

The data obtained through various experimental measurements are easily compared with a time series, as they typically show some quantity measured all along the depth of an ice core. This depth is often, at short intervals, treated as a regular linear time series thus making it possible to use some of the known signal analysis methods. Of course, when considering the entirety of an ice core, the linearity disappears as thinning and compression makes the depth series non linear. But when considering short lengths of core it is possible to estimate a linearity, assuming conformity in this specific layer.

## 0.1 Back Diffusion

Due to diffusion in firn and ice, some of the water isotopic signal is lost. Some of this signal can be restored by investigating the diffusion process, and through filtering and deconvolution techniques(REFERENCES). For the data of this thesis two different restoration techniques are presented: a spectral method, determining the effect of mixing and diffusion as a spectral filter(REFERENCES), and a kernel restoration method much like the ones used to restore pixel resolution in images (REFERENCES).

### 0.1.1 Spectral Analysis

#### Power Spectral Densities

A very useful tool for analyzing signals exhibiting oscillatory effects is analysis of the signals power spectrum. Instead of considering the signal in time, it is transformed to the spectral domain, where it is possible to obtain an estimate of both the signal and the underlying noise. This is crucial for enhancing the signal and filtering away noise. But to be able to examine these effects, first the data must be transformed. A range of different methods may be used to compute the frequency transform of the depth series, here I present the three I have been working with. Since the data are discrete and experimental, I will be presenting the discrete and applicable mathematical models.

When considering a signal, it may be of interest to investigate how the energy of said signal is distributed with frequency. The total power is defined as:

$$\text{Total Power} = \int_{-\infty}^{\infty} |X(\tau)|^2 d\tau. \quad (1)$$

Using Parseval's theorem (REFERENCE) (assuming that the signal has a finite total energy), the power of the signal can alternatively be written as

$$\int_{-\infty}^{\infty} |X(\tau)|^2 d\tau = \int_{-\infty}^{\infty} |\tilde{X}(\tau)|^2 df \quad (2)$$

where  $\tilde{X}(f)$  is the spectral (Fourier) transform of the signal, from time to frequency domain, defined as:

$$\tilde{X}(f) = \int_{-\infty}^{\infty} X(\tau) e^{2\pi i f \tau} d\tau \quad (3)$$

and the inverse spectral (Fourier) transform, from frequency to time domain, defined as:

$$X(t) = \int_{-\infty}^{\infty} \tilde{X}(f) e^{-2\pi i f \tau} df. \quad (4)$$

Both  $X(t)$  and  $\tilde{X}(f)$  represent the same function, just in different variable domains. Often, the angular frequency  $\omega$  is used instead, with the relation between  $\omega$  and  $f$  being  $\omega \equiv 2\pi f$ , giving the Fourier and inverse Fourier transforms as:

$$\begin{aligned} \tilde{X}(\omega) &= \int_{-\infty}^{\infty} X(t) e^{i\omega\tau} d\tau \\ X(\tau) &= \int_{-\infty}^{\infty} \tilde{X}(\omega) e^{-i\omega\tau} d\omega \end{aligned} \quad (5)$$

From Equation 2 we can interpret the integrand on the right hand side  $|\tilde{X}(f)|^2$  as a density function, describing the energy per unit frequency. This is a property which is able to reveal much information about the considered signal, and it is useful to define this as the (one-sided) Power Spectral Density:

$$P_X(f) \equiv |\tilde{X}(f)|^2 + |\tilde{X}(-f)|^2 \quad 0 \leq f < \infty \quad (6)$$

This entity ensures that the total power is found just by integrating over  $P_X(f)$  from 0 to  $\infty$ . When the function is purely real, the PSD reduces to  $P_X(f) = 2|\tilde{X}(f)|^2$ .

In the above the transform used to define the PSD was presented as the Fourier transform. When working with discrete data, as is very common when analyzing real world data, there are a number of different ways of estimating the PSD. In the following three different methods will be presented, all used in this thesis.

**RETHINK THIS PART. DO NOT USE TIME ON ALL THE CALCULATIONS. WRITE THE GENERAL IDEAS**

**OF THE METHODS AND STATE HOW TO CALCULATE/COMPUTE. SMALL CODE SNIP TO GIVE GENERAL IDEA.**

### Discrete and Fast Fourier Transform

The definition of the continuous Fourier transform and its inverse was presented in the above. The Fourier transform is as seen a way of representing the function under consideration as an infinite sum of periodic components. When the function is discrete, so will the Fourier transform be, and the integral is replaced with a sum. This gives us the Discrete Fourier Transform (DFT) which transforms the signal into a sum of separate components contributing at different frequencies. The DFT is dependent on the sampling interval,  $\Delta$ , and we can describe our discrete signal  $X$  as a function of  $N$  discrete time steps  $t_k = k \cdot \Delta$ , where  $k = 0, 1, \dots, N - 1$ :

$$X_k \equiv X(t_k) \quad (7)$$

This sample size is supposed to be representative for the entire discrete function, if the function continues beyond the  $N$  sampled points. When sampling discretely at interval  $\Delta$ , there will be a special frequency, the Nyquist critical frequency, defined through the sampling size as:

$$f_{NQ} \equiv \frac{1}{2\Delta}. \quad (8)$$

This frequency is of great importance in transformation of discrete signals. If the continuous signal is sampled at an interval  $\Delta$  is bandwidth limited to frequencies smaller in magnitude than  $f_{NQ}$ ,  $\tilde{X}(f) = 0$  for  $|f| \geq f_{NQ}$  - i.e. the transformed function has only non-zero values inside the Nyquist interval,  $\tilde{X}(-f_{NQ}), \dots, \tilde{X}(f), \dots, \tilde{X}(f_{NQ})$ . This means that the function is completely determined since we have all information about the signal contained in our available frequency space.

On the other hand, which is much more likely, if the continuous signal consists of frequencies both inside and outside the Nyquist interval, then all spectral information outside of this range will be falsely interpreted as being inside this range. Thus a wave inside the interval with a frequency of  $f_n$  will have a number of wave siblings outside of the interval, with frequencies of  $k \cdot \frac{1}{\Delta} f_n$ ,  $k$  being integers, which will be aliased into the Nyquist interval and give rise to an increased power at the frequency  $f_n$ .

When analyzing an already measured discrete signal, this might give rise to

some headache. What can be done is to assume that the signal has been sampled competently and then assume that the Fourier transform is zero outside of the Nyquist interval. After the analysis it will then be possible to determine if the signal was indeed competently sampled, as the Fourier series will go to zero at  $f_{NQ}$  given a correct assumption, and go to a fixed value, if the sampling was not done competently.

Now with the basics of understanding the limits of frequency transform of a discretely sampled signal, it is possible to estimate the DFT of the signal  $X_k \equiv X(t_k)$ . Since the Fourier transform is a symmetric transformation it is easiest to assume that  $N$  is even.

Since the input information is of size  $N$  we should expect only to sample the frequency transform  $\tilde{X}(f)$  at only discrete values of  $f$  in the range between the upper and lower critical Nyquist frequencies,  $-f_{NQ}$  to  $f_{NQ}$ :

$$f_n \equiv \frac{n}{N\Delta}, \quad n = -\frac{N}{2}, \dots, \frac{N}{2} \quad (9)$$

This will indeed actually give rise to  $N + 1$  values, since 0 will be in the interval as well, but the limit frequencies are actually not independent, but all frequencies between are, which reduces it to  $N$  samples.

Now the integral from Equation 3 needs to be estimated as a sum:

$$\tilde{X}(f_n) = \int_{-\infty}^{\infty} X(\tau) e^{2\pi i f_n \tau} d\tau \approx \sum_{k=0}^{N-1} X_k e^{2\pi i f_n t_k} \Delta = \Delta \sum_{k=0}^{N-1} X_k e^{2\pi i k \frac{n}{N}} \quad (10)$$

The Discrete Fourier Transform is thus defined as:

$$\tilde{X}_n \equiv \sum_{k=0}^{N-1} X_k e^{2\pi i k \frac{n}{N}} \quad (11)$$

This gives the approximate relation between the DFT estimate and the continuous Fourier transform  $\tilde{X}(f)$  when sampling at size  $\Delta$  as:

$$\tilde{X}(f_n) \approx \Delta \tilde{X}_n \quad (12)$$

The inverse DFT is given as:

$$X_n \equiv \frac{1}{N} \sum_{n=0}^{N-1} X \tilde{X}_n e^{-2\pi i k \frac{n}{N}} \quad (13)$$

Computation of the DFT can be very slow and tiresome, since it involves complex multiplication between a number of vectors and matrices. If we write

Equation 11 as  $\tilde{X}_n = \sum_{k=0}^{N-1} W^{nk} X_k$ , where  $W$  is a complex number  $W \equiv e^{2\pi i/N}$ . This shows that the vector  $X_k$  must be multiplied with a complex matrix which (n,k)th component consists of the constant  $W$  to the power of  $nk$ . This matrix multiplication evidently leads to a process of  $O(N^2)$ . Fortunately, a number of different algorithms(REFERENCES) have been developed for fast and efficient computation of the discrete Fourier transform. One of these is called the Fast Fourier Transform (FFT), which can reduce the computations to just  $O(N \log_2 N)$ ! In this thesis the FFT used is the one implemented in the numpy.fft Python package(REFERENCES) which is based on the works of (REFERENCES). See said article for implementation details. One important thing about this specific algorithm is that for the algorithm to function most efficiently, the number of points computed in the frequency space must be of a power of 2, following the use of base  $\log_2$

### Discrete Cosine Transform

The DFT is generally defined for complex inputs and outputs, where the sine components of the transform describe the complex part and the cosine describes the real part.

Since the data analyzed in this thesis is purely real, it makes sense, for computational speed, to only work with the cosine parts of the transform. This leads to the Discrete Cosine Transform (DCT) which transforms real inputs to real outputs.

#### 0.1.2 Spectral Filtering

##### Wiener Filtering

Through spectral analysis it is possible to treat the noise of the signal consistently. The goal is to create spectral filters which enhances the signal while minimizing the effect of the noise, thus increasing the signal-to-noise ratio (SNR).

Theoretically, without any diffusion, the change in isotopic concentration would be described through a step function, going from one constant concentration to another. This step function can be described by the Heaviside function:

$$D(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \quad (14)$$

In reality, a number of different mixing processes change this step function, and the measured signal will be a smooth curve,  $s(t)$ , which corresponds to

the convolution of  $S(t)$  with the mixing response function  $M(\tau)$

$$d(t) = \int_{-\infty}^{\infty} D(\tau) \cdot M(t - \tau) d\tau \quad (15)$$

### 0.1.3 Signal Restoration by Optimal Diffusion Length

#### Kernel Estimation

As is well known, in the spectral domain, convolution is multiplication and the mixing is described as the multiplication between the Fourier transform of  $S$  and  $M$ :

$$\tilde{d} = \tilde{D} \cdot \tilde{M} \quad (16)$$

By differentiation with respect to time, the mixing filter  $M$  is unaffected, and differentiation of the measured system response, the Heaviside function,  $S'$  is a delta function, which Fourier transformed is unity, leading to:

$$\tilde{d}' = \tilde{D}' \cdot \tilde{M} = \tilde{M} \quad (17)$$

The mixing filter can thus be determined by measuring the system response to a step function, differentiating performing Fourier transform of the result  $d'$ .

After determination of the mixing filter  $\tilde{M}$ , the unmixed signal  $D$  can be estimated in theory by inverse Fourier transform of

$$\tilde{D} = \tilde{d} \cdot \tilde{M}^{-1} \quad (18)$$

During the mixing, cycles with short wavelengths are heavily washed out, and through the restoration in Eq. 18, the amplitudes corresponding to these wavelengths are heavily amplified by the filter. This method though has a drawback, which is that when the measurements contain noise, the restored signal will be dominated by high-frequency noise, greatly amplified by the mixing filter. Thus it is a problem of retaining as much (short wavelength) signal as possible while simultaneously attempting to amplify the high-frequency noise as little as possible. This optimal trade-off can be found by creating an optimum filter for the considered measured isotopic signal:

$$\delta_M(z) = \delta_m(z) + \eta(z) \quad (19)$$

This optimal (Wiener) filter  $\tilde{F}$ , defined for each wave number  $k = 2\pi\omega$ , is presented as the ratio between pure signal and pure signal plus noise described

in Power Spectral Densities as:

$$\tilde{F}(k) = \frac{|\tilde{\delta}_m(\omega)|^2}{|\tilde{\delta}_m(\omega)|^2 + |\tilde{\eta}(\omega)|^2} \quad (20)$$

In this work, the power spectral densities of the signal and the noise, respectively, are determined through analysis of the power spectral density of the combined signal/noise PSD.

The PSD of the noise free measured signal,  $|\tilde{\delta}_m(\omega)|^2$ , is assumed describe as

$$|\tilde{\delta}_m(\omega)|^2 = P_0 e^{-k^2 \sigma_{\text{tot}}^2} \quad (21)$$

where  $\sigma_{\text{tot}}^2$  describes the total estimated diffusion length of the mixing.

The noise is assumed to be red noise, described by an autoregressive process of first order, AR1:

$$|\tilde{\eta}(\omega)|^2 = \frac{\sigma_\eta^2 \Delta z}{|1 + a_1 \exp(-2\pi i \omega \Delta z)|^2} \quad (22)$$

where  $\sigma_\eta^2$  is the variance of the red noise,  $a_1$  is the AR1 coefficient and  $\Delta z$  is the resolution of the time/depth data. It is then possible to estimate the parameters  $P_0$ ,  $\sigma_{\text{tot}}^2$ ,  $\sigma_\eta^2$  and  $a_1$  by curve fitting, separately, the two expressions in Eq. 21 and 22 to the data. The estimated parameters are varied to find the optimal guess to use for the filter.

## 0.2 Enhanced Resolution and Peak Detection

### 0.2.1 Interpolation

Interpolation is a tool that can be used - and misused - to extract more information out of a given set of data. Used correctly, interpolation can reveal more information than is initially available and disclose connections not apparent at first, but used incorrectly, it can be manipulated to infer misleading correlations and lead to inaccurate conclusions. Thus it is a tool that must be used with care. Aiming to avoid incorrect deductions and inferences one should at first gain as much knowledge about the data at hand as possible. By understanding how the data have come about and gaining knowledge about the underlying physical theories a somewhat deficient data set can robustly and securely be interpolated to accommodate the needs of the analysis. In the case of this thesis, both knowledge about data gathering and the physics at play have been gained and thus some of the common fallacies may be avoided.

The limits of the data available is due to the discrete sampling, leading to a minimum sampling of about 26 samples per meter of ice. When considering that the depth series of 32 years between Tambora and Laki is just above 10 meters, this means that each meter of ice needs to contain at least three years on average. 26 samples per three years might not sound as a bad sampling interval, but if the goal is to show seasonality and give a best estimate of annual layer thickness, interpolation could be put to good use to be able to give better estimates of the exact placement of peaks and valleys.

### Existence, Uniqueness and Conditioning

Considering any attempt to create an interpolant to fit a number of data points, the questions of uniqueness and existence is a matter of matching the data points with the number of parameters in the interpolant. If there are too few parameters, the interpolant does not exist, as it will not pass through all data points. If there are too many, the interpolant will not be unique. Formally this can be described through a system of linear equations.

For any data set consisting of  $(t_i, y_i)$ ,  $i = 1, \dots, m$  points, an interpolant can be chosen from a function space spanned by some suitable set of basis functions,  $\phi_1(t), \dots, \phi_n(t)$ . The interpolant can then be described as a linear combination of these basis functions:

$$f(t) = \sum_{j=1}^n x_j \phi_j(t) \quad (23)$$

The interpolant can then be found by determining the parameters  $x_j$  by requiring that the interpolant  $f$  must pass through the  $M$  data points  $(t_i, y_i)$ :

$$f(t_i) = \sum_{j=1}^n x_j \phi_j(t_i) = y_i, \quad i = 1, \dots, m \quad (24)$$

This can of course also be written compactly in matrix form as a system of linear equations:

$$\mathbf{A}\mathbf{x} = \mathbf{y} \quad (25)$$

In this equation  $\mathbf{A}$  is the  $m \times n$  basis matrix, which entries consists of the value of the  $n$  basis functions evaluated at the  $m$  data points,  $a_{ij} = \phi_j(t_i)$ , the  $m$  vector  $\mathbf{y}$  consists of the known data values  $y_i$ , and the  $n$  vector  $\mathbf{x}$  consists of the unknown, to be determined, parameters  $x_j$ .

From linear algebra we know, that if we choose the number of basis function ot be equal to the number of data points,  $n = m$ , the basis matrix will be square, and thus - given the matrix is nonsingular - the system will be determined, and the data points can be fit exactly. Though in some problems it is beneficial



to choose the system to be either overdetermined (less parameters than data points, the data cannot be fit exactly) or underdetermined (more parameters than data points, giving freedom to allow satisfaction of additional properties or conditions).

So the existence and uniqueness of an interpolant is given by the non-singularity of the basis matrix, be it square or not and the conditioning of the matrix points to the parameters' sensitivity to perturbations. An ill-conditioned basis matrix will lead to high sensitivity in the parameters, but this problem can still be approximately solvable through Gaussian elimination with partial pivoting, but this solution will mean that the coefficients may be poorly determined.

### Polynomial Interpolation

The most common way to determine an interpolant is through polynomials. Denoting a set of all polynomials of degree at most  $k$ ,  $k \geq 0$  as  $\mathbb{P}_k$ , it can be seen that this set forms a vector space of dimension  $k+1$ . The basis functions that span this vector space can be chosen to be composed of a number of different functions and this choice has a great influence on both the cost of computation and manipulation of the interpolant, and the sensitivity of the parameters, i.e. the conditioning of the basis matrix.

Considering  $n$  data points it is obvious to choose  $k = n - 1$  so that the dimension of the vector space matches the number of data points. The maybe most natural choice of basis for  $\mathbb{P}_{n-1}$  is one that consists of the first  $n$  monomials<sup>1</sup>,

$$\phi_j(t) = t^{j-1}, \quad j = 1, \dots, n. \quad (26)$$

Thus any given polynomial  $p_{n-1} \in \mathbb{P}_{n-1}$  will be of the form

$$p_{n-1}(t) = x_1 + x_2 t + \dots + x_n t^{n-1}. \quad (27)$$

In this basis the system of  $n \times n$  linear equations will be of the form

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ 1 & t_1 & \dots & t_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_1 & \dots & t_1^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{y}. \quad (28)$$

This type of matrix with geometric progression, i.e. the columns are successive powers of some independent variable  $t$  is called a Vandermonde matrix.

REFERENCE!!!

<sup>1</sup>Roughly speaking, a polynomial with only one term.

Maybe an illustration here? See pp. 314 in Scientific Computing.

When using the monomial basis and using a standard linear equation solver to determine the interpolant coefficients requires  $\mathcal{O}(n^3)$  work and often results in ill-conditioned Vandermonde matrices  $\mathbf{A}$ , especially for high-degree polynomials. This ill-conditioning is due to the monomials of higher and higher degree being more and more indistinguishable from each other. This makes the columns of  $\mathbf{A}$  nearly linearly dependent, resulting in almost singular matrices, and thus highly sensitive coefficients. For high enough  $n$ , the Vandermonde matrix becomes efficiently singular, to computational precision at least, though, as mentioned, this can be worked around, but requires some additional computational work.

### Piecewise Polynomial Interpolation and Splines

REFERENCES!!

The amount of work needed to solve the system as well as the conditioning of the system can be improved by using a different basis all together. Some different bases superior to the monomial that are worth mentioning are the Lagrange basis functions, the Newton basis functions and the orthogonal polynomials. But for this thesis we take a step further into the interpolation theory, as the choice of basis functions might not be enough to work around some of the problems connected with fitting a single polynomial to a large number of data points (i.e. oscillatory behaviour in the interpolant, nonconvergence or issues around the boundaries).

These practical and theoretical issues can be avoided through the use of piecewise polynomial interpolation, with the advantage that a large number of data points can be fitted with low-degree polynomials.

When turning to piecewise polynomial interpolation of the data points  $(t_i, y_i)$ ,  $i = 1, \dots, n$ ,  $t_1 < t_2 < \dots < t_n$ , a different polynomial is chosen for each subinterval  $[t_i, t_{i+1}]$ . Each point  $t_i$ , where the interpolant changes is called knots or control points. The simplest piecewise interpolation is piecewise linear interpolation, where each knot is connected with a straight line. If we consider this simple example it appears that by eliminating the problems of nonconvergence and unwanted oscillatory behaviour, the smoothness of the interpolant is sacrificed. This might be true for this simplistic example but since there are a number of degrees of freedom in choosing each piecewise polynomial interpolant, the smoothness can be reintroduced by exploiting a number of these measures. One way of doing this is by demanding knowledge of both the values and the derivatives of the interpolant at each data point. This just adds more equations to the system, and thus to have a well-defined solution, the number of equations must match the number of parameters. This type of interpolation is known as Hermite interpolation. The most common choice

for this interpolation, to still maintain simplicity and computational efficiency, is cubic Hermite interpolation. This introduces a piecewise cubic polynomial with  $n$  knots, and thus  $n - 1$  interpolants each with 4 parameters to fit, leading to  $4(n - 1)$  parameters to be determined. Since each of the  $n - 1$  cubics must match the data points at each end of the subinterval, it results in  $2(n - 1)$  equations, and requiring the derivative to be continuous, i.e. match at the end points, an additional of  $n - 2$  equations are taken in. This leads to a system consisting of  $2(n - 1) + (n - 2) = 3n - 4$  equations to fit to the  $4n - 4$  parameters. This leaves  $n$  free parameters, meaning that a cubic Hermite interpolant is not unique and the remaining free parameters can be used to accommodate further or additional constraints that might be around the problem at hand.

### Cubic Spline Interpolation

One way of using the remaining free parameters is by introducing *splines*. A cubic spline is, given the spline definition, a piecewise cubic polynomial, a polynomial of degree  $k = 3$ , and must then be  $k - 1 = 2$  times differentiable. Thinking back on the Hermite cubic, we were left with  $n$  free parameters. By demanding continuity of also the second derivative, we introduce  $n - 2$  new parameters, leaving only 2 final parameters to be free. These 2 remaining parameters can be fixed through a number of different requirements, e.g. by forcing the second derivative at the endpoints to be zero, which leads to the *natural* spline.

A spline is a piecewise polynomial of degree  $k$  that is continuously differentiable  $k - 1$  times.

The Hermite and spline interpolations are useful for different cases. The Hermite cubic might be more appropriate for preserving monotonicity if it is known that the data are monotonic. On the contrary, the cubic spline may enforce a higher degree of smoothness as it takes the second derivative into account as well.

### On Data

For the purpose of this thesis, interpolation of data needs to be fast, efficient and result in a function as smooth as possible. The last criterion is due to the knowledge of the nature of the data. The measurements are not continuous but should indeed in theory be so. Thus a good choice for interpolation of the data examined in this thesis would be the cubic spline interpolation. An instance of a such interpolation can be seen in Figure ??.

Cubic spline interpolation has been used in two instances during this analysis, both times through the Python SciPy package `scipy.interpolate.CubicSpline`.

REFERENCE!!

Firstly, to assure equally spaced data points, so as to be able to perform a useful frequency analysis through spectral transformation, see Section, ??. Secondly cubic spline interpolation was used to improve on peak detection in the final back diffused data. The final data have a rather low resolution, leading to an initial guess of peak positioning that might be shifted due to the

discretization. Through cubic spline interpolation it is possible to construct a smooth estimate of a signal of higher resolution, leading to a peak positioning estimate that might be less shifted, see Figure ??.

### **0.2.2 Detrending and Standardising**

#### **0.2.3 Cycle Length Estimation of Detrended Signal**