0.1 σ Estimation Method

The general idea of the optimal σ estimation method is to back diffuse a depth series defined on an interval where the time span (i.e. the number of peaks and troughs expected in that section) is known. This allows us to use the diffusion length as a tuning parameter, to find the diffusion length estimate which generates the right number of peaks and troughs and fulfills the imposed constraints in the back diffused depth series. If more than one diffusion length meet these constraints, the largest diffusion length to still fulfill the constraints is sought after. This diffusion length is then assumed to be the optimal guess on the diffusion length in that interval, which allows for a temperature estimate, following the temperature dependence of the diffusion length, as described in Section **??**.

The algorithm consists of two modules, where one module describes the numerical back diffusion, given an inputted depth series, core specification and specific σ_0 estimate (which is either manually inputted or estimated from the spectral analysis). The flowchart describing the processes carried out in this module can be seen in Figure 0.1. Many of the sections in this module are only necessary in the initialization of the algorithm as these parts do not change if the inputted diffusion length estimate is changed. In Figure 0.1 anything carried out above the *Frequency Filters* block to the left is only computed once, and the same with anything to the right of it, except the σ_0 estimate. The density and diffusion profile calculations, the spectral analysis and the Wiener filter construction is inherent to the depth series alone, and these analyses are carried out as previously described in this thesis.

The second module is responsible for the optimization. This module examines the parameter space containing the diffusion length estimates, and utilizes a constrained direct search method to find the optimal diffusion length estimate. This method is illustrated through a flowchart in Figure 0.2.

0.1.1 Module 1: Initialization and Back Diffusion

The first module of the algorithm, containing initialization and describing the general back diffusion method, is based on many of the aspects and models presented in the previous Chapters **??** and **??**. Therefore the description of this module will focus on the work flow and not so much on the specific details of each process, as these have already been presented.

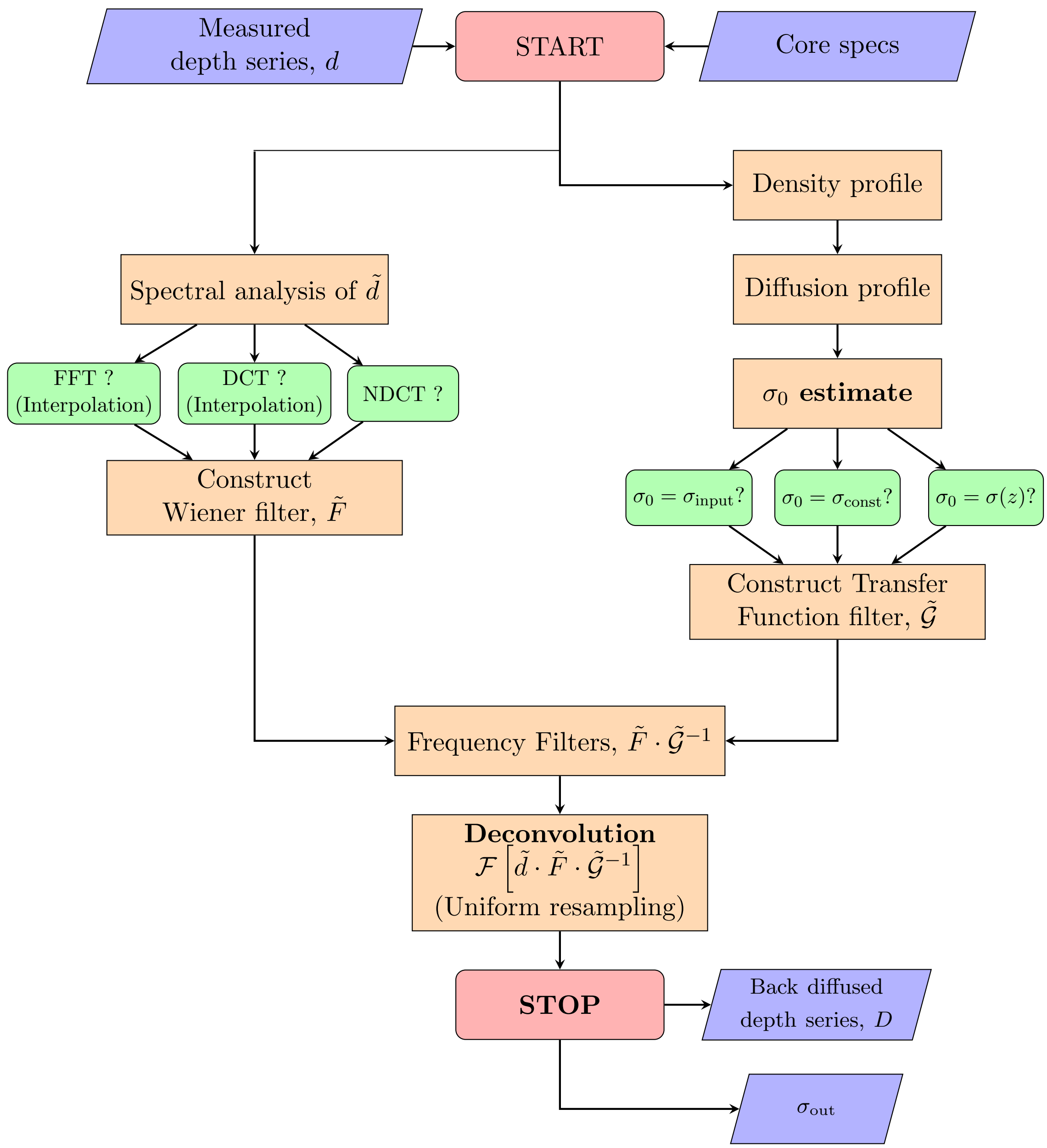

Figure 0.1: Flowchart of initialization method for back diffusion of a depth series given a diffusion length estimate.

The module takes an input of a measured depth series in a given interval, d, and the specifications concerning the drill site and the ice core in general. From here the work flow splits in two, one route(left) analyzing the depth series, and one(right) giving an estimate of σ at that depth, based on models.

The right flow describes how the σ_0 estimate is given on the basis of the core specifications passed into the algorithm. First, a HL-density profile is modelled, based on the necessary input parameters of annual accumulation rate A_0 and drill site surface temperature T_0, and the optional inputs of surface density ρ_0 and measured depth versus density data. Secondly, this density profile is used to compute a diffusion length profile, by the use of the Iso-CFM.

The modelled diffusion length profile is then used to find a theoretical σ_0 estimate. This then used as the standard deviation in the Gaussian transfer function filter, $\mathcal{G}$ used for deconvolution, unless a σ_0 is inputted manually. From the modelled diffusion length profile, it is also possible to choose not just a constant σ_0, but an option for a depth-varying diffusion length, $\sigma(z)$, used in the transfer function is also available.

The left flow shows the analysis carried out on the measured depth series, d. First, the depth series is transformed to the frequency domain, with the users choice of spectral transformation method. If FFT or DCT is chosen, the depth series will be interpolated to resemble equispaced data. Then the frequency series is analyzed according to Section **??**, where fits to the signal and noise are made, that again are used to construct an optimal Wiener filter, $\tilde{F}$. During this process, it is optional to choose the standard deviation of the estimated noise free signal, σ_{tot}, as the diffusion length to use in the deconvolution.

Finally, the right and the left flow are combined to construct the final restoration filter, $\tilde{R} = \tilde{F}\tilde{\mathcal{G}}^{-1}$, which is used for optimal enhancement and restoration when deconvoluting the signal to $D = \mathcal{F}\left[\tilde{d} \cdot \tilde{F} \cdot \tilde{\mathcal{G}}^{-1}\right]$, and the module returns the back diffused series D and the diffusion length used, σ_{out}.

0.1.2 Module 2: Estimating the Optimal σ

The second module contains the optimization of σ. It is based on a constrained direct search method which examines the one dimensional σ space. A flow chart of the module can be seen in Figure 0.2. The module takes three

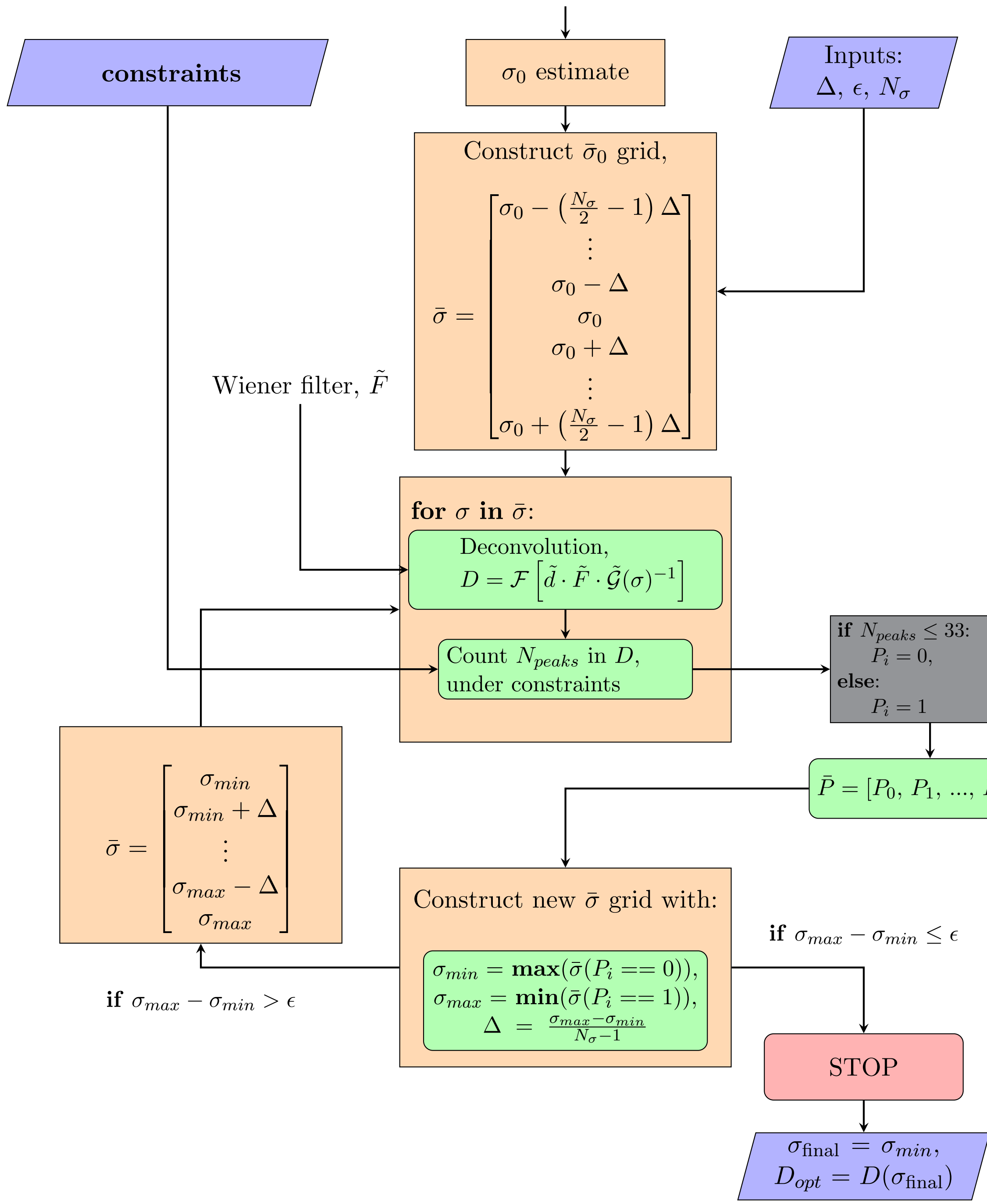

Figure 0.2: For Method (optimization)

initialization parameters as input, the starting grid size Δ, a small size ϵ and the number of grid points to examine N_σ, as well as the constraints imposed on the depth interval, described in the following section, 0.1.3.

The method is initialized by an input σ_0 estimate, the right flow in module 1, from either models, spectral analysis or manual input, which is used to create the first coarse $\bar{\sigma}$ grid. The method also carries out the left flow from module 1 as part of the initialization. These steps are not needed repeated again.

Then the module carries out the deconvolution method on the $\bar{\sigma}$ grid and uses the imposed constraints to count number of peaks in the back diffused depth series, along with controlling that the imposed constraints are complied with. The algorithm then assigns a value of either $0(N_{peaks} \leq 33)$ or $1(N_{peaks} > 33)$ to a new vector $\bar{P}$. These values then decide where to create the next search grid from, as it sets the new minimum diffusion length to $\sigma_{\min} = \mathbf{max}(\bar{\sigma}(P_i = 0))$ and the maximum to $\sigma_{\max} = \mathbf{min}(\bar{\sigma}(P_i = 1))$. This narrows down the search area and creates a finer grid. The test $\sigma_{\max} - \sigma_{\min}$ is then performed, and the deconvolution is performed once again, if $\sigma_{\max} - \sigma_{\min} > \epsilon$. If not, then the algorithm stops, presenting the final diffusion length as the last $\sigma_{\text{final}} = \sigma_{\min}$ and the final back diffused series as $D_{\text{opt}} = D(\sigma_{\text{final}})$.

The direct search method is very simple, and the main reason that this search method works is due to the investigation of the relation between number of counted peaks versus diffusion length. This relation was examined by brute force: the number of peaks given diffusion length space was computed manually from 0.01 m to 0.15 m. This showed clearly that - in the area of interest, i.e. resulting in $N_peaks = 33$ - the number of counted peaks increases as the diffusion length increases, see Figure **??**.

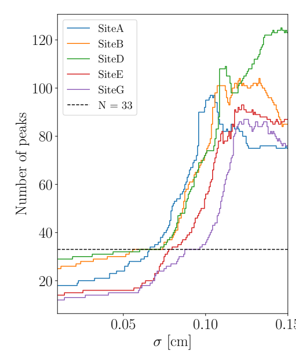

Figure 0.3: Number of peaks estimated given diffusion length, based on diffusion length in the interval [0.01; 0.15] m.

0.1.3 Constrained Peak Detection

The main constraint when back diffusing these depth series is the number of years between the two volcanic events detected in the ice. The number of especially winters between the two events are fixed to $N_P = 33$, as this number of winter is expected in the interval, even with a two month variation from the estimated event position. The counted number of summers can on the other hand vary a bit.
For better peak detection, a number of other constraints have been implemented. Since the data is a proxy for a continuous physical process, the

temperature, it is reasonable to set up constraints representing some of the logical expectations for this type of signals.

Firstly, restrictions may be demanded of the distance between peaks. The annual layer thickness, λ_A, may help with setting some limitations on the peak distances, as it is not likely for the average peak distance to be much smaller than λ_A. The peaks are expected to show the same annual cycle as the rest of the signal, representing summers.

Secondly, for the prominence of the peaks, i.e. the amplitude of the signal at a depth, it is assumed that individual peaks may not have a prominence of less than a certain percentage of the standard deviation of the signal at the given depth. This makes certain that smaller peaks or troughs, maybe representing a warm period in a winter or a cold period in summer, are not counted as annual peaks or troughs. This is one way to constrain peak prominence, but a more efficient and accurate way might be to consider the amplitude of the entire ice core signal. As it may be assumed that the amplitude and prominence at a given depth, will be somewhat smaller than the prominence of the peaks at a shallower depth, due to the general diffusion in the firn. By analyzing the amplitude attenuation of the ice core, the average attenuation at a given depth could be used as a restriction for the peak prominence. This is something that would have been implemented, if time had allowed it.

Thirdly, a constraint on how the general pattern of the trough and peak detection must look is imposed. As the temperature variations represent the change from summer to winter, it is assumed that the general pattern must be to detect a peak P, then a trough T, then a peak P, and so on, creating a pattern of ...$PTPTPTP$.... Since the deposition time of volcanic material in the ice is assumed to be Gaussian, the pattern is not restricted to start with either a peak or a trough, as this may vary when drawing a location from the distribution. Thus, the number of peaks is set at $N_P = 33$ and the number of troughs is accepted with a variation, as long as the general pattern is intact.

Finally, the diffusion length estimate is kept at a positive value with an upper limit, that can be set manually, depending on the conditions of the site and the depth.

The optimal parameter choices for the constraints are presented in:

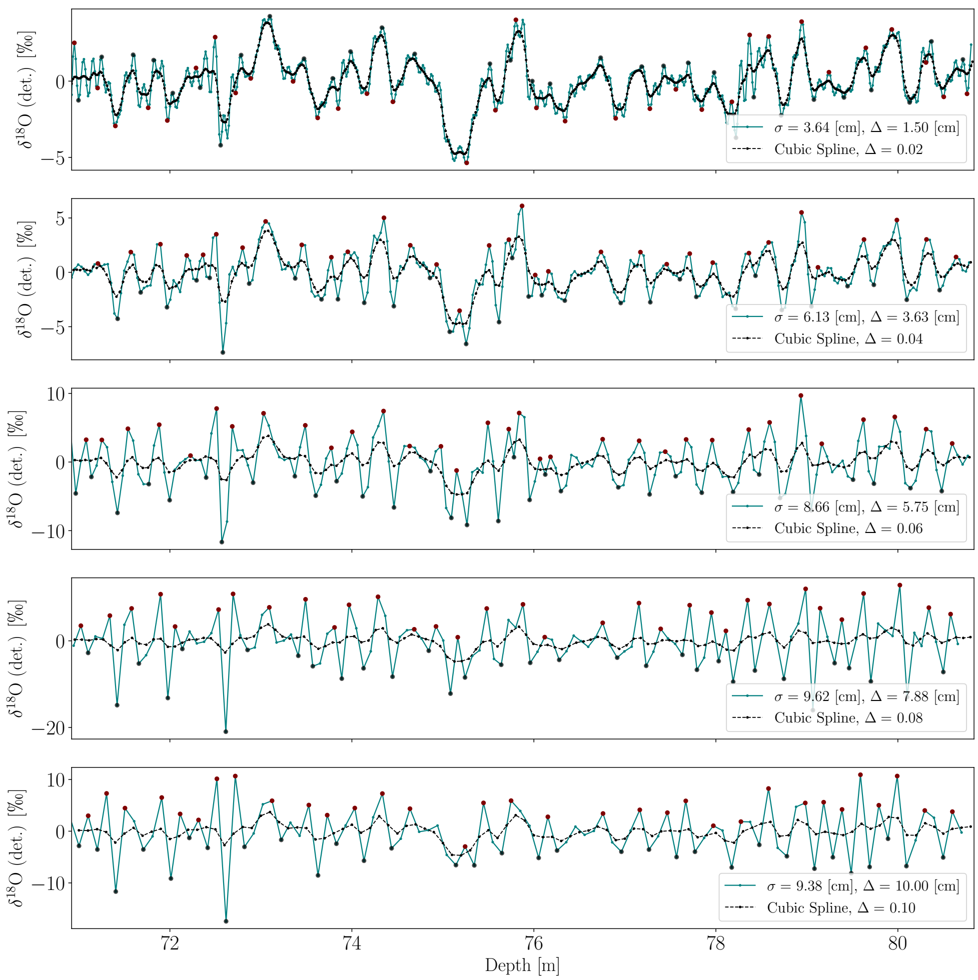

Figure 0.4: Site A, illustration of the effect of five different cubic spline resamplings before deconvolution. Original sample sizes lie in the interval [3.80; 4.00] [cm]. Resampling at smaller sample sizes show a tendency to restore some signal frequencies that are not necessarily inherit in the original signal. The resample should thus not be chosen too small as this would introduce some false signal into the results.

N_P	33
N_T	33 ± 1
Peak(trough) prominence	50 % of SD_{signal}
Peak(trough) distance	50 % of λ_A
$[\sigma_{min}, \sigma_{max}]$ [cm]	[0, 15]
Pattern	...$PTPTPTPTP$...

Table 0.1: The general constraints used in the method to optimize the diffusion length estimate.

0.2 Method Testing and Stability

Throughout this section a number of different tests of the algorithm will be presented. The tests are performed to examine the stability of the method, the accuracy of the Laki and Tambora positions and how the choice of parameters(interpolation, spectral transform type) changes the resulting diffusion length estimate.

0.2.1 No constraints versus constraints

Figure 0.5 shows the depth series between Laki and Tambora events of the ice core drilled at Site B, back diffused through the algorithm described in the above. The difference between the blue and the green back diffused signals is that the blue is back diffused using the just presented constraints, and the green signal is back diffused with only the constraint $N_P = 33$. It is clear that the imposed constraints, especially the ones concerning peak distance and prominence influences the final result of the optimal diffusion length. Particularly, the more constraint algorithm clearly leaves out from the count some 'shoulders' before an actual peak. These shoulders could be actual peaks but with the imposed constraints, they are omitted. This is something that could be further developed and examined. The appearance of these 'shoulders' could be peaks, but it is also liekly that they are remnants of some noise effect that is not quite filtered in the frequency filter construction. If that is the case, then it is a positive thing that the algorithm sorts them out.

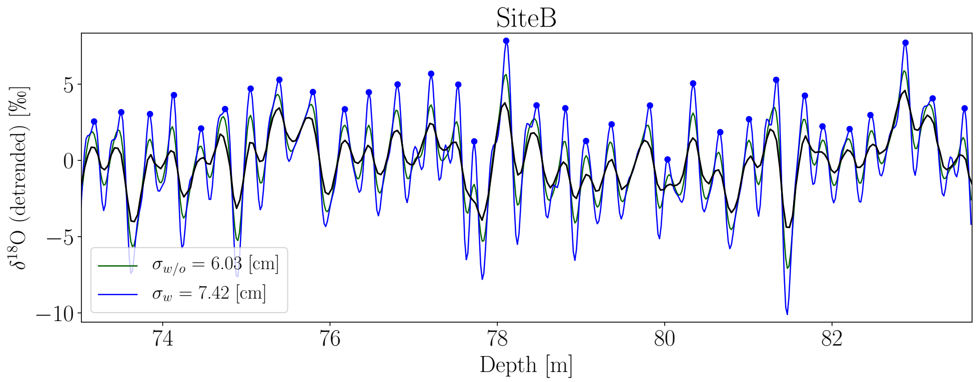

Figure 0.5: Example of how the imposed constraints effect the final diffusion length estimate for the Laki to Tamborad depth section of the core drilled at Site B. The black line shows the data, the green the back diffused data using a method with less constraints, and the blue shows the back diffused data when using the imposed constraint. The blue dots represents the peaks counted in the constrained method.

0.2.2 Effects of Interpolations

The choice of interpolation, before and after deconvolution, does affect the final result by introducing some effects not inherent in the originally measured signal. Therefore it is important to examine exactly how these interpolations influence the final σ estimate. Here that is examined be running the algorithm with different resampling sizes.

0.2.2.1 Interpolation of Data Before Deconvolution

The first interpolation is needed, if the fast spectral transforms FFT or FCT are used, as one of the conditions of the algorithms is that the data are evenly spaced. At first, this was implemented in the analysis, but this had the risk of excluding some information that might lie in the unevenly sampled data. Later, the method was abandoned in favor of implementing a nonuniform spectral transform (NUFT or NDCT), which is slower than the FFT and FCT, but carries all information from the unevenly sampled signal into the spectral domain. Luckily, this nonuniform transform needs only be carried out once, as the inverse transform, i.e. resampling in time domain, can be done uniformly without loss of information and any future spectral transforms can then be performed through FFT or FCT.

Even though the first interpolation method was later abandoned, some analysis was carried out with it to examine the effect of the size of the resampled, interpolated data on the final diffusion length estimate. Examples of a resampled signal can be seen in Figure ?? and Figure ??. Figure ?? shows how sample resolution affects information from the signal. The higher sampling resolution, the more information is retained. But higher sampling resolution also means more data to be analyzed, which might slow down any analysis algorithms developed. This might create some headache if an entire ice core length of a couple thousand meters should be examined, but for this study only af few meters are of interest, and thus it should not create delays in the computation time.

To examine the effect of the resampling resolution on the final diffusion length estimate when conducting a spline interpolation before carrying out the back-diffusion, the full diffusion length analysis has been performed with 100 new interpolation resampling sizes in the range $[\Delta_{\text{min}}; \Delta_{\text{max}}]$. This gives an idea of the stability of the method considering both sample size of the raw data and resampling by interpolation. The minimum and maximum interpolation samplings are presented in Table 0.2 and an illustration of the test results can be seen in Figure 0.6.

Figure 0.6 shows that if the sampling size of the interpolation if decreased, it becomes difficult for the algorithm to determine a diffusion length that fulfills the constraints. This is due to the spectral transforms and back diffusion method being sensitive to smaller variations that the spline interpolation introduces to the signal, as can be seen in the first panel in Figure 0.4. Furthermore, Figure 0.6 shows a less stable diffusion length estimate as the resampling size increases, and a general tendency to result in higher diffusion lengths as many features become washed out in the signal and needs more enhancement by cranking up the diffusion length estimate, see final panel in Figure 0.4. The interpolation before deconvolution is only necessary for running the method with the spectral transforms that are based on uniform sampling, i.e. the FFT and the DCT. For these two methods a choice of interpolation size can be made, but the general setting in the algorithm is to resample at the smallest sampling size found in the depth interval, Δ_{min}.

Site	Δ_{min} [cm]	Δ_{max} [cm]	Δ_{OG} [cm]
A	1.0	10.6	3.8-4.0
B	1.0	11.7	3.8-4.0
D	1.0	12.0	3.7-3.9
E	1.0	11.4	4.1-4.4
G	1.0	10.3	4.0-4.2

Table 0.2: Minimal and maximal new sample resolution used for testing interpolation before back-diffusion. Each test is run with 100 different new sample resolutions between

0.2.2.2 Interpolation of Data After Deconvolution

The second interpolation is carried out after deconvoluting and back-diffusing the signal, but before detecting peaks. Splines are especially effective when trying to find features like peaks in data which underlying signal is continuous,

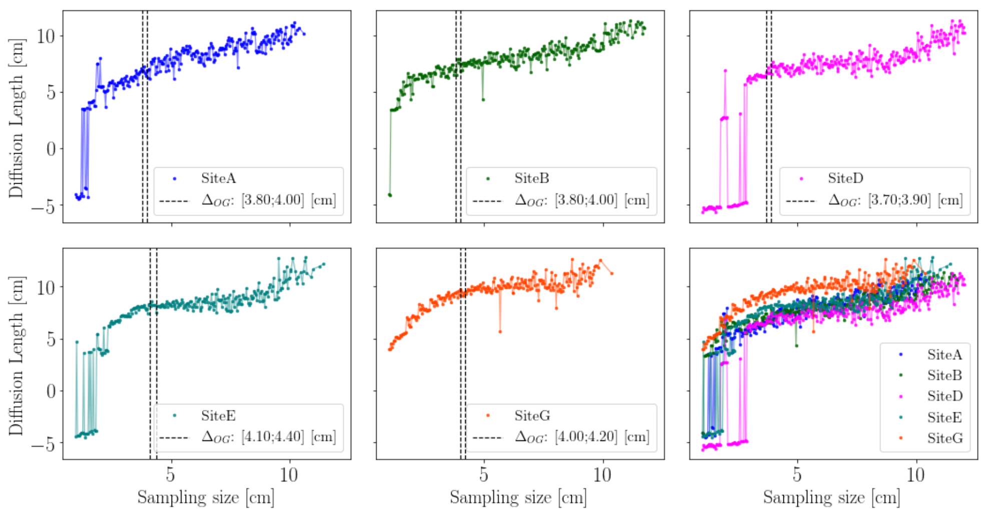

Figure 0.6: Diffusion length estimates versus resamples through cubic spline interpolation before deconvolution for Alphabet cores from sites A, B, D, E and G.

smooth and differentiable, but the sampling is discrete and thus the data are discrete and non-smooth. The isotopic signal under examination here is assumed to be truly smooth and continuous throughout the core - unless any gaps are present. Thus the cubic spline interpolation is a good tool for estimating a higher resolution version of the final back-diffused data series to use for peak detection. This makes the detection of peaks and troughs more precise, as there might not be a discrete data point exactly at the top of a peak, but the spline interpolation then estimates where the most likely top of the peak must be, on the basis of the existing data. Examples of three different interpolation samplings are presented in Figure 0.7. The effect of resampling after deconvolution on the final diffusion length estimate is illustrated in Figure **??**.

As the actual isotopic signal is continuous and that the discretization is introduced by different measurement samplings, it is assumed that the spline interpolation after deconvolution results in a more likely peak detection, when decreasing the numerical resampling size. Therefore, for the method, a resample size of $\Delta_{\min}/2$ is chosen for interpolation after back diffusion and before peak detection. In Figure 0.8 The diffusion length estimate versus the resampling size after deconvolution and shows a convergence towards a fixed diffusion length as sampling size is decreased, and a much noisier diffu-

Figure 0.7: Site A, effect of cubic spline interpolation after signal has been deconvoluted. The interpolation is introduced to make peak detection more stable.

sion length estimates as sampling size is increased. Furthermore, at certain larger sampling sizes the sought after pattern of peaks and troughs is not even reached. This resampling is carried out both for the FFT, DCT and NDCT spectral analysis methods, as the inverse NDCT can be resampled uniformly without any loss of information.

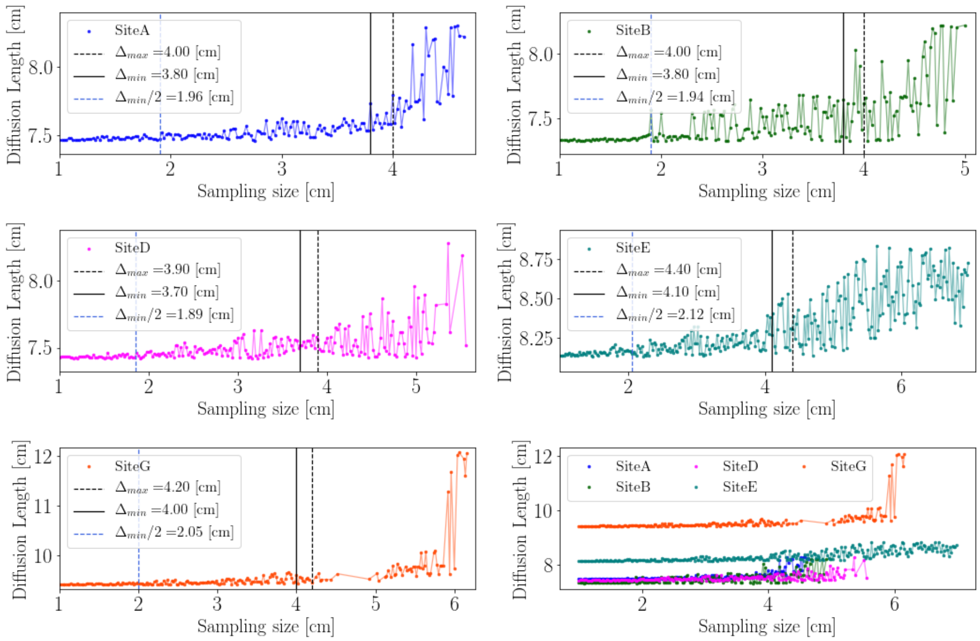

Figure 0.8: Final diffusion length estimate, given new resample by cubic spline interpolation after deconvolution for Alphabet cores from sites A, B, D, E and G.. The original sample size interval is illustrated as black vertical lines.

0.2.3 Spectral Transform's Effect on Diffusion Length

The different ways of performing spectral transformation also influences the final σ estimate, not only due to the transformation itself, but for FFT and DCT, the methods demand an interpolation, which in itself influences the results.

Figure **??** shows an example of the qualitative differences between using FFT, DCT or NDCT for spectral analysis. The most obvious visual difference between the transforms is in the end sections of the interval. This might be due to some specific boundary conditions imposed on the fast Fourier and Cosine transforms. Furthermore, by careful visual inspection, it can be seen that the NDCT seems to cater to some effects of the nonuniform samplings, that the FFT and DCT do not.

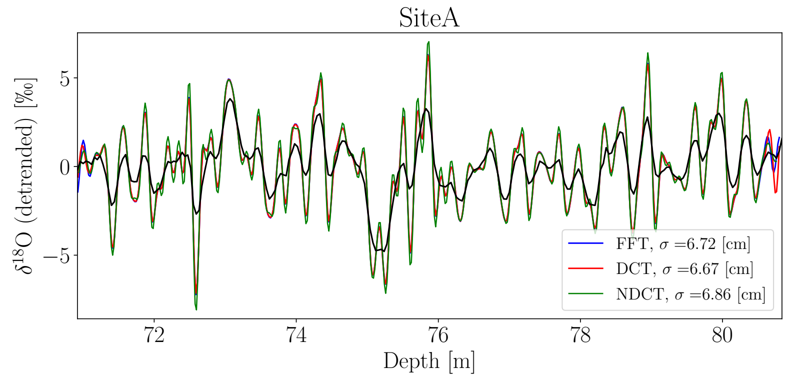

Figure 0.9: A visual example of the differences in final back diffused data when using different spectral transforms.

There are not grave differences between the three spectral transforms, but some difference there is. What might also be taken into consideration is, that this method might be run a large number of times - as is the case later on in this thesis - for example to estimate accuracy of the method, and one might therefore want to choose the method that gives the fastest results. Thus, the speed of the different transforms has been tested, as can be seen in Figure 0.10, with 200 separate runs where the Laki and Tambora positions have been drawn from a distribution with a standard deviation of 2 months from the estimated location. Not surprisingly, the NDCT is much slower than the DCT and the FFT, and it might prove efficient to choose either DCT or FFT if the algorithm has to run many times. For the case of this thesis, the final results have been made with the NDCT, so as not to miss any of the effects that might come from unevenly sampled data.

0.2.4 Laki and Tambora as Gaussian Distributions

As mentioned previously, the Laki and Tambora event depth locations are not exact. Thus to accommodate for error in this positioning, the algorithm has been tested with Laki and Tambora locations drawn from Gaussian distributions. This was examined in four ways:

- Variation in both Laki and Tambora position, corresponding to the entire events.

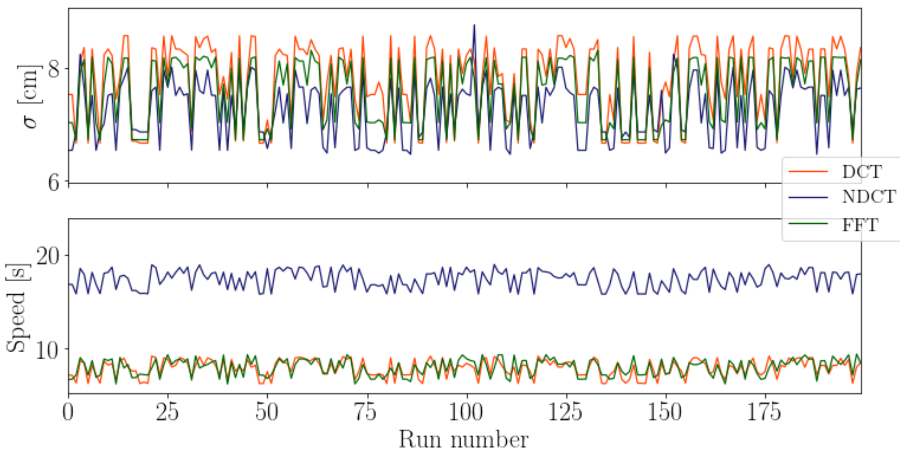

Figure 0.10: Diffusion length estimate along with speed of algorithm given the three different spectral transforms examined in this work. The volcanic event depths have been drawn from a Gaussian distribution with a width corresponding to 1 month around the estimated mean depth of the event.

- Variation in only the Tambora position while keeping a fixed section length, corresponding to the mean value $\bar{d}_L - \bar{d}_T$.
- Variation in only the Laki position while keeping a fixed section length, corresponding to the mean value $\bar{d}_L - \bar{d}_T$.
- Variation corresponding to a Gaussian distribution with a standard deviation of a depth that resembles to months in time.

The first variation of both Laki and Tambora can be seen in Figure 0.11. The method has been run 500 times with new locations drawn for each run, and the same constraints imposed each time. This results in an estimate of the diffusion length for Site A between Laki and Tambora of $\sigma = 7.32 \pm 0.67$ [cm]. But considering the lengths of the volcanic events, as seen in the ECM data, one might question this result. Due to more or else time incorporated as the depth section is widened or narrowed, some of the constraints might not be very well chosen any more. It might be that a larger depth interval could gain an extra peak or trough, so that the constraint should be $N_P = 34$ instead of $N_P = 33$. This is some work that could be continued on the algorithm, ensuring that the extra interval length increase or decrease loosens the count constraint.

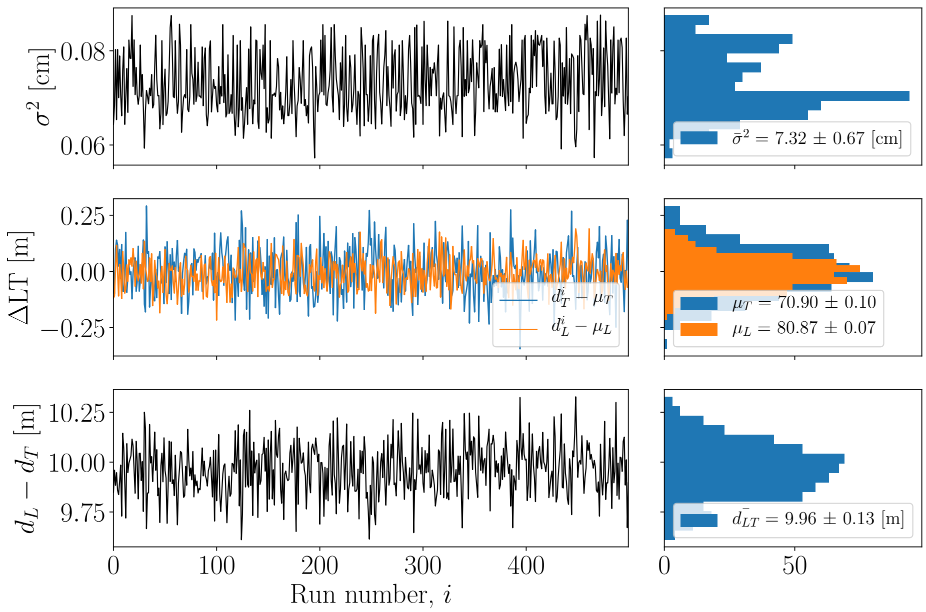

Figure 0.11: Diffusion length estimates when varying the depth locations of the volcanic events. The locations of both Laki and Tambora events have been drawn from Gaussian distributions as the ones presented in Section **??**.

The next method examines what happens when only changing the Laki or Tambora position. The results for 500 runs can be seen in Figure 0.13 and 0.12. An interesting feature shows here, and is also visible in some of the later results: the diffusion length estimates seem to not be Gaussian distributed, but concentrated around two or more different diffusion lengths. This could be an effect of the direct search algorithm, which might quantize the possible σ estimates when creating the grid. This could be examined by trying to randomize the grid creation. Furthermore, it would be interesting to examine if there is a correlation between the positioning of the depth interval and the estimated diffusion length.

The final method is an investigation of drawing the locations of both Laki and Tambora from distributions with a standard deviation of what corresponds to two months and a mean value of where the middle of the volcanic event is estimated to be. An illustration of how much this is in depth is shown in Figure 0.14. The results can be seen in Figure **??**. Again, the possible effect of the quantized grid search can be seen.

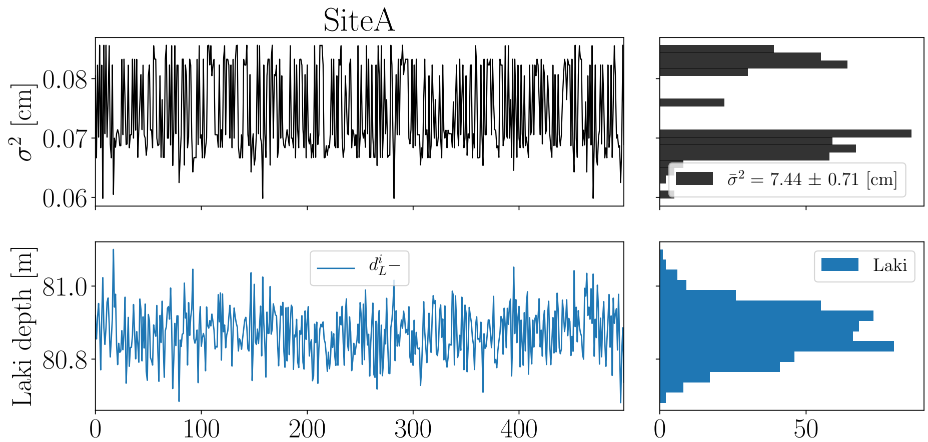

Figure 0.12: Diffusion length estimates for Site A when varying only the Laki volcanic event. The locations are drawn from Gaussian distributions as the ones presented in Section **??**. The depth section is kept at a constant length, corresponding to the mean distance value, $\bar{d}_{\mathrm{Laki}} - \bar{d}_{\mathrm{Tambora}}$.

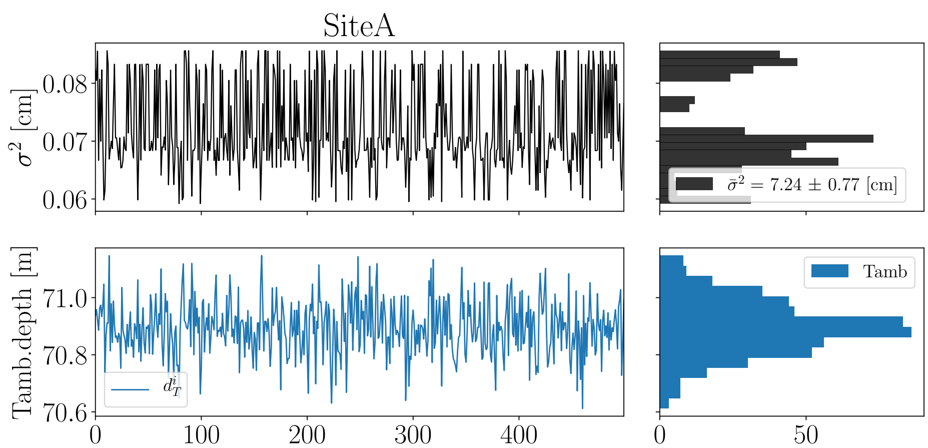

Figure 0.13: Diffusion length estimates for Site A when varying only the Tambora volcanic event. The locations are drawn from Gaussian distributions as the ones presented in Section **??**. The depth section is kept at a constant length, corresponding to the mean distance value, $\bar{d}_{\mathrm{Laki}} - \bar{d}_{\mathrm{Tambora}}$.

0.3 Possible Algorithm Upgrades

- Better peak detection, through intelligent pattern recognition.

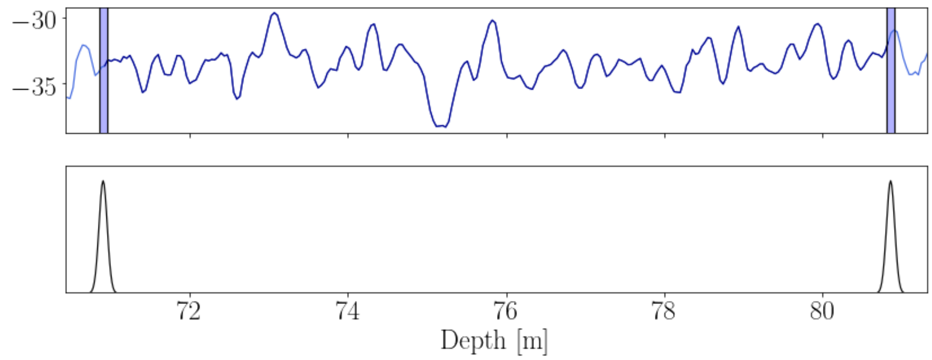

Figure 0.14: Illustration of the method implemented to manage the uncertainty of the exact depth location of the volcanic events. The method establishes a Gaussian distribution with a mean of the estimated middle of the volcanic event and a standard deviation of what corresponds to two months.

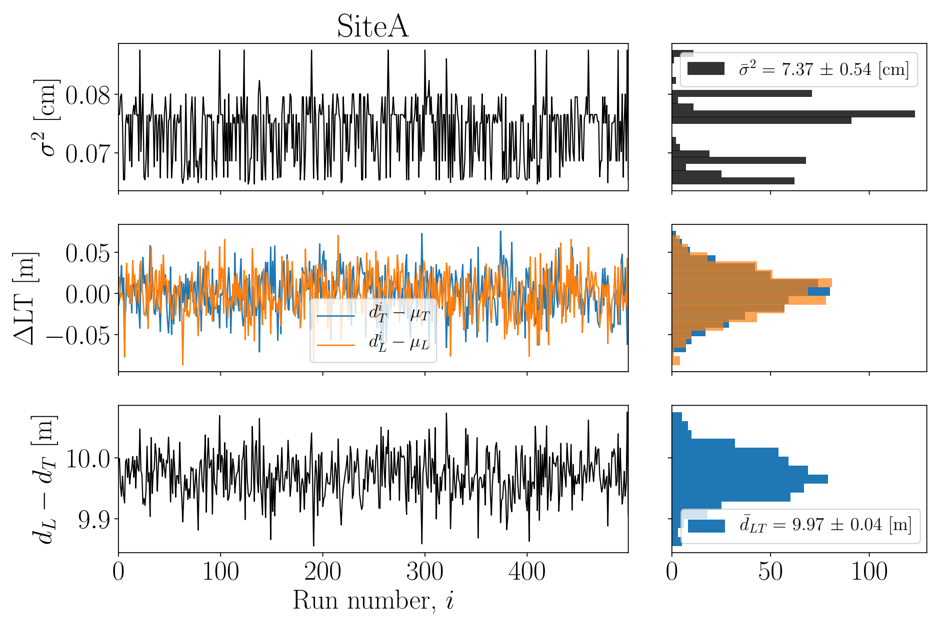

Figure 0.15: 500 runs with locations of Laki and Tambora events drawn from a Gaussian distribution with a standard deviation of two months. Using NDCT as spectral transform.

- Taking summer-FALL-winter-SPRING into account.
- More detailed constrained optimization. If length changes, chane ex-

pected number of peaks, so on.

- Examine the frequency filter a bit more - is it allowing too much noise? Maybe also make variable.
- More data! Or, higher sampling resolution.
- Examine what happens with the quantized results. Maybe create a better optimization routine with more randomization.
- Why do we look for the maximal σ to fulfill constraints? Maybe interesting to see how wide a range of σ that fulfills constraints! This could give a temperature range instead of just one temperature.
- Implement $\sigma(z)$ and figure out how to relate this to temperature.
- Figure out a way to make faster algorithm when deconvoluting with $\sigma(z)$.

0.3.1 Peak Detection

0.3.2 Optimization Routine