0.1 APPENDIX I: Firn Diffusivity

0.2   APPENDIX II: Data - AWI B-cores

0.2.1   AWI B-Cores: Core Specifications

Make specifications for cores.

Site A		
d_L^{CH}	[m]	80.85
d_T^{CH}	[m]	70.9
A_0 W.E	[m]	0.307
A_0 I.E.	[m]	0.281519
T_0	[°C]	-29.41
ρ_0	[kg m^{-3}]	343.0
z_0		0.55
s_L^{CH}	[cm]	30.0
s_T^{CH}	[cm]	65.0
ρ_L	[kg m^{-3}]	836.0
ρ_T	[kg m^{-3}]	812.0
d_L^{TQ}	[m]	80.87
d_T^{TQ}	[m]	70.90
s_L^{TQ}	[cm]	34.0
s_T^{TQ}	[cm]	40.0

Table 0.1: Core specifications for core drilled at Site A. d describes depth of event, A describes accumulation rate, T describes temperature, ρ describes density at given depth and s describes the width of a given event. Subscripts L and T stands for volcanic events Laki and Tambora, respectively, subscript 0 describes initial surface condition, and superscripts CH and TQ represents the original and corrected values for depth and width of events.
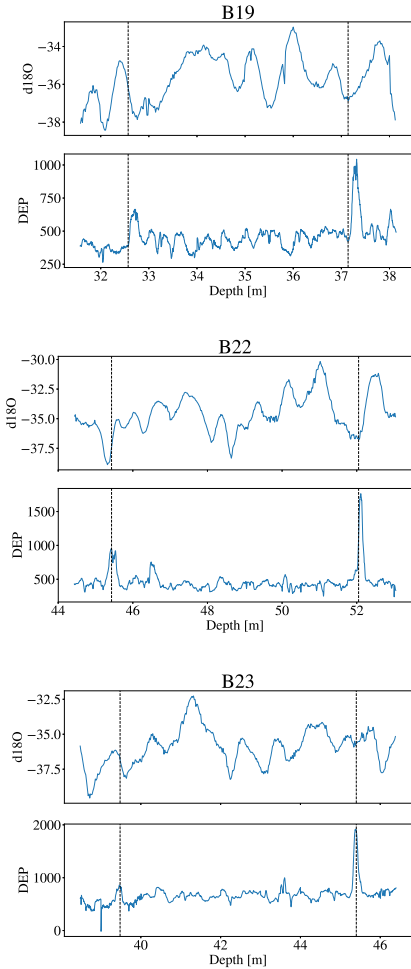
TERRIBLE

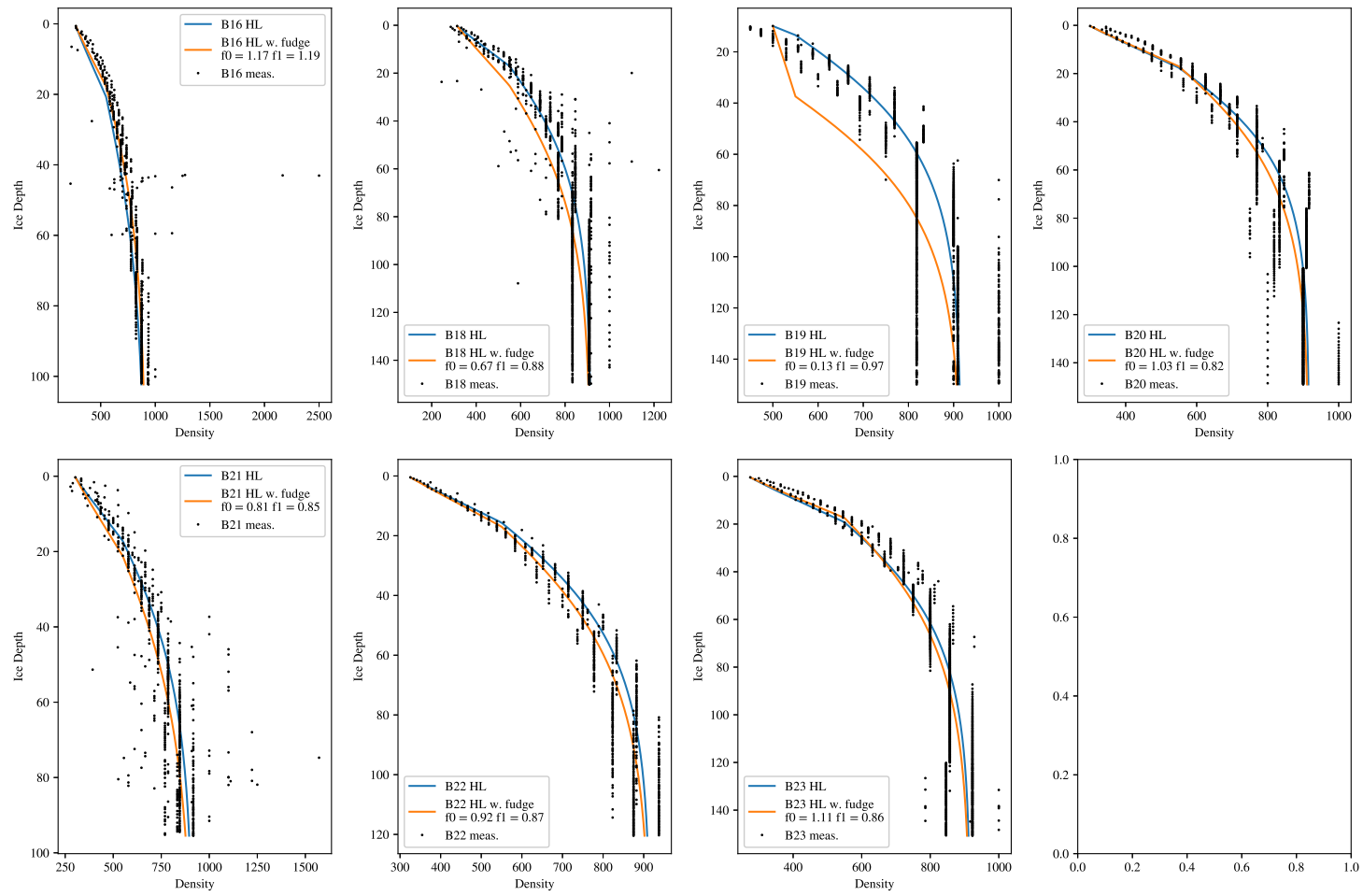## REASONABLE

GOOD

B16

B18

B19

B20

B21

B22

B23
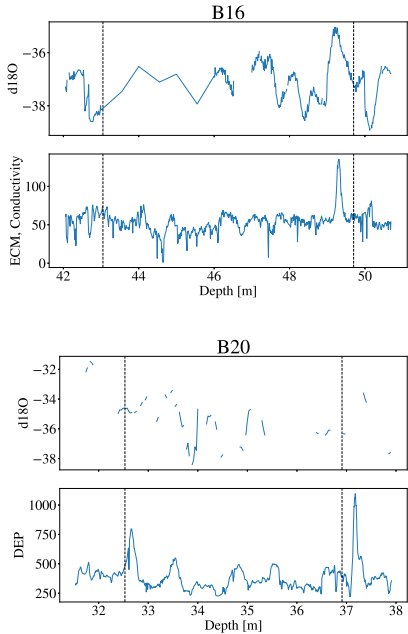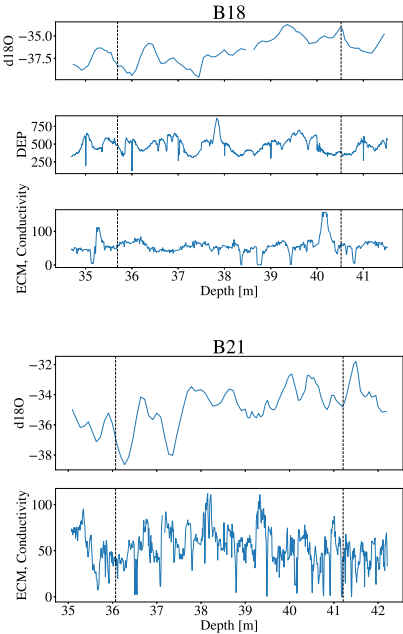
Figure 0.1:
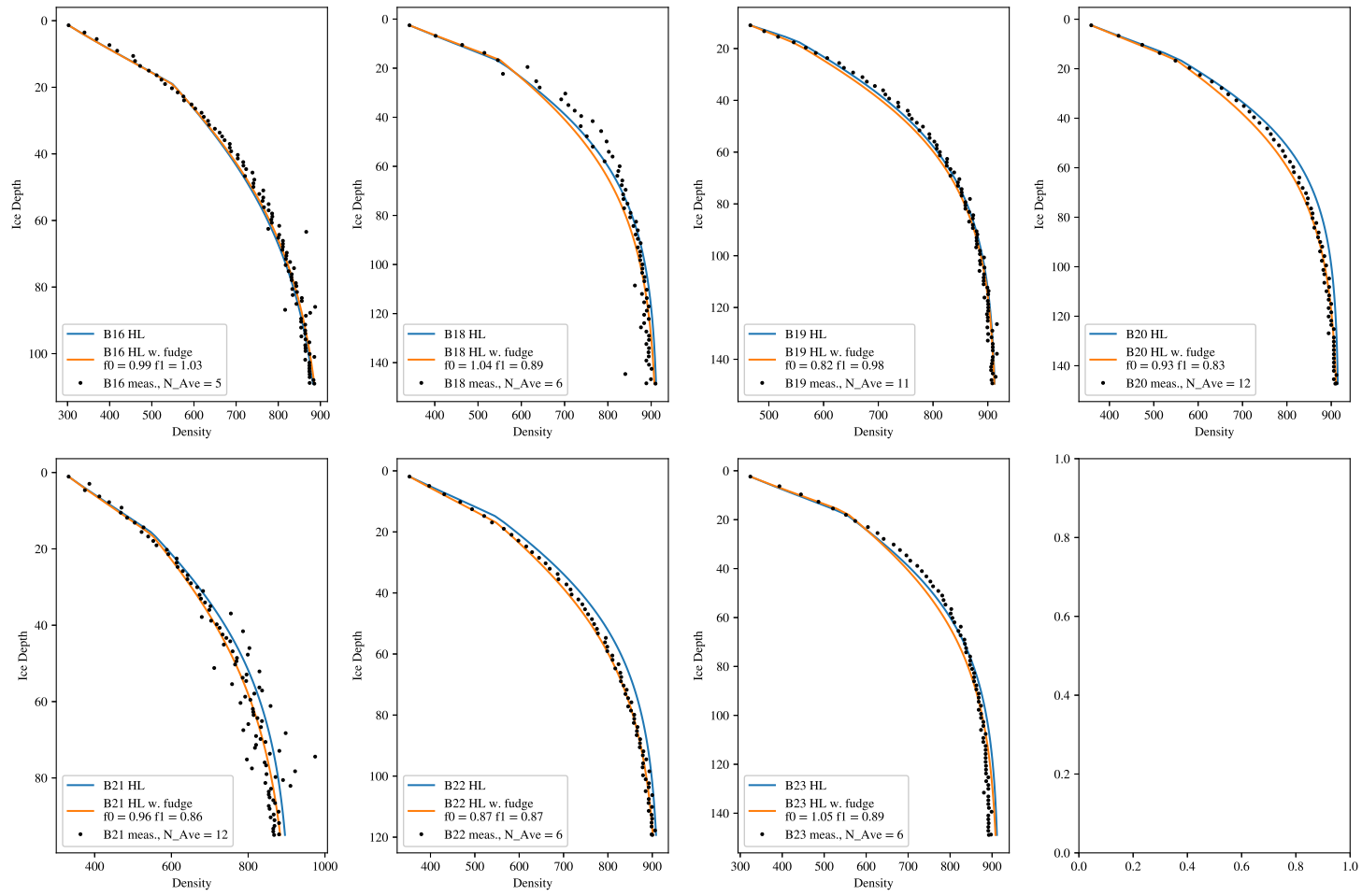AWI B-cores density measurements and HL-modeled densities.

Figure 0.2: AWI B-cores averaged density measurements and related HL-modeled densities.

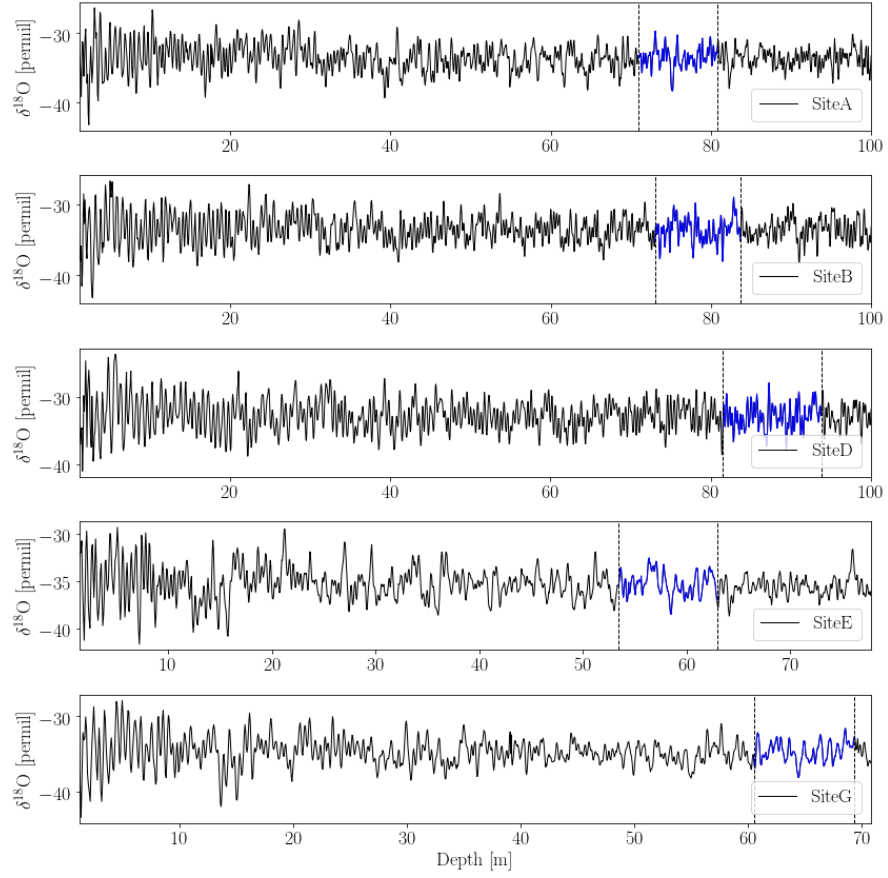## 0.3   APPENDIX III: Data - Alphabet Cores

Figure 0.3: All six Alphabet cores under examination, with Laki to Tambora depth section highlighted.
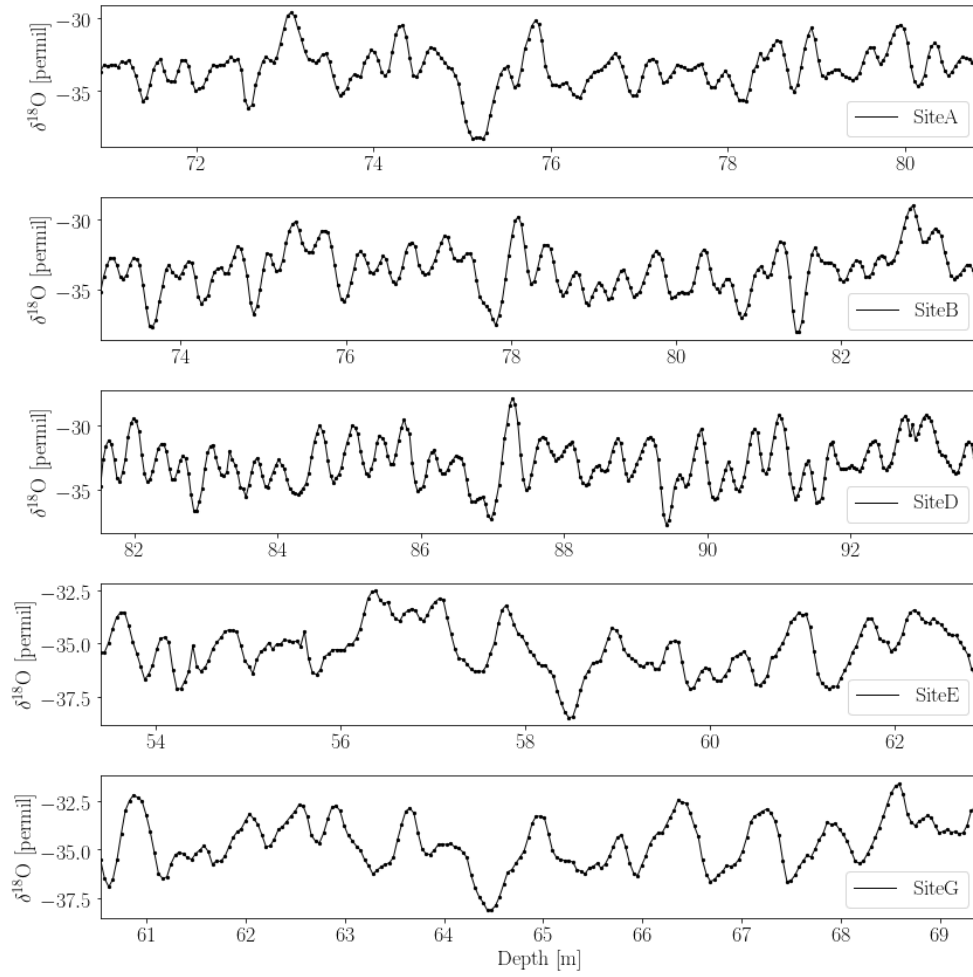
Figure 0.4: All six Alphabet cores, showing only the depth series concerning the Laki to Tambora section.

0.3.1 Alphabet Cores: Core Specifications

		Site A	Site B	Site D	Site E	Site G
d_L^{CH}	[m]	80.85	83.7	93.8	62.95	69.4
d_T^{CH}	[m]	70.9	73.0	81.5	53.4	60.5
A_0 W.E	[m]	0.307	0.327	0.365	0.225	0.251
A_0 I.E.	[m]	0.282	0.300	0.335	0.206	0.230
T_0	[°C]	-29.41	-29.77	-28.3	-30.37	-30.1
ρ_0	[$\mathrm{kg\,m^{-3}}$]	343.0	355.0	350.0	325.0	-
z_0		0.55	0.55	0.825	0.675	-
s_L^{CH}	[cm]	30.0	45.0	55.0	35.0	35.0
s_T^{CH}	[cm]	65.0	55.0	70.0	40.0	75.0
ρ_L	[$\mathrm{kg\,m^{-3}}$]	836.0	841.0	857.0	786.0	807.0
ρ_T	[$\mathrm{kg\,m^{-3}}$]	812.0	816.0	839.0	749.0	778.0
d_L^{TQ}	[m]	80.87	83.82	93.95	62.9	69.38
d_T^{TQ}	[m]	70.90	73.01	81.55	53.43	60.48
s_L^{TQ}	[cm]	34.0	30.0	30.0	45.0	35.0
s_T^{TQ}	[cm]	40.0	57.0	55.0	48.0	82.0

Table 0.2: Core specifications for core drilled at Site A. d describes depth of event, A describes accumulation rate, T describes temperature, ρ describes density at given depth and s describes the width of a given event. Subscripts L and T stands for volcanic events Laki and Tambora, respectively, subscript 0 describes initial surface condition, and superscripts CH and TQ represents the original and corrected values for depth and width of events.

0.4 APPENDIX IV: Spectral Transforms

0.4.0.1 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, Δ, and we can describe our discrete signal X as a function of N discrete time steps $t_k = k \cdot \Delta$, where $k = 0, 1, ..., N - 1$:

$$X_k \equiv X(t_k) \tag{1}$$

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 Δ, there will be a special frequency, the Nyquist critical frequency, defined through the sampling size as:

$$f_{NQ} \equiv \frac{1}{2\Delta}. \tag{2}$$

This frequency is of great importance in transformation of discrete signals. If the continuous signal is sampled at an interval Δ 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, $\tilde{X}(-f_{NQ}), ..., \tilde{X}(f), ..., \tilde{X}(f_{NQ})$. 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}:

$$f_n \equiv \frac{n}{N\Delta}, \qquad n = -\frac{N}{2}, ..., \frac{N}{2} \tag{3}$$

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 **??** needs to be estimated as a sum:

$$\tilde{X}(f_n) = \int_{-\infty}^{\infty} X(\tau)e^{2\pi i f_n \tau}dt \approx \sum_{k=0}^{N-1} X_k e^{2\pi i f_n t_k}\Delta = \Delta \sum_{k=0}^{N-1} X_k e^{2\pi i k \frac{n}{N}} \tag{4}$$

The Discrete Fourier Transform is thus defined as:

$$\tilde{X}_n \equiv \sum_{k=0}^{N-1} X_k e^{2\pi i k \frac{n}{N}} \tag{5}$$

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

$$\tilde{X}(f_n) \approx \Delta \tilde{X}_n \tag{6}$$

The inverse DFT is given as:

$$X_n \equiv \frac{1}{N}\sum_{n=0}^{N-1} X\tilde{X}_n e^{-2\pi i k \frac{n}{N}} \tag{7}$$

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 5 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 implementing a wide range of different theories from complex number arithmetic and prime-factoring to group and number theory ([**?**],[**?**], [**?**] and others) 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 $\mathcal{O}(N \log_2 N)$. In this thesis the FFT used is the one implemented in the `scipy.fft` Python package [**?**], which is based on the works of [**?**]. 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 $\log_2$

0.4.0.2   Nonuniform Discrete Fourier Transform

All FFT algorithms evaluates the DFT definitions from Eqs. 5 to 7 in fast and efficient ways. But one key assumption for these methods is that the data under examination are equispaced, i.e. uniformly distributed, based on the summation in Eq. 5. The computations thus expect uniform data as input and returns uniform data as output. Unfortunately this is not always the case for data collected in physical experiments. In this case the basic assumptions for the calculations of both DFT and FFT are flawed.

The most general form of a nonuniform transform would be the one that takes non-equispaced data as input and also returns non-equispaced transforms as output. Firstly we wish to create a nonuniform discrete Fourier transform (NUDFT) that transforms a sequence of N complex numbers $X_0, ..., X_{N-1}$ to a different sequence of M complex numbers, $\tilde{X}_0, ..., \tilde{X}_{M-1}$. The one-dimensional NUDFT the computes the transformed vector $\tilde{\boldsymbol{X}} = (\tilde{X}_0, ..., \tilde{X}_{M-1})^{\mathrm{T}}$, with entries computed as the sum

$$\tilde{X}_k = \sum_{n=0}^{N-1} X_n e^{-2\pi i p_n f_k}, \qquad 0 \leq k \leq M - 1. \tag{8}$$

The values $X_0, ..., X_{N-1}$ are sample values, $p_0, ..., p_{N-1}$ are sample positions and $f_0, ..., f_{M-1}$ are frequencies. The NUDFT vector $\tilde{\boldsymbol{X}}$ is found by computing M sums with each N terms. This meaning that the computational cost will be of order $\mathcal{O}(M \cdot N)$, and if $M = N$ then of $\mathcal{O}(N^2)$. The NUDFT reduces to the DFT if the points are equispaced, $p_n = \frac{n}{N}$, and the frequencies are integers, $f_k = k$, and can be computed at the cost of the FFT, $\mathcal{O}(N \log_2 N)$. In the literature there are many who have presented different ways to develop a

fast NUDFT ([**?**], [**?**], [**?**], [**?**] among others), generally referred to as NUFFT or NFFT. In this work though, the main focus is on the discrete cosine transform, and the NFFT methods will not be described in depth

0.4.0.3   Discrete Cosine Transform

The Fourier transform in any of its many forms is designed to process complex-valued signals, always producing a complex-valued spectrum, even for signals that were strictly real-valued. The real-valued or complex-valued part of the Fourier spectrum is on their own not enough to represent the full information of the signal, since neither the cosine nor he sine functions (corresponding to the real and the complex parts of the spectrum respectively), constitute a complete set of basis functions. Nonetheless, a purely real-valued signal has a symmetric Fourier spectrum, meaning that it is only necessary to compute half the number of spectral coefficients, without losing any signal information. Since the signals analyzed in this thesis are strictly real, one way to use this knowledge to improve on the works of this project is to consider a different, less expensive, purely real spectral transform. The cosine transform [**?**] seems to do the trick: it uses only cosine functions as basis functions and operates with only real-valued signal and spectral coefficients, and have properties similar to the Fourier transform.

For the discrete version of the cosine transform, DCT, and its inverse, IDCT, a number of different definitions have been proposed, but for this work, the originally formulations by [**?**] are used. These are often referred to as "The DCT" and "The IDCT", and other times as DCT-II and DCT-III. The entries of the computed discrete cosine transform vector,$\tilde{X}_0, ..., \tilde{X}_{M-1}$, of a real-valued signal of N data points, $X_0, ...X_{N-1}$, is computed as

$$\tilde{X}_k = 2 \sum_{n=0}^{N-1} X_n \cos\left(\frac{\pi(2n+1)k}{2N}\right), \qquad 0 \le k < M. \tag{9}$$

To orthonormalize the base functions,$\phi_k(n)$, the coefficients are multiplied by a scaling factor f:

$$f = \begin{cases} \frac{1}{\sqrt{2N}}, & \text{if} k = 0 \\ \frac{1}{\sqrt{4N}}, & \text{otherwise} \end{cases} \tag{10}$$

so that the base functions, $\phi_k[n] = 2f \cos\left(\frac{\pi(2n+1)k}{2N}\right)$, meet the condition:

$$\sum_{n=0}^{N-1} \phi_k[n]\phi_l[n] = \delta_{lk}. \tag{11}$$

The inverse of the DCT, the so called DCT-III, is defined, unnormalized, as:

$$X_k = \tilde{X}_0 + 2\sum_{n=1}^{N-1} \tilde{X}_n \cos\left(\frac{\pi n(2k+1)}{2N}\right), \qquad 0 \le k < N \tag{12}$$

and orthonormalized:

$$X_k = \frac{\tilde{X}_0}{\sqrt{N}} + \sqrt{\frac{2}{N}}\sum_{n=1}^{N-1} \tilde{X}_n \cos\left(\frac{\pi n(2k+1)}{2N}\right), \qquad 0 \le k < N \tag{13}$$

Only when the DCT-III is orthonormalized is it exactly the inverse of the orthonormalized DCT-II. If they are both unnormalized, the DCT-III is the inverse of the DCT-II up to a factor $2N$. As with the DFT, the DCT can directly be computed at a cost of $\mathcal{O}(N \cdot M)$, and can also reduced to $\mathcal{O}(N \log N)$. The fast DCT algorithm(FCT) used here is based on [?] as it is implemented in the `scipy.fft.dct` package[?].

0.4.0.4 Nonuniform Discrete Cosine Transform

Again, as with the FFT, the FCT works under the key assumption that data is equispaced. Though when data is nonuniform, the DCT is described as:

$$\tilde{X}_k = 2\sum_{n=0}^{N-1} X_n \cos\left(2\pi f_k\left(p_n + \frac{1}{2N}\right)\right), \qquad 0 \le k < M - 1 \tag{14}$$

with, in the most general case, nonuniformly spaced signal, $p_o, ..., p_{N-1}$, data and frequency data, $f_0, ..., f_{M-1}$. The inverse of NDCT, the INDCT, is computed as:

$$X_k = \frac{\tilde{X}_0}{\sqrt{N}} + \sqrt{\frac{2}{N}}\sum_{n=1}^{N-1} \tilde{X}_n \cos\left(\left(p_n + \frac{1}{2N}\right)2\pi f_k\right), \qquad 0 \le k < N - 1 \tag{15}$$

It is possible to develop algorithms with the computational cost of $\mathcal{O}(N \log N)$ for NDCT and INDCT as it is for the NDFT ([**?**], [**?**]) but it has showed to be out of the scope of this project and has not been implemented. This of course slows down the final optimization algorithm, as it requires a number of spectral transformations. In Section **??** it is described how the final algorithm has been designed to minimize the use of NDCT, and thus speeding up the final computations.

SIGNAL-MEM: Write this entire section - maybe not necessary? Maybe use in reconstruction of missing data...

0.4.0.5 Maximum Entropy Method (Burg's Method)

0.5 APPENDIX V: Splines and Interpolations

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

$$f(t) = \sum_{j=1}^{n} x_j \phi_j(t) \tag{16}$$

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):

$$f(t_i) = \sum_{j=1}^{n} x_j \phi_j(t_i) = y_i, \quad i = 1, ..., m \tag{17}$$

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

$$\boldsymbol{Ax = y} \tag{18}$$

In this equation $\boldsymbol{A}$ 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 $\boldsymbol{y}$ consists of the known data values y_i, and the n vector $\boldsymbol{x}$ 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.

0.5.0.2 Polynomial Interpolation

The most common way to determine an interpolant is through polynomials. Denoting a set of all polynomials of degree at most k, $\quad k \geq 0$ 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[1],

$$\phi_j(t) = t^{j-1}, \quad j = 1, ..., n. \tag{19}$$

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

$$p_{n-1}(t) = x_1 + x_2 t + \cdots + x_n t^{n-1}. \tag{20}$$

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

$$\boldsymbol{Ax} = \begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-1} \\ 1 & t_1 & \cdots & t_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_1 & \cdots & t_1^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \boldsymbol{y}. \tag{21}$$

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

COMP-INTERP: REFERENCE!!!

When using the monomial basis and using a standard linear equation solver to determining the interpolants coefficients requires $\mathcal{O}(n^3)$ work and often results in ill-conditioned Vandermonde matrices $\boldsymbol{A}$, 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

---

[1]Roughly speaking, a polynomial with only one term.

makes the columns of $\boldsymbol{A}$ 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.
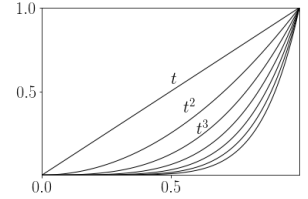

Figure 0.5: Illustration of the first eight monomials.

0.5.0.3 Piecewise Polynomial Interpolation and Splines

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).

COMP-INTERP: REFERENCES!!

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), $i = 1, ..., n$, $t_1 < t_2 < \cdots < t_n$, 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 explotiting 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 pint. 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

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

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.

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.

For the case of this study, cubic spline interpolation is used to either evenly redistribute slightly unevenly sampled data or to enhance resolution for more precise peak detection. The general method for cubic spline interpolation used here is described in the following.

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

$$\hat{d}_0 = \Delta \lceil \frac{d_0}{\Delta}, \rceil \tag{22}$$

$$\hat{d}_{\hat{n}-1} = \Delta \lfloor \frac{d_{n-1}}{\Delta} \rfloor. \tag{23}$$

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

$$\hat{n} = 1 + \frac{\hat{d}_{\hat{n}-1} - \hat{d}_0}{\Delta}, \quad \hat{n} \in \mathbb{Z}. \tag{24}$$

Thus our new depth array will be given as

$$\hat{\mathbf{d}} = \hat{\mathbf{d}}_0 + j \cdot \Delta, \quad j = 0, ..., \hat{n} - 1. \tag{25}$$

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.

0.6 APPENDIX VII: Diffusion Illustrated Through Synthetic Data

SIGNAL-
SYNTHDATA:
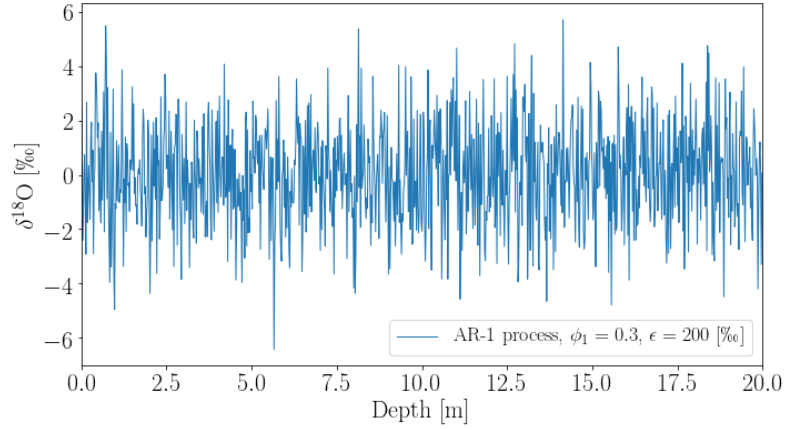Write about syn-
thetic data genera-
tion.

Figure 0.6: Illustration of modelled auto-regressive process of first order (AR-1).
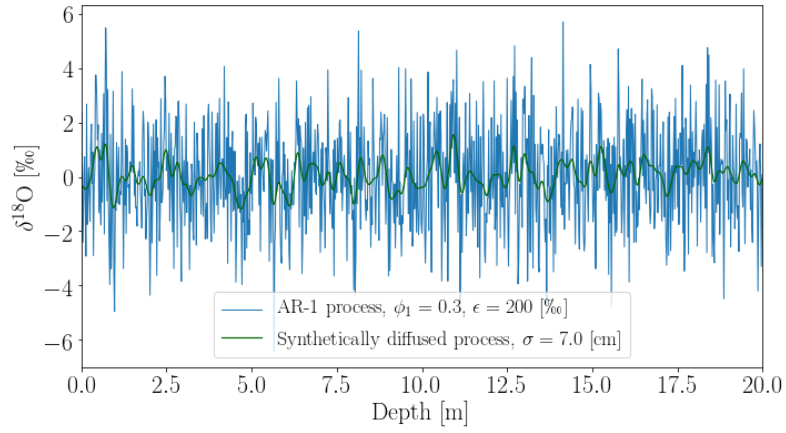

Figure 0.7: Illustration of AR-1 process and the same process diffused as in firn.
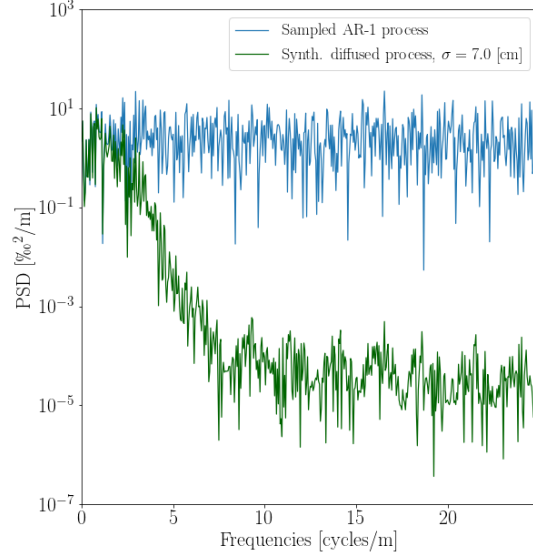
Figure 0.8: AR-1 process, raw and diffused, illustrated in the spectral domain.

0.7 APPENDIX IIX: Herron Langway Empirical Model

Sorge's law(REFERENCES) assumes that the relation between snow density ρ and depth h is invariant with time, given a constant snow accumulation and temperature. Furthermore, annual layer thinning by plastic flow is ignored.

Densification of firn, which can be described as the proportional change in air space, is linearly related to change in stress due to the weight of the overlying snow(REFERENCES):

$$\frac{d\rho}{\rho_i - \rho} = \text{const.} \, \rho \, dh \tag{26}$$

By integration, this implies a linear relation between $\ln\left[\frac{\rho}{\rho_i - \rho}\right]$ and h.

When considering real data, analysis shows that $\ln\left[\frac{\rho}{\rho_i - \rho}\right]$ vs h. plots have two linear segments(EXAMPLE), corresponding to the first and second stages of densification, with separation of segments at $\rho = 0.55$ and $\rho = 0.8$. These segments on the plots will yield two different slopes with slope constants:

$$C = \frac{d \ln\left[\frac{\rho}{\rho_i - \rho}\right]}{dh}, \rho < 0.55 \qquad (27a)$$

$$C' = \frac{d \ln\left[\frac{\rho}{\rho_i - \rho}\right]}{dh}, 0.55 < \rho < 0.8 \qquad (27b)$$

To find the densification rate, $\frac{d\rho}{dt}$, substitute $\frac{dh}{dt} = \frac{A}{\rho} \rightarrow dt = \frac{\rho}{A} dh$ and use the differentiation $\frac{\partial}{\partial t}\left[\ln\left[\frac{x(t)}{k-x(t)}\right]\right] = \frac{k\frac{dx}{dt}}{(k-x(t))x(t)}$

$$C = \frac{\rho}{A} \frac{d \ln\left[\frac{\rho}{\rho_i - \rho}\right]}{dt}$$

$$= \frac{\rho}{A} \frac{\rho_i}{\rho(\rho_i - \rho)} \frac{d\rho}{dt}$$

$$= \frac{1}{A} \frac{\rho_i}{\rho_i - \rho} \frac{d\rho}{dt}$$

leading to

$$\frac{d\rho}{dt} = \frac{CA}{\rho_i}(\rho_i - \rho) \qquad (28a)$$

$$\frac{d\rho}{dt} = \frac{C'A}{\rho_i}(\rho_i - \rho) \qquad (28b)$$

To continue from here two assumptions are made. The first is that the temperature and the accumulation rate dependencies may be separated, and that they thereby have no inter-correlation. The second is that the rate equations may be written as:

$$\frac{d\rho}{dt} = k_0 A^a (\rho_i - \rho), \rho < 0.55 \qquad (29a)$$

$$\frac{d\rho}{dt} = k_1 A^b (\rho_i - \rho), 0.55 < \rho < 0.8 \qquad (29b)$$

where k_0 and k_1 are Arrhenius type rate constants which are only temperature dependent, and a and b are constants determining the significance of the accumulation rate and are dependent on the densification mechanisms.
a and b may be determined by comparing slopes for densification at different sites of nearly equivalent conditions as:

$$a = \frac{\ln\left(\frac{C_1}{C_2}\right)}{\ln\left(\frac{A_1}{A_2}\right)} + 1 \qquad (30)$$

and equivalently for b, with C_1' and C_2'.

k_0 and k_1 can be estimated by observing values of k at different temperatures and plotting $\ln(k)$ versus temperature - a so-called Arrhenius plot(REFERENCES) - to find A and E_a in equations:

$$k = Ae^{-\frac{E_a}{k_B T}} = Ae^{-\frac{E_a}{RT}} \tag{31}$$

$$\ln(k) = \ln(A) - \frac{E_a}{R}\frac{1}{T}$$

leading to values of k_0 and k_1 of:

$$k_0 = 11e^{-\frac{10160}{RT}} \qquad \text{(32a)}$$

$$k_1 = 575e^{-\frac{21400}{RT}} \qquad \text{(32b)}$$

Depth-density and depth-age calculations

Assuming that temperature, annual accumulation rate and initial snow density are known, the following calculations can be made:

- Density at depth h, $\rho(h)$

- Depth at pore close-off, $\rho = 0.55$

- Depth-age relationship from surface to pore close-off (stage 1 and 2).

1. stage of densification: Depth-density profile:

$$\rho(h) = \frac{\rho_i Z_0}{1 + Z_0} \tag{33}$$

where $Z_0 = e^{\rho_i k_0 h + \ln\left[\frac{\rho_0}{\rho_i - \rho_0}\right]}$. In this segment, the depth-density is independent of accumulation rate. The critical density depth can be calculated as:

$$h_{0.55} = \frac{1}{\rho_i k_0}\left[\ln\left[\frac{0.55}{\rho_i - 0.55}\right] - \ln\left[\frac{\rho_0}{\rho_i - \rho_0}\right]\right] \tag{34}$$

and the age at close-off depth as:

$$t_{0.55} = \frac{1}{k_0 A}\ln\left[\frac{\rho_i - \rho_0}{\rho_i - 0.55}\right] \tag{35}$$

24

ICE-DENS: Figure out where this comes from.

2. stage of densificaion: The depth-density profile

$$\rho(h) = \frac{\rho_i Z_1}{1 + Z_1} \tag{36}$$

where $Z_1 = e^{\rho_i k_1 (h - h_{0.55}) \frac{1}{A^{0.5}} + \ln\left[\frac{0.55}{\rho_i - 0.55}\right]}$. The age of firn at a given density ρ:

$$t_\rho = \frac{1}{k_1 A^{0.5}} \ln\left[\frac{\rho_1 - 0.55}{\rho_1 - \rho}\right] \tag{37}$$

An estimate of the mean annual accumulation rate can be made from the slope C' and the mean annual temperature:

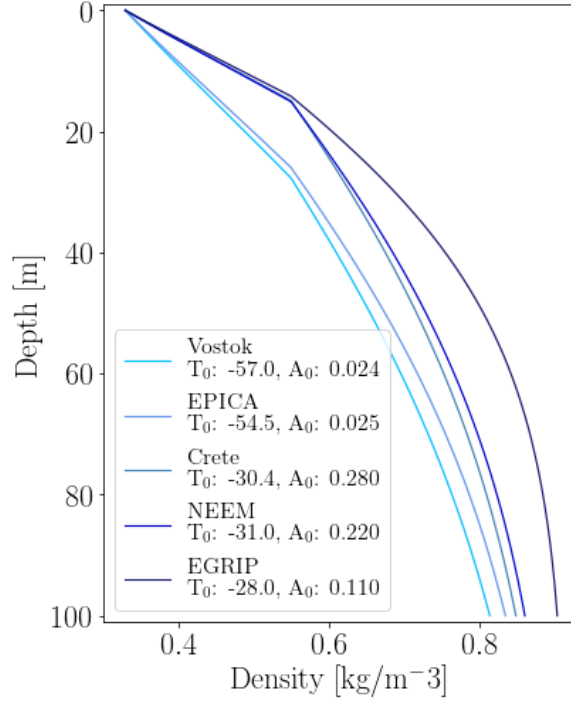$$A = \left(\frac{\rho_i k_1}{C'}\right)^2 \tag{38}$$

Figure 0.9: Density profile examples given five different initial conditions representing present day conditions at the five different ice core locations. Temperature, T_0, is in °C and accumulation, A_0, is in meter of water equivalent per year.

0.8 APPENDIX IX: Iso-Community Firn Model

0.8.0.1 Firn Diffusivity

The Iso-CFM framework contains a number of different new modules added to the CFM. A specific module for calculation of the firn diffusivity, $D(\rho(z))$, is provided, containing several different methods for the calculations of the individual parameters contained in the calculation of the diffusivity constant, using the formulation in [?, Johnsen et al., 2000]:

$$D_i(z) = \frac{m\,p\,D_{\text{air}}}{R\,T(z)\,\alpha_{s/v}^i\,\tau(z)}\left(\frac{1}{\rho(z)} - \frac{1}{\rho_{\text{ice}}}\right) \tag{39}$$

with $i \in \text{O}^{18}, \text{O}^{17}, \text{D}$ representing the three different types of water isotopic ratios generally examined. The different terms in 39 each describes the following:

m: molecular weight in [kg]

$R = 8.314478 \left[\frac{\text{m}^3 \text{Pa}}{\text{K mol}} \right]$: molar gas constant

T Temperature [K]

p: saturation vapor pressure over ice in [Pa]

The saturation vapor pressure over ice can be calculated in two different ways, as in [?, Murphy & Koop 2005]:

$$p = \exp\left(28.9074 - \frac{6143.7}{T} \right) \tag{40}$$

$$p = \exp\left(9.5504 - \frac{5723.265}{T} + 3.530 \ln(T) - 0.0073\,T \right) \tag{41}$$

where Eq. 41 takes the temperature dependence of the latent heat of sublimation of ice into account when integrating the Clausius-Clapeyron equation. A third expression is presented in [?, Johnsen et al., 2000] as:

$$p = 3.454 \cdot 10^{12} \exp\left(\frac{-6133}{T} \right) \tag{42}$$

which will be the one used for analytical calculations of diffusion length in this project.

D_{air}: diffusivity of water vapor in air,

calculated from $P_0 = 1\text{Atm}$, $T_0 = 273.15\text{K}$, T temperature in [K] and P ambient pressure in [Atm], as in [?, Hall and Prupacher, 1976]:

$$D_{\text{air}} = 2.1 \cdot 10^{-5} \left(\frac{T}{T_0} \right)^{1.94} \left(\frac{P_0}{P} \right) \tag{43}$$

From [?, Merlivat, 1978] the additional diffusivity of water isotopes ratios for ^{18}O and ^{2}H vapor were defined as

$$D_{\text{air}^2\text{H}} = \frac{D_{\text{air}}}{1.0251} \tag{44}$$

$$D_{\text{air}^{18}\text{O}} = \frac{D_{\text{air}}}{1.0285} \tag{45}$$

$\alpha^i_{s/v}$: solid-to-vapour fractionation factor. $i = {}^{18}\text{O}$, D, ${}^{17}\text{O}$,

For both $\alpha^{18}_{s/v}$ and $\alpha^2_{s/v}$ there are multiple options for parameterisation of the fractionation factor. Considering $\alpha^{18}_{s/v}$, one can choose between [**?**, Majoube 1971] and [**?**, Ellehøj et al., 2013], respectively as:

$$\ln(\alpha^{18}_{s/v}) = \frac{11.839}{T} - 28.224 \cdot 10^{-3} \tag{46}$$

and

$$\ln(\alpha^{18}_{s/v}) = 0.0831 - \frac{49.192}{T} + \frac{8312.5}{T^2} \tag{47}$$

For $\alpha^2_{s/v}$ the parameterisation from [**?**, Merlivat and Nief, 1967], [**?**, Ellehøj et al., 2013] or [**?**, Lamb et al., 2017], respectively as:

$$\ln(\alpha^2_{s/v}) = \frac{16288}{T^2} - 9.45 \cdot 10^{-2}, \tag{48}$$

$$\ln(\alpha^2_{s/v}) = 0.2133 - \frac{203.1}{T} + \frac{48888}{T^2} \tag{49}$$

or

$$\ln(\alpha^2_{s/v}) = \frac{13525}{T^2} - 5.59 \cdot 10^{-2}. \tag{50}$$

The parameterisation of the fractionation factor related to the ${}^{17}\text{O}$ water isotopic ratios is based on [**?**, Barkan and Luz, 2005] as $\alpha^{17}_{s/v} = 0.529\alpha^{18}_{s/v}$. For a comparison of the different parameterisations see [**?**, Gkinis et al., 2021]. The default choices in the iso-CFM modules is [**?**, Majoube 1971] for $\alpha^{18}_{s/v}$ and [**?**, Merlivat and Nief, 1967] for $\alpha^2_{s/v}$

τ: firn tortuosity

In [**?**, Johnsen et al., 2000] a parameterisation of the firn tortuosity was presented as:

$$\frac{1}{\tau} = \begin{cases} 1 - b_\tau \left(\frac{\rho}{\rho_{\text{ice}}}\right)^2, & \rho \leq \frac{\rho_{\text{ice}}}{\sqrt{b_\tau}} \\ 0, & \rho > \frac{\rho_{\text{ice}}}{\sqrt{b_\tau}} \end{cases} \tag{51}$$

where $b_\tau = 1.3$ and $\rho_{\text{ice}} = 917\frac{\text{kg}}{\text{m}^3}$, implying for $\frac{1}{\tau} = 0$ a close-off density of $\rho_{\text{co}} = 804.3\frac{\text{kg}}{\text{m}^3}$. This close-off density refers to the density at the depth where diffusive fluxes stop and $\frac{D_{\text{air}}}{D_{\text{eff}}} \to \infty$. Different parameterisations have

been suggested, some defined from the total porosity, but for this project, the expression used is the one given in Eq. 51.

0.8.0.2 Diffusion Length Profiles

The iso-CFM computes a numerical solution for σ using a time-stepping scheme, as is the case for the original CFM, to estimate the most likely diffusion length profile at a given site. From each time step the CFM computes $\frac{d\rho}{dt}$ and T, and the iso-CFM uses these results to calculate the quantity $\frac{d\sigma^2}{dt}$:

$$\frac{d\sigma^2}{dt} = 2\left(D(t) - \frac{\sigma^2}{\rho}\frac{d\rho}{dt}\right) \tag{52}$$

Eq. 52 shows that the diffusion length signal throughout the ice is a result of two processes, opposing each other: the always positive diffusivity term $D(t)$, and the densification process contributing negatively to the change over time, $-\frac{\sigma^2}{\rho}\frac{d\rho}{dt}$. After a certain depth, the densification term comes to dominate and thus the entire equation becomes negative and the value of the diffusion length is decreasing, see Figure 0.10.

To simplify the work of this thesis, the numerical module of the CFM and the iso-CFM has not been implemented in the final computations, and the diffusion length profiles referred to in the rest of the project are calculated through an analytical method, using equations derived from Eq. 52 analytically. A short walk-through of the derivations will be presented here as they are described in [?, Gkinis et al., 2021]. By substitution of variables rearrangement Eq. 52 becomes:

$$\frac{d\sigma^2}{d\rho} + \frac{2\sigma^2}{\rho} = 2\left(\frac{d\rho}{dt}\right)^{-1}D(\rho) \tag{53}$$

which can be converted to integral form:

$$\sigma^2(\rho) = \frac{1}{\rho^2}\int_{\rho_0}^{\rho} 2\rho'^2 \left(\frac{d\rho'}{dt}\right)^{-1}D(\rho')d\rho' \tag{54}$$

Then, by using the densification rate parameterisation given in [?, Herron and Langway, 1980], the expression becomes:

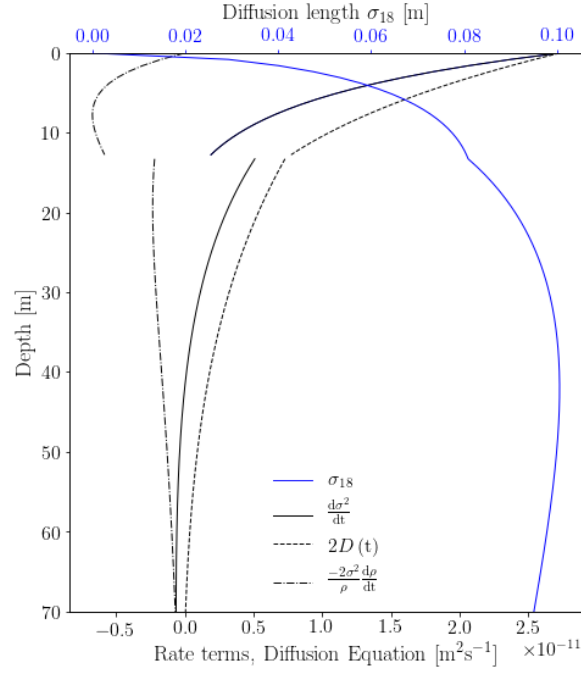$$\frac{d\rho(z)}{dt} = k(T)\,A^{\nu}\,(\rho_{\text{ice}} - \rho(z)), \tag{55}$$

Figure 0.10: Contribution of the diffusion(dashed) and densification(dot-dashed) terms from Eq. 52 to the final analytical diffusion length solution (blue).

where $k(T)$ is an Arrhenius-type densification rate constant, dependent on temperature and densification zone described by:

$$k_0(T) = 0.011 \, \exp\left(-\frac{10160}{RT}\right), \qquad \nu_0 = 1 \tag{56}$$

in the upper densification zone, $\rho < 550\rho_{\text{co}}$. In the lower densification zone, $\rho \geq \rho_{\text{co}}$, it is described as:

$$k_1(T) = 0.575 \, \exp\left(-\frac{21400}{RT}\right), \qquad \nu_1 = 0.5. \tag{57}$$

Using the parameterization of the diffusivity coefficient from Eq. 39 and expressing the term $1/\tau = 1 - b_{tau}\left(\frac{\rho}{\rho_{\text{ice}}}\right)^2$ in densities, the diffusivity coefficient can be described as a function of density:

$$D_i(\rho) = \frac{m\,p\,D_{\text{air}}}{R\,T\,\alpha^i_{s/v}}\left(1 - b_\tau\left(\frac{\rho}{\rho_{\text{ice}}}\right)^2\right)\left(\frac{1}{\rho} - \frac{1}{\rho_{\text{ice}}}\right). \tag{58}$$
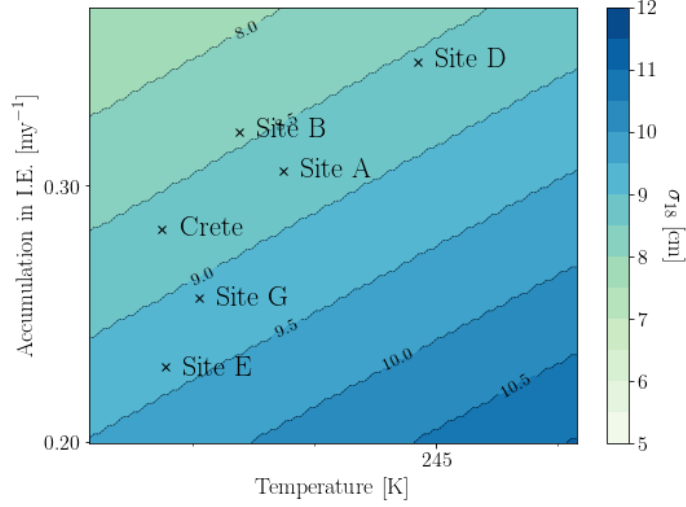
Figure 0.11: Crete and surrounding Alphabet cores, as their analytical solutions place them according to observed temperature and accumulation rate.

By then inserting in Eq. 54, defining $\frac{m\,p\,D_{\text{air}}}{R\,T\,\alpha^i_{s/v}} = \zeta$, and integrating the final analytical equations for the diffusion length in upper and lower densification zones can be obtained:

$$\sigma^2(\rho < \rho_{\text{co}}) = \frac{\zeta}{\rho^2\,k_0\,A^{\nu_0}\,\rho_{\text{ice}}}\left[\rho^2 - \rho_0 - \frac{b_\tau}{2\rho_{\text{ice}}^2}(\rho^4 - \rho_0^4)\right] \tag{59}$$

$$\begin{aligned}\sigma^2(\rho \geq \rho_{\text{co}}) =&\frac{\zeta}{\rho^2\,k_1\,A^{\nu_1}\,\rho_{\text{ice}}}\left[\rho^2 - \rho_{\text{Cr}} - \frac{b_\tau}{2\rho_{\text{ice}}^2}(\rho^4 - \rho_{\text{Cr}}^4)\right]\\&+ \frac{\zeta}{\rho^2\,k_0\,A^{\nu_0}\,\rho_{\text{ice}}}\left[\rho_{\text{Cr}}^2 - \rho_0 - \frac{b_\tau}{2\rho_{\text{ice}}^2}(\rho_{\text{Cr}}^4 - \rho_0^4)\right]\end{aligned} \tag{60}$$

The analytical equations have been used for creating a contour plot of the analytical solutions for σ_{18} at the close-off density, ρ_{co}.

These analytical equations are used to compute diffusion lengths to compare with the optimal diffusion length estimates computed from the raw data. One could advantageously spend some time and energy on using the iso-CFM to numerically compute the comparison diffusion lengths with different
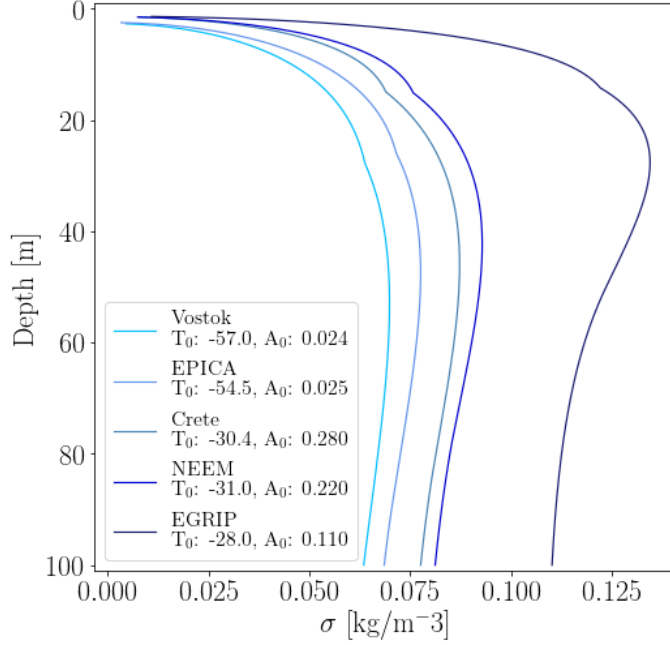
Figure 0.12: Analytically calculated diffusion length profile examples given five different initial conditions representing present day conditions at the five different ice core locations. Temperature, T_0, is in °C and accumulation, A_0, is in meter of water equivalent per year.

temperature and accumulation forcing to recreate a diffusion length profile corresponding to the largest likelihood at a given drill site. Since the iso-CFM do consist of many different modules all with different possibilities for parameterisation, it is outside the scope of this project to develop a iso-CFM diffusion length estimate. In-depth methodology and results from the iso-CFM can be found in [?, Gkinis et al., 2021].

0.9   APPENDIX X: Dielectric Profiling

A method was later developed to demonstrate how both acids and salts play a decisive role in the determination of the electrical behavior of ice. The dielectric response of an ice core can be used to determine the total ionic concentration of the core. For ECM the measurements are sensitive to the fluctuating distance between ice core and electrodes, and after each measurement a fresh piece of ice needs to be prepared to repeat a measurement.

A new dielectric profiling technique (DEP) was developed (REFERENCES) with the advantages over the ECM that no direct contact is needed between the electrodes and the ice, so that the ice can stay in a protective polythene sleeve and the experiment easily can be repeated on the same piece of ice. Together the ice core and the polythene sleeve creates a complete system, where the plastic acts as an electrical blocking layer.
The dielectric response is measured by a sweeping of the AF-LF frequency range for the entire ice-polythene system. At LF the conductivity of the composite system is within a few percentages of the intrinsic behavior of the ice itself. At HF-VHF frequencies it also approximates well enough (REFERENCES).

The measured dielectric parameters are the conductivity of ice at HF-VHF range, denoted σ_∞ where ∞ signifies a frequency much higher than the relaxation frequency, f_r, of the dominant dispersion in the system. Both of these parameters display clear chemical response signals which can be used either alone or in combination with other ice core analysis measurements like ECM and isotope analysis.
If the core under analysis is chemically analyzed for Na^+, Mg^{2+}, Cl^-, SO_4^{2-} and NO_3^-, a number of important parameters, which can be used to evaluate the response of the dielectric parameters, can be calculated(REFERENCES):

- The salt parameter, which represents the total marine cation concentration calculated with the assumed marine ratios as:

$$[\text{salt}] = 1.05([Na^+] + [Mg^{2+}]) \tag{61}$$

- XSO_4, the excess sulphate, which represents the amount the sulphate concentration is above the expected if the salt and sulphate ions were in normal sea salt ratios. Excess sulphate is essentially sulphuric acid, which is the main acidic component of the ice.

- The strong acid content of the ice has been calculated as(assuming no other ions present in significant quantities):

$$[\text{acid}] = [\text{Cl}^-] + [\text{SO}_4^{2-}] + [\text{NO}_3^-] - 1.05([\text{Na}^+] + [\text{Mg}^{2+}]) \qquad (62)$$

From data, it can be seen that acid and salt concentration peaks clearly affect σ_∞ and f_r(EXAMPLES, REFERENCES). The relationship between salt and acid, and the two dielectric parameters have been derived through non-linear regression analysis. In PAPER(REFERENCES) the linear responses for the DEP at -22°C were:

$$\sigma_\infty = (0.39 \pm 0.01)[\text{salt}] + (1.43 \pm 0.05)[\text{acid}] + (12.7 \pm 0.3) \qquad (63)$$

with 76.6 % variance

$$f_r = (440 \pm 11)[\text{salt}] + (612 \pm 65)[\text{acid}] + (8200 \pm 400) \qquad (64)$$

with 68.4 % variance. σ_∞ is measured in μS/m, f_r in Hz and [acid] and [salt] in μEq/l. The total ionic concentration of the ice core is strongly linked to the dielectric parameters, and a regression between the total anion concentration and the dielectric parameters gives:

$$[\text{anions}] = [\text{salt}] + [\text{acid}] = 0.022\sigma_\infty^{1.89} + 10^{-6}f_r^{1.61} - 0.2 \qquad (65)$$

with 86.7 % variance.

The DEP complements the ECM technique by not only reacting to acids alone, as ECM does, but responds to both neutral salts and acids. The acid term is here associated with the DC conductivity, the same way it is also detected by ECM. The dielectric dependence on salts is consistent with the Bjerrum L defect[2]affecting every one or two salt ions in the ice, indicating that a large fraction of the neutral salt is incorporated into the ice lattice. [3]

The sensitivity to salt concentrations allows for identifications of periods with major storms and open seas which are also important identifiers for paleo climate research, along with the volcanic eruption detection made possible through the ECM.

---

[3]A Bjerrum defect is a crystallographic defect specific to ice, partly responsible for the electrical properties of ice. Usually a hydrogen bond will normally have one proton, but with a Bjerrum defect it will have either two protons (D defect) or no proton (L defect).(REFERENCES)