core isotopic profile from Site A, with a zoom in on the estimated depth series spanning from Tambora to Laki.

Core	Drilled	Core length	Geographic position		Elevation	Laki	Tambora	Mean accum. rate	Temp. at 10m	Temp at 20m
ID	[Yr]	[m]	Latitude [°N]	Longitude [°E]	[m a.s.l.]	[m]	[m]	[m ice/Yr]	[°C]	[°C]
Crete	1974	404.0	71.12	322.68	3172	74.75	64.70	0.280	-30.40	-30.16
Milcent	1973	398.0	70.30	315.00	2410			0.530	-22.30	-0.00
Camp C	1977	100.1	77.18	298.89	1880	91.50	78.50	0.380	-24.29	-24.35
SiteA	1985	128.6	70.63	324.18	3092	80.85	70.90	0.307	-29.41	-29.41
SiteB	1984	105.6	70.65	322.52	3138	83.70	73.00	0.327	-29.77	-29.48
SiteC	1984	24.9	70.68	321.21	3072			0.340	-29.10	-28.54
SiteD	1984	100.1	70.64	320.38	3018	93.80	81.50	0.365	-28.30	-27.89
SiteE	1985	77.8	71.76	324.15	3087	62.95	53.40	0.225	-30.37	-30.41
SiteF	1985	25.7	71.49	324.12	3092			0.237	-30.42	-30.36
SiteG	1985	70.8	71.15	324.16	3098	69.40	60.50	0.251	-30.10	-30.01
SiteH	1985	26.2	70.87	324.16	3102			0.277	-29.59	-29.53

Table 0.1: Overview of specifications of all examined Greenlandic ice cores.

water isotopic - in this case  $\delta^{18}\text{O}$  - data measured at the same depths. By doing so it is possible to identify known volcanic horizons in the ECM data and thus getting a sharp marker of when the precipitation of that given depth fell. In Figure 0.3 the matched ECM and  $\delta^{18}\text{O}$  profiles can be seen with Tambora marked at depth ---- and Laki at depth ----.

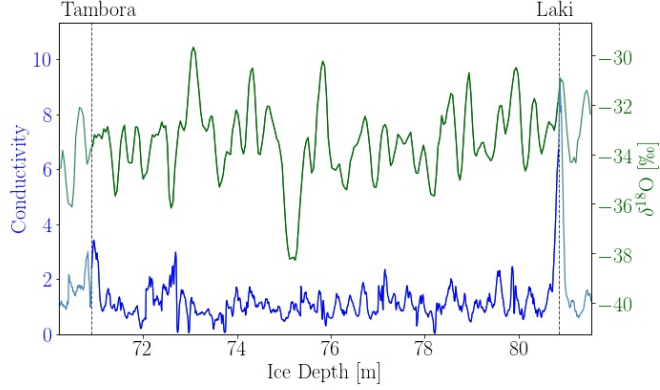


Figure 0.3: Two depth profiles from the core drilled at Site A showing accordingly the measured  $\delta^{18}\text{O}$  isotopic values and the conductivity measurements in the depth ranging from the estimated Tambora eruption to the Laki eruption.

### 0.0.2 Preliminary Computations

From the inputs a number of preliminary computations need to be carried out before the optimization of the diffusion length can be performed. For the depth series this consists of first interpolating the data to make sure that the depth series is evenly sampled, then an analysis of signal and noise in the frequency spectrum needs to be carried out, and from this spectral analysis a Wiener filter can be constructed to find the optimal high frequency cut off, where as much noise as possible is filtered away while losing as little of the signal as possible. These computations are all based on the depth series data and use only signal analysis theory.

The input of core specifications on the other hand is used to give a more theoretical view of the situation. From these preliminary inputs, containing accumulation rate, borehole temperatures, altitudes and other conditions, it is possible to use ice core and ice flow theory to compute initial estimates of density and diffusion profiles at the given site. From these profiles along with the volcanic horizons indicating the series of interest, an initial estimate of the diffusion length at the depth of Laki to Tambora.

Firstly, the computations associated with inputted depth series are considered and later the ones associated with the core specifications are described.

### Depth Series: Spline Interpolation

When working with ice core data, it is not always a certainty that the data you get from the field are as continuous or as evenly sampled as one could have hoped. A lot of factors are at play when drilling, transporting, dividing and cutting an ice core for laboratory studies. Some parts of an ice core contain brittle zones where the ice is likely to break, leaving a section almost impossible to analyze. When transporting the drilled ice cores they are susceptible to both melting and contamination and evaporation of the outermost ice layer of the core. And finally of course there is a number of uncertainties that occur when cutting an ice core into discrete bits to be measured. The quality of data varies rather greatly from site to site, from drill team to drill team and of course from laboratory to laboratory.

When looking at the depth series only it might not be an issue to have

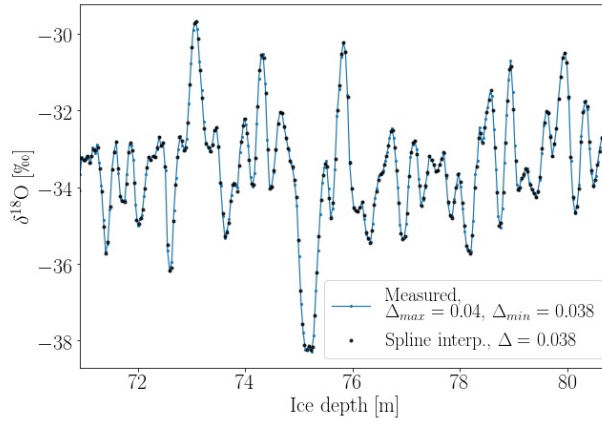


Figure 0.4: The unevenly measured data from Site A in the depth ranging from Tambora to Laki along with the now evenly sampled cubic spline interpolated data. Sampled with an interval of  $\Delta = 0.038$ .

various spacing between the discrete measurements, but when it comes to spectral analysis, it is of necessity to obtain a depth series signal with an even distribution. To accommodate this need interpolation of the signal can be of use. In the case of the ice core drilled at Site A, it can be seen from Figure 0.4 that the raw measured signal has a minimum sampling size of  $\Delta_{\min} = 0.038$  and a maximum sampling size of  $\Delta_{\max} = 0.040$ . To be able to analyze this signal in the spectral domain the signal can be numerically resampled with

an even sampling rate. This is done by choosing the minimum sampling size available,  $\Delta_{\min} = 0.038$ , and redistributing the depth data points to be evenly spaced with this sampling size. The redistribution is carried out as follows.

Assuming the original depth array  $\mathbf{d}$  is distributed as  $d_{i-1} < d_i < d_{i+1}$  with  $i = 0, \dots, n-1$  has a minimum sampling distance as  $\Delta_{\min}$  we define the new sampling distance for the new depth array  $\hat{\mathbf{d}}$  as  $\Delta = \Delta_{\min}$  - again assuming that  $\hat{\mathbf{d}}_{j-1} < \hat{\mathbf{d}}_j < \hat{\mathbf{d}}_{j+1}$  with  $j = 0, \dots, \hat{n}-1$ . This makes it possible to define the first and last value of the new array as

$$\hat{\mathbf{d}}_0 = \Delta \lceil \frac{d_0}{\Delta} \rceil, \quad (1)$$

$$\hat{\mathbf{d}}_{\hat{n}-1} = \Delta \lfloor \frac{d_{n-1}}{\Delta} \rfloor. \quad (2)$$

From this the number of values in the new array,  $\hat{n}$ , can be determined as

$$\hat{n} = 1 + \frac{\hat{\mathbf{d}}_{\hat{n}-1} - \hat{\mathbf{d}}_0}{\Delta}, \quad \hat{n} \in \mathbb{Z}. \quad (3)$$

Thus our new depth array will be given as

$$\hat{\mathbf{d}} = \hat{\mathbf{d}}_0 + j \cdot \Delta, \quad j = 0, \dots, \hat{n} - 1. \quad (4)$$

The original data are then used to define a cubic spline interpolation function to which the redistributed depth data points can be matched. For this part of the data analysis the `SciPy.interpolate` Python (REFERENCE) package with `SciPy.interpolate.CubicSpline` for the cubic spline interpolation. This gives a new distribution of data points as seen in Figure 0.4.

```

1 def interpCores(self, pad = 1):
2
3     # Method in BackDiffuse class
4
5     isoData = self.d180Data
6     d_in = isoData['depth']
7     x_in = isoData['d180']
8
9
10    if self.interpAll:
11        valMin = d_in.min()
12        valmax = d_in.max()
13    else:
14        valMin = self.depthMin - pad
15        valMax = self.depthMax + pad
16
17
```

```

18     d = d_in[(d_in >= valMin) & (d_in <= valMax)]
19     x = x_in[(d_in >= valMin) & (d_in <= valMax)]
20
21     diff = np.diff(d)
22     Delta = round(min(diff), 3)
23
24     d_min = Delta * np.ceil(d.values[0]/Delta)
25     d_max = Delta * np.floor(d.values[-1]/Delta)
26
27     n = int(1 + (d_max - d_min)/Delta)
28
29     j_arr = np.linspace(0,n,n)
30     dhat0 = d_min + (j_arr - 1)*Delta
31
32     f = interpolate.CubicSpline(d,x)
33
34     xhat0 = f(dhat0)
35
36     dhat = dhat0[(dhat0 >= self.depthMin) & (dhat0 <= self.
depthMax)]
37     xhat = xhat0[(dhat0 >= self.depthMin) & (dhat0 <= self.
depthMax)]
38
39     return dhat, xhat, Delta

```

Listing 1: Spline interpolation of  $\delta^{18}O$  data.

## Depth Series: Spectral Analysis

Now with evenly spaced depth series data it is possible to transform the data to the frequency domain and perform spectral analysis on the signal. The basic theory concerning Fast Fourier Transform(FFT), Discrete Cosine Transform(DCT) and Power Spectral Densities(PSD) is discussed in Section ?? . In this section the theory is merely put to use. Since the data under examination is completely real with no imaginary parts, pointing towards the sensibility of working only with cosines to enhance computational speed. Furthermore the DCT implies different boundary conditions than the FFT, meaning that some of the odd spectral behaviour, like aliasing can be avoided by using a different transform.

For the depth series between Laki and Tambora at Site A, the PSD computed through the DCT as  $PSD = |DCT|^2$  can be seen in Figure 0.5a. This spectrum shows a higher intensity in the low frequency area and a lower intensity at high frequencies indicating a low frequency signal and some high frequency noise.

Make a comment on  
Nyquist frequency.

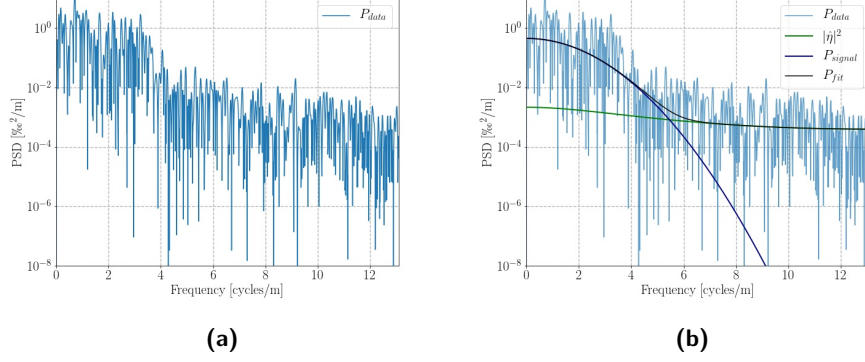
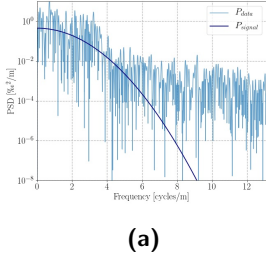


Figure 0.5: Spectral Analysis. **(a)** The power spectral density(PSD) of the interpolated data from Site A shown in Figure 0.4. The PSD was computed through a Discrete Cosine Transform(DCT). **(b)** Spectral estimate of the depth series from Tambora to Laki along with fit to entire spectral data set and the separated estimates of the noise and signal functions, as described in section ??



Since the signal is that of a real physical phenomena, and even one that we do have some acquaintance with, it is relevant to use some knowledge of the theory to describe the apparent spectrum, which has also been done in section ?. Recalling the definitions of the noise and the signal as they are assumed to be based on the physical theory, it is possible to carry out a fitting routine to determine the parameters of the noise, signal and combined model function. This routine is performed in the following steps:

- Define the noise, signal and model functions to fit:

$$|\eta(\omega)|^2 = \frac{\sigma_\eta^2 \Delta z}{|1 + a_1 \exp(-2\pi i \omega \Delta z)|^2} \quad (5)$$

$$P_{\text{signal}}(k) = P_0 \cdot e^{-k^2 \sigma_{tot}^2} \quad (6)$$

$$P_{\text{model}} = |\eta(\omega)|^2 + P_{\text{signal}}(\omega) \quad (7)$$

- Define two methods to calculate residuals between the measured data,  $\mathbf{y}$ , and the modelled estimate  $\mathbf{m}$ . The listings of these methods can be seen in Listings 2 and 3.

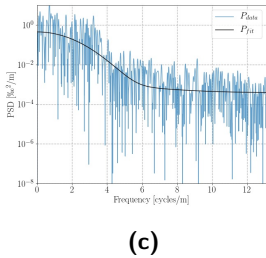


Figure 0.6: **(a)** Signal estimate given through fitting to all data (noise and signal). **(b)** Noise estimate given through fitting to all



```

1 def calc_res(params, x, y, dt, weights):
2     # Set parameters as input parameters
3     P0, s_eta2, s_tot2, a1 = params
4
5     # Define signal and noise function based on given params
6     Noise = self.func_Noise(x, s_eta2, a1, dt)
7     Signal = self.func_Signal(x, P0, s_tot2)
8
9     # Define model as sum of noise and signal
10    Pmod = Noise + Signal
11    # Calculate (weighted) log residual
12    res = weights*(np.log10(y) - np.log10(np.copy(Pmod)))
13
14    return res

```

Listing 2: Residual calculations for spectral fit.

```

1 def sum2_res(params, x, y, dt, weights):
2
3     # Calculate sum of the squared residuals.
4     return np.sum(calc_res(params, x, y, dt, weights)**2)

```

Listing 3: Calculation of the sum of squared residuals.

- Define boundaries for the parameters examined:  $P_0, \sigma_\eta, \sigma_{tot}, a_1$ .
- Make initial parameter guess :
- Optimization routine to minimize residuals. This routine is carried out by using the Python package `sp.optimize.fmin_l_bfgs_b`, an optimization through the limited-memory BFGS bound constraint algorithm .

Write boundaries used - explain why.

Write initial guesses

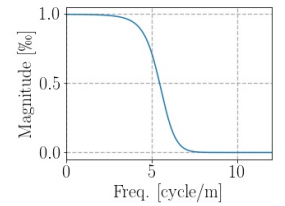
REFERENCE!!!  
Maybe write more?  
No...

The signal, noise and combined fit can be seen separately in Figure 0.6a, 0.6b and 0.6c, respectively, and plotted together in Figure 0.5.

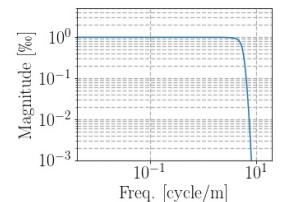
### Depth Series: Wiener Filter

When the optimal fit to the spectral data has been found, it is possible to construct an optimal filter to cut off as much high frequency noise as possible while still maintaining the most signal. This is commonly called a Wiener filter . Recalling from Section ?? this filter is described as the ratio between the signal alone and the signal plus noise model:

$$\tilde{F}(k) = \frac{P_{\text{signal}}(k)}{P_{\text{signal}}(k) + |\tilde{\eta}(k)|^2}. \quad (8)$$



(a)



(b)

Figure 0.7: (a) Wiener filter on linear scale. (b) Wiener filter on double log-arithmetic scale.

This yields a filter that annihilates high frequencies above a certain cut but at the same time leaves room around this cut for both dampening low frequencies and not completely killing high frequencies in that area. This logistic curve can be seen plotted on a linear and a double logarithmic scale in Figures 0.7a and 0.7b respectively.

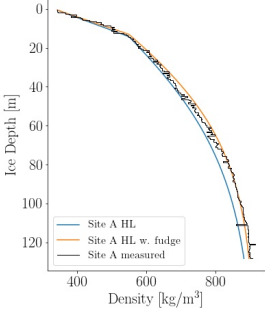


Figure 0.8: Depth density profile at Site A.

### Core Specifications: Density Profile

Right here is something very important.

As well as here.

### Core Specifications: Diffusion Profile

### Core Specifications: $\sigma_0$ estimate

### 0.0.3 Back Diffusion/Deconvolution

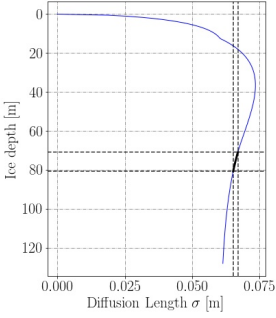


Figure 0.9: Estimated diffusion profile at Site A given a Herron Langway model.

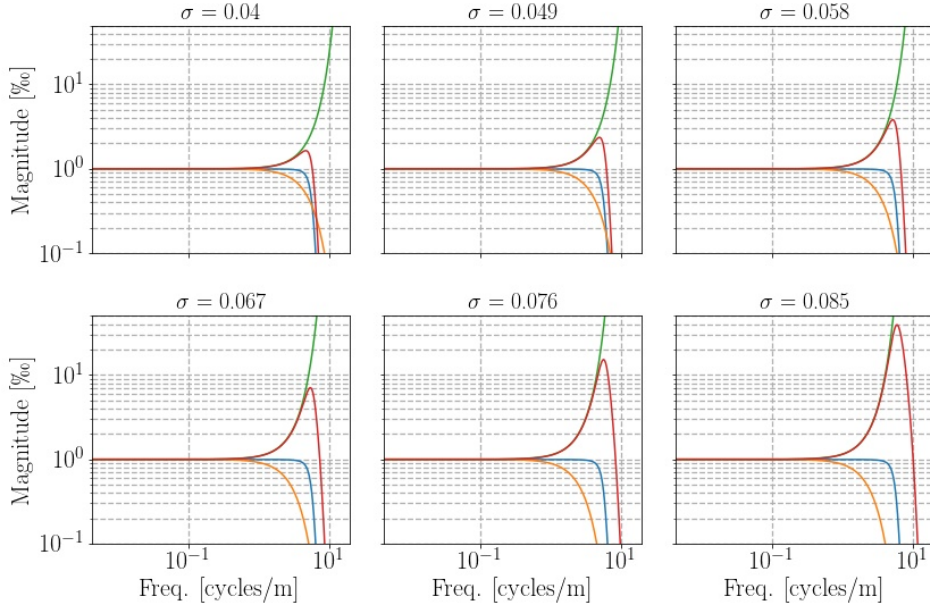


Figure 0.10: Frequency filter examples ranging from diffusion length 0.04 m to 0.085 m.

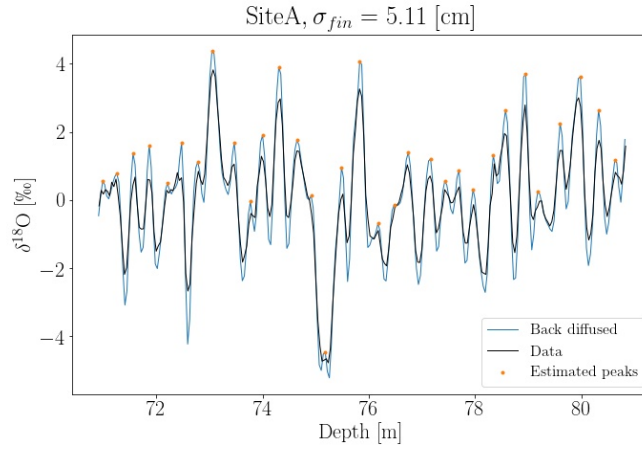


Figure 0.11: Best estimate of deconvoluted depth series given an annual count of 32 years - marked as orange dots. The best estimate is taken as the largest diffusion length that still yields 32 years in the depth span from Tambora to Laki.

#### 0.0.4 Peak Counting

#### 0.0.5 Decision Algorithm

#### 0.0.6 Output

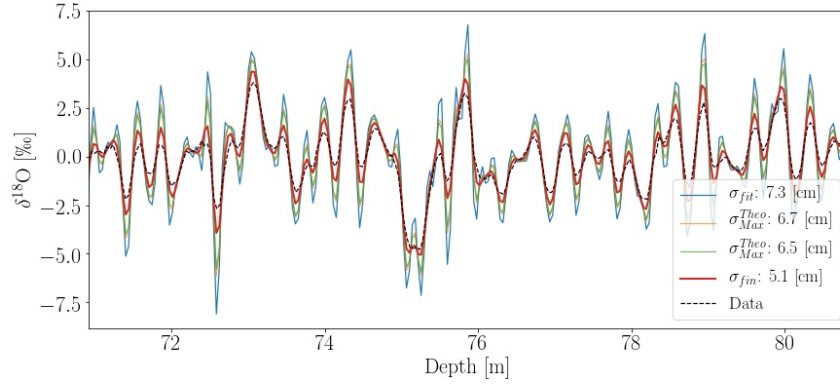


Figure 0.12: Estimated back diffused data series with different diffusion length estimates: diffusion length estimate from spectral fit ( $\sigma_{fit}$ ), maximum ( $\sigma_{Max}^{Theo}$ ) and minimum ( $\sigma_{Min}^{Theo}$ ) theoretically estimated diffusion lengths and final estimated diffusion length.