## 0.1 Estimating $\sigma$ from Data: First Estimate

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.

### 0.1.1 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,

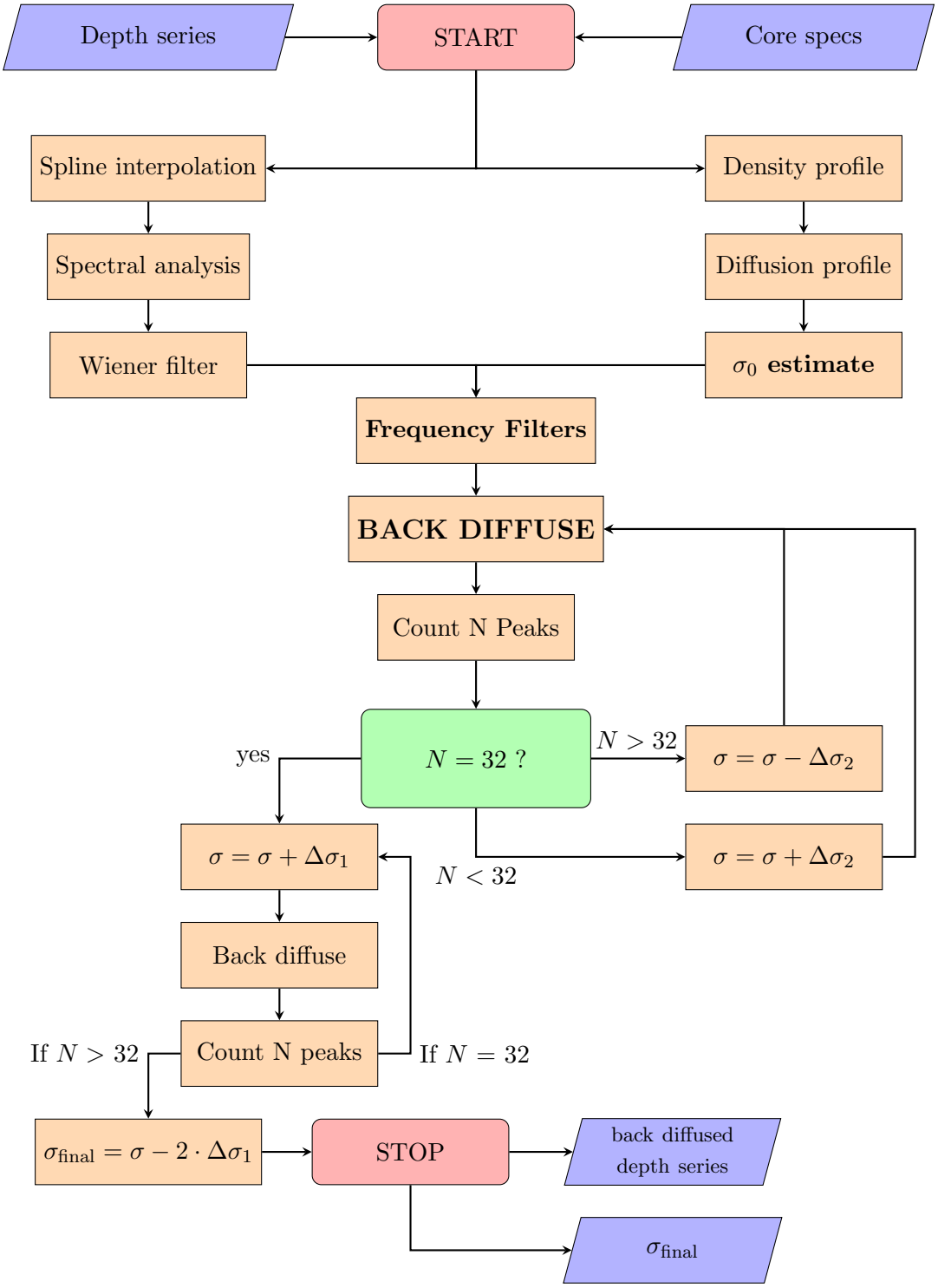



Figure 0.1: Flowchart of method for diffusion length computation.

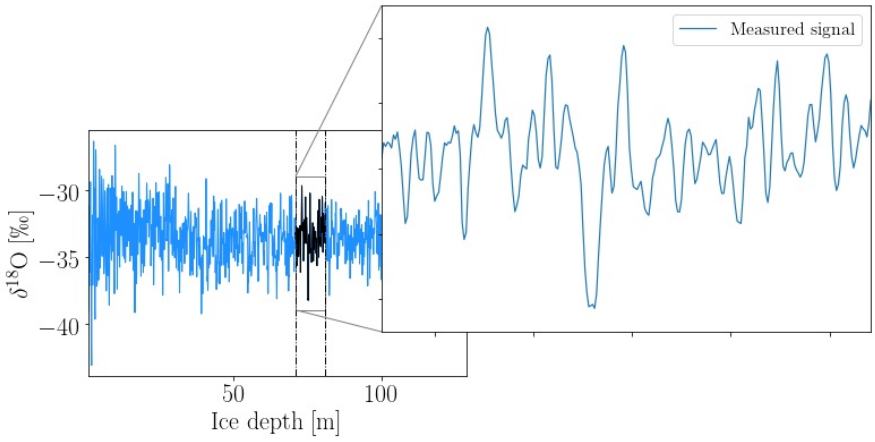



Figure 0.2: The entire ice core isotopic profile from Site A, with a zoom in on the estimated depth series spanning from Tambora to Laki.

| Core | Drilled | Core length | Geographic position | | Elevation | Laki | Tambora | Mean accum. rate | Temp. at 10m | Temp at 20m |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Latitude | Longitude | | | | | | |
| ID | [Yr] | [m] | [°N] | [°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.

measured raw data from Site A. Dating of the ice cores has been carried out by matching Electrical Conductivity Measurements (ECM, Section **??**) with water isotopic - in this case $\delta^{18}$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 $\delta^{18}$O profiles can be seen with Tambora marked at depth ____ and Laki at depth ____.

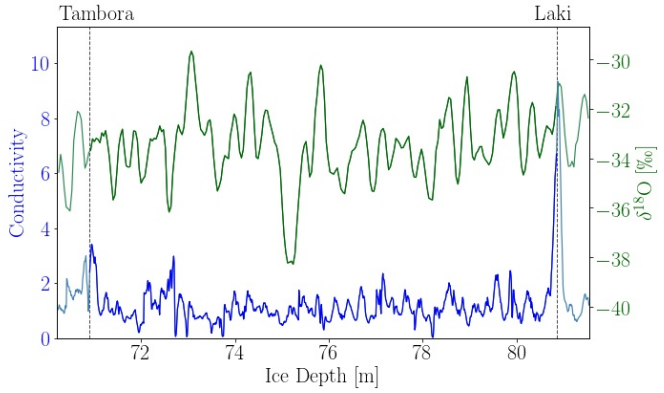



Figure 0.3: Two depth profiles from the core drilled at Site A showing accordingly the measured $\delta^{18}$O isotopic values and the conductivity measurements in the depth ranging from the estimated Tambora eruption to the Laki eruption.

### 0.1.2 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 interolating 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 the core site specifications are considered and later the ones associated with inputted depth series are described.

#### Core Specifications: Density Profile

| Variable | Default |
|---|---|
| $T_0$ | (218.15 K) |
| $\dot{b}_0$ | (0.027 $\frac{\text{m}}{\text{yr}}$) |
| $\rho_{\text{surf}}$ | (330 $\frac{\text{kg}}{\text{m}^3}$) |
| $\rho_{\text{ice}}$ | 917 $\frac{\text{kg}}{\text{m}^3}$ |
| $\rho_{\text{Cr}}$ | 550 $\frac{\text{kg}}{\text{m}^3}$ |
| $\rho_{\text{CO}}$ | 804 $\frac{\text{kg}}{\text{m}^3}$ |
| $f_0^{\text{init}}$ | 1 |
| $f_1^{\text{init}}$ | 1 |
| $z_{\text{max}}$ | - |

From the input core specifications, specifically the surface temperature, $T_0$ and accumulation rate, $\dot{b}_0$, it is possible to estimate a density/depth profile through the Herron-Langway model described in section **??**. This section will contain a brief walk through of the algorithm used to estimate this profile. The Herron-Langway density profile program needs an estimate of surface temperature and annual accumulation rate, but does also take depth-density measurements into the calculations, if they exist and a surface density can also be specified to be taken into account. The adjustable input values and their default values can be seen in Table 0.2. The parameters describe surface temperature, $T_0$, annual accumulation rate, $\dot{b}_0$, surface density, $\rho_0$, density of ice, $\rho_{\text{ice}}$, critical density, $\rho_{\text{Cr}}$, close off density, $\rho_{\text{CO}}$, fit fudge parameters for upper and lower densification zones, $f_0$ and $f_1$, and depth and density measurements, $z_{\text{meas}}$ and $\rho_{\text{meas}}$.

The program is built to estimate a semi-analytical estimate of the depth-density profile given certain surface conditions. Furthermore it returns not only the density profile but computes the derivatives $\frac{d\rho}{dz}$ and $\frac{d\rho}{dt}$, and a timescale estimate for the profile. Two calculations repeats themselves, the computation of the values $k_0$ and $k_1$ and will just be mentioned this once:

$$k_0 = f_0 \cdot 11 \cdot e^{-\frac{10160}{T_0 \cdot R}} \tag{1}$$

$$k_1 = f_1 \cdot 575 \cdot e^{-\frac{21400}{T_0 \cdot R}} \tag{2}$$

The program consists of the following different modules:

- `calc_dens0`: Calculate surface density. Use default surface value, $\rho_{\text{surf}} = 330\frac{\text{kg}}{\text{m}^3}$, if no measurements available. Otherwise calculate surface density by using $z_{\text{init}} = z_{\text{meas}}[0]$ and $\rho_{\text{init}} = \rho_{\text{meas}}[0]$ as:

$$\rho_{\text{surf}} = \frac{\rho_{\text{ice}}}{1 + \frac{\rho_{\text{ice}} - \rho_{\text{init}}}{\rho_{\text{init}}} \cdot e^{z_{\text{init}} \cdot \rho_{\text{ice}} \cdot k_0}} \tag{3}$$

- `calc_zCr`: Calculate ice depth of critical density (i. e. depth of boundary between first and second stage of densification), $\rho_{\text{Cr}} = 550\frac{\text{kg}}{\text{m}^3}$:

$$z_{\text{Cr}} = \frac{1}{\rho_{\text{ice}} \cdot k_0} \cdot \left( \ln\left[\frac{\rho_{\text{Cr}}}{\rho_{\text{ice}} - \rho_{\text{Cr}}}\right] - \ln\left[\frac{\rho_{\text{surf}}}{\rho_{\text{ice}} - \rho_{\text{surf}}}\right] \right) \tag{4}$$

- `model_0`, `model_1`: Models describing the two first stages of densification. Computed as:

$$\rho_0(z) = \rho_{\text{ice}} \cdot \left( \frac{Z_0(z)}{1 + Z_0(z)} \right), \qquad Z_0(z) = e^{\rho_{\text{ice}} \cdot k_0 \cdot z + \ln(\frac{\rho_{\text{surf}}}{\rho_{\text{ice}} - \rho_{\text{surf}}})} \tag{5}$$

$$\rho_1(z) = \rho_{\text{ice}} \cdot \left( \frac{Z_1(z)}{1 + Z_1(z)} \right), \qquad Z_1(z) = e^{\frac{\rho_{\text{ice}} \cdot k_1 \cdot \frac{z - z_{Cr}}{\sqrt{\dot{b}_0}} + \ln(\frac{\rho_{\text{Cr}}}{\rho_{\text{ice}} - \rho_{\text{Cr}}})}} \quad (6)$$

where $\rho_0(z)$ corresponds to all $z \leq z_{\text{Cr}}$ and $\rho_1(z)$ to all $z > z_{\text{Cr}}$.

- `fit_f0`, `fit_f1`: Both modules are only used if any measured data is available. If depth and density data exists then these routines will compute the optimal (least-squares) fudge parameters $f_0$ and $f_1$, see Equations 1 and 2, by fitting the data with the models for $\rho_0$ and $\rho_1$ described above. These updated fudge parameters are then henceforth used in the programs computations.

- `dens_to_depth`: From a given density, calculate the corresponding depth. Same as `calc_zCr`, but for any density, in both stages of densification.

$$z_0(\rho) = \frac{1}{\rho_{\text{ice}} \cdot k_0} \left( \ln \left[ \frac{\rho}{\rho_{\text{ice}} - \rho} \right] - \ln \left[ \frac{\rho_{\text{surf}}}{\rho_{\text{ice}} - \rho_{\text{surf}}} \right] \right), \qquad \rho \leq \rho_{\text{Cr}} \quad (7)$$

$$z_1(\rho) = \frac{\dot{b}_0}{\rho_{\text{ice}} \cdot k_1} \left( \ln \left[ \frac{\rho}{\rho_{\text{ice}} - \rho} \right] - \ln \left[ \frac{\rho_{\text{Cr}}}{\rho_{\text{ice}} - \rho_{\text{Cr}}} \right] \right) + z_{\text{Cr}}, \qquad \rho > \rho_{\text{Cr}} \quad (8)$$

- `drho_dz_0`, `drho_dz_1`: Compute the first derivative with respect to $z$, both stages of densification:

$$\frac{d\rho_0}{dz} = \rho_{\text{ice}}^2 \cdot k_0 \cdot Z_0 \cdot \left[ \frac{1 + 2 \cdot Z_0}{(1 + Z_0)^2} \right], \qquad \rho_0 \leq \rho_{\text{Cr}} \quad (9)$$

$$\frac{d\rho_1}{dz} = \rho_{\text{ice}}^2 \cdot k_1 \cdot Z_1 \cdot \frac{1}{\sqrt{\dot{b}_0}} \cdot \left[ \frac{1 + 2 \cdot Z_1}{(1 + Z_1)^2} \right], \qquad \rho_0 > \rho_{\text{Cr}} \quad (10)$$

with $Z_0$ and $Z_1$ as described in Equations 5 and 6.

- `drho_dt`: Compute the time evolution of the density profile as the derivative of $\rho$:

$$\frac{d\rho_0}{dt} = k_0 \cdot \dot{b}_0 \cdot (\rho_{\text{ice}} - \rho_{\text{surf}}), \qquad \rho_0 \leq \rho_{\text{Cr}} \quad (11)$$

$$\frac{d\rho_1}{dt} = k_1 \sqrt{\cdot \dot{b}_0}, \qquad \rho_1 > \rho_{\text{Cr}} \quad (12)$$

- `time_scale_HL`: From the best optimized Herron-Langway depth-density profile estimate it is possible to compute a age-density profile estimate

for the firn column. For the first stage of densification this is computed as:

$$t_0(\rho) = \frac{1}{k_0 \dot{b}_0} \cdot \ln\left[\frac{\rho_{\text{ice}} - \rho_{\text{surf}}}{\rho_{\text{ice}} - \rho}\right], \qquad \rho_0 \leq \rho_{\text{Cr}} \tag{13}$$

The age of the second stage of densification is computed in the same fashion, and then adding the age at the boundary between the two stages, that is, the critical density, $t_{\text{Cr}}$:

$$t_1(\rho) = \frac{1}{k_1 \sqrt{\dot{b}_0}} \cdot \ln\left[\frac{\rho_{\text{ice}} - \rho_{\text{Cr}}}{\rho_{\text{ice}} - \rho}\right] + t_{\text{Cr}}, \qquad \rho_{\text{Cr}} < \rho < \rho_{\text{ice}} \tag{14}$$

In Figure 0.4 an example of the semi-analytical depth-density modelling can be seen for the core drilled at Site A. It shows the measured depth and density data(black), the purely analytical model(blue) and the semi-analytical fit-to-data-model(orange). The first and second stages of densification can clearly be seen in both data and model as a kink in a depth just shy of 20 m.

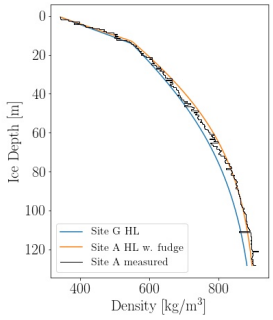



Figure 0.4: Depth density profile at Site A.

**Core Specifications: Diffusion Profile**

**Core Specifications: $\sigma_0$ estimate**

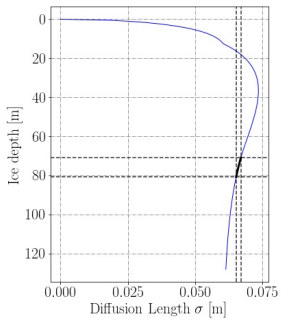



Figure 0.5: Estimated diffusion profile at Site A given a Herron Langway model.

$\sigma(z)$ vs $\sigma_C$

### 0.1.3 Back Diffusion/Deconvolution

**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 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

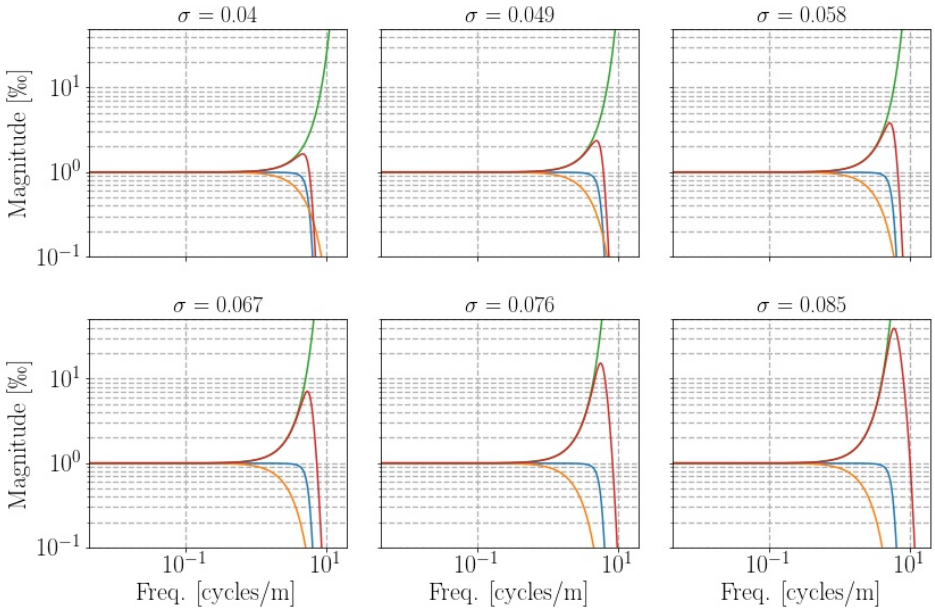



Figure 0.6: Frequency filter examples ranging from diffusion length 0.04 m to 0.085 m.

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 0.7 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 $i = 0, ..., n-1$ 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 $j = 0, ..., \hat{n} - 1$. This makes it possible to define the first and last value of the new array as

$$\hat{\mathrm{d}}_0 = \Delta \lceil \frac{\mathrm{d}_0}{\Delta}, \rceil \tag{15}$$

$$\hat{\mathrm{d}}_{\hat{n}-1} = \Delta \lfloor \frac{\mathrm{d}_{n-1}}{\Delta} \rfloor. \tag{16}$$

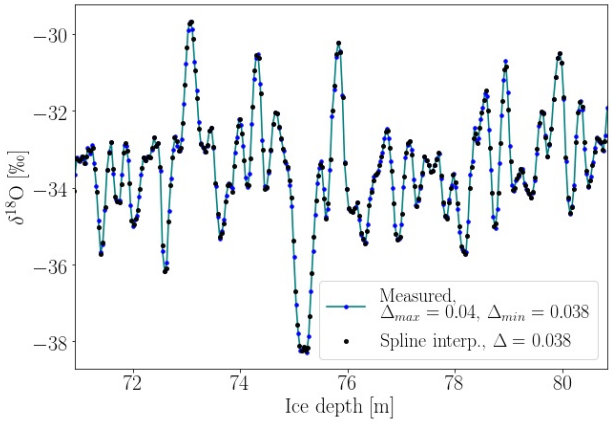



Figure 0.7: The unevenly measured data from Site A in the depth ranging from Tambora to Laki along with the now evenly sampled cubic spline interpolated data. Sampled with an interval of $\Delta = 0.038$.

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

$$\hat{n} = 1 + \frac{\hat{\mathrm{d}}_{\hat{n}-1} - \hat{\mathrm{d}}_0}{\Delta}, \qquad \hat{n} \in \mathbb{Z}. \tag{17}$$

Thus our new depth array will be given as

$$\hat{\mathbf{d}} = \hat{\mathrm{d}}_0 + j \cdot \Delta, \qquad j = 0, ..., \hat{n} - 1. \tag{18}$$

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 0.7.

```
def interpCores(self, pad = 1):

    # Method in BackDiffuse class

    isoData = self.d18OData
    d_in = isoData['depth']
    x_in = isoData['d18O']


    if self.interpAll:
```

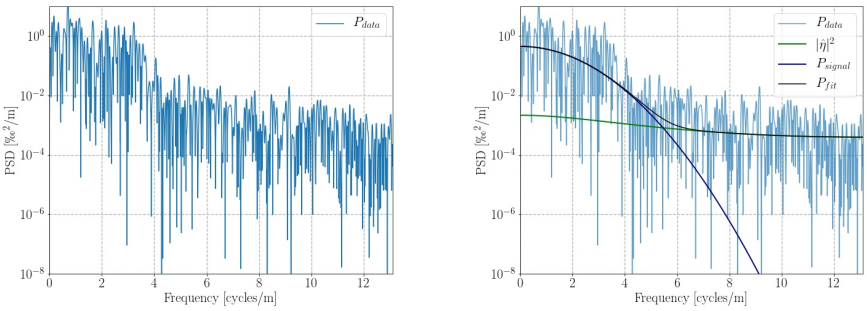
```
        valMin = d_in.min()
        valmax = d_in.max()
    else:
        valMin = self.depthMin - pad
        valMax = self.depthMax + pad


    d = d_in[(d_in >= valMin) & (d_in <= valMax)]
    x = x_in[(d_in >= valMin) & (d_in <= valMax)]

    diff = np.diff(d)
    Delta = round(min(diff), 3)

    d_min = Delta * np.ceil(d.values[0]/Delta)
    d_max = Delta * np.floor(d.values[-1]/Delta)

    n = int(1 + (d_max - d_min)/Delta)

    j_arr = np.linspace(0,n,n)
    dhat0 = d_min + (j_arr - 1)*Delta

    f = interpolate.CubicSpline(d,x)

    xhat0 = f(dhat0)

    dhat = dhat0[(dhat0 >= self.depthMin) & (dhat0 <= self.
    depthMax)]
    xhat = xhat0[(dhat0 >= self.depthMin) & (dhat0 <= self.
    depthMax)]

    return dhat, xhat, Delta
```

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

### Implementation of Spectral Transforms

### 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




**(a)** **(b)**

Figure 0.8: Spectral Analysis. **(a)** The power spectral density(PSD) of the interpolated data from Site A shown in Figure 0.7. 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 **??**

of the odd spectral behaviour, like aliasing can be avoided by using a different transform.

METH-SPECT: Make a comment on Nyquist frequncy.

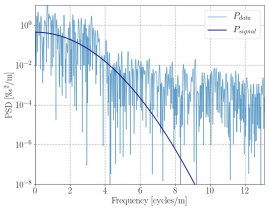
For the depth series between Laki and Tambora at Site A, the PSD computed through the DCT as PSD = $|\text{DCT}|^2$ can be seen in Figure 0.8a. This spectrum shows a higher intensity in the low frequency area and a lower intensity at high frequencies indicating a low frequency signal and some high frequency noise.

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:

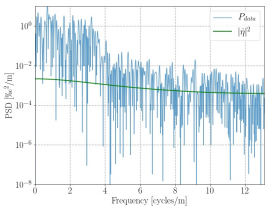
- Define the noise, signal and model functions to fit:

$$|\eta(\omega)|^2 = \frac{\sigma_\eta^2 \Delta z}{|1 + a_1 \exp(-2\pi i \omega \Delta z)|^2} \tag{19}$$

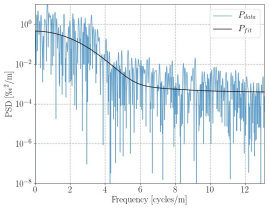
$$P_{\text{signal}}(k) = P_0 \cdot e^{-k^2 \sigma_{tot}^2} \tag{20}$$



**(a)**



**(b)**



**(c)**

Figure 0.9: (a) Signal co

$$P_{\text{model}} = |\eta(\omega)|^2 + P_{[\text{signal}]}(\omega) \tag{21}$$

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

```
def calc_res(params, x, y, dt, weights):
    # Set parameters as input parameters
    P0, s_eta2, s_tot2, a1 = params

    # Define signal and noise function based on given params
    Noise = self.func_Noise(x, s_eta2, a1, dt)
    Signal = self.func_Signal(x, P0, s_tot2)

    # Define model as sum of noise and signal
    Pmod = Noise + Signal
    # Calculate (weighted) log residual
    res = weights*(np.log10(y) - np.log10(np.copy(Pmod)))

    return res
```

Listing 2: Residual calculations for spectral fit.

```
def sum2_res(params, x, y, dt, weights):

    # Calculate sum of the squared residuals.
    return np.sum(calc_res(params, x, y, dt, weights)**2)
```

Listing 3: Calculation of the sum of squared residuals.

METH-SPECTFIT: Write boundaries used - explain why.

- Define boundaries for the parameters examined: $P_0$, $\sigma_\eta$, $\sigma_{tot}$, $a_1$.

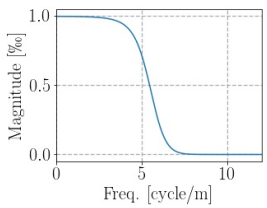
METH-SPECTFIT: Write initial guesses

- Make initial parameter guess :

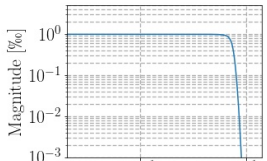
METH-SPECTFIT: REFERENCE!!! Maybe write more? No...

- Otimization 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 .

The signal, noise and combined fit can be seen separately in Figure 0.9a, 0.9b and 0.9c, respectively, and plotted together in Figure 0.8.




**(a)**




### 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

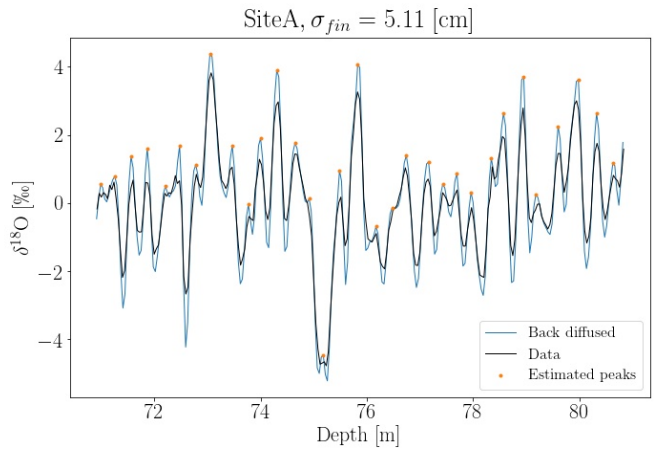



Figure 0.11: Best estimate of deconvoluted depth series given an annual count of 32 years - marked as orange dots. The best estimate is taken as the largest diffusion length that still yields 32 years in the depth span from Tambora to Laki.

filter . Recalling from Section **??** this filter is described as the ratio between the signal alone and the signal plus noise model:

$$\tilde{F}(k) = \frac{P_{\text{signal}}(k)}{P_{\text{signal}}(k) + |\tilde{\eta}(k)|^2}. \tag{22}$$

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 0.10a and 0.10b respectively.

Now, the one branch of the preliminary work considering the raw depth series, has been dealt with and the remaining work before back diffusion can be carried out consisting of using the specifications of the core at the given drill site to give a qualified first guess on diffusion length.

METH-WIENER: REFERENCE!!!

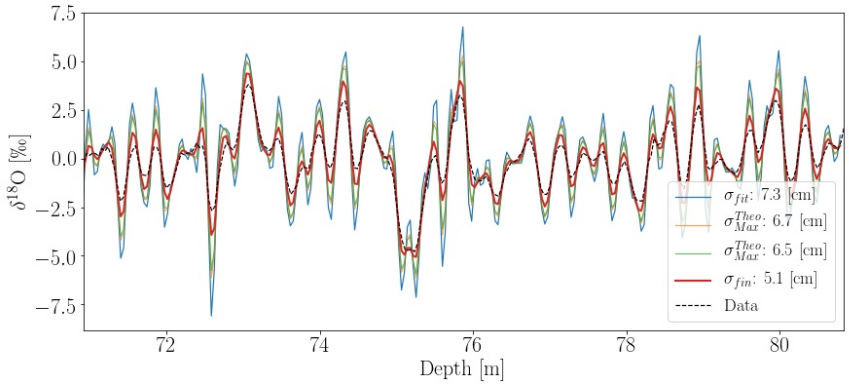



Figure 0.12: Estimated back diffused data series with different diffusion length estimates: diffusion length estimate from spectral fit ($\sigma_{fit}$), maximum ($\sigma^{Theo}_{Max}$) and minimum ($\sigma^{Theo}_{Min}$) theoretically estimated diffusion lengths and final estimated diffusion length.

### 0.1.4 Peak Counting

**Spline Interpolation for Optimal Peak Detection**

## 0.2 Estimating $\sigma$ from Data: Optimal Diffusion Length

### 0.2.1 Decision Algorithm

### 0.2.2 Output

METH-OUT: Remember: write about sampling sigma

## 0.3 Method Testing and Stability

METH-TESTS: Write entire section.

### 0.3.1 Diffusion Length Estimate Versus Number of Counted Peaks

METH-TESTMAX: Write entire section.

### 0.3.2 Interpolation of Data Before Deconvolution

METH-TESTINT1: Write entire section.

### 0.3.3 Interpolation of Data After Deconvolution

METH-TESTINT2: Write entire section.

## 0.4 Upgrading the Algorithms

METH-UPGRADE: Write entire section.

### 0.4.1 Peak Detection through Machine Learning

METH-UPGRADEPEAK: Do the ML Peak Detection, then Write entire section.

### 0.4.2 Linear Timescale

METH-UPGRADETIME: Incorporate Linear Timescale in Recursivity - then Write entire section.

### 0.4.3 Standardization and Detrending

METH-UPGRADESTAND: Maybe not necessary?Write entire section.

### 0.4.4 Recursivity and New Constraints

METH-UPGRADERECURS: Develop recursive algorithm with new constraints for peak finding - then write entire section.

## 0.5 Reconstruction of Missing Data

### 0.5.1 Interpolation Method

METH-MISSDATA: Write entire section. Maybe not necessary...

### 0.5.2 Maximum Entropy Method

METH-MISSDATAINTERP: Write entire section. Show baddddd figures.