Geochemistry, Geophysics, Geosystems

TECHNICAL REPORTS: METHODS

10.1029/2019GC008380

Waiting for Forcot: Accelerating FORC Processing 100× Using a Fast-Fourier-Transform Algorithm

Thomas A. Berndt[1] **and Liao Chang**[1]

[1]Laboratory of Orogenic Belts and Crustal Evolution, School of Earth and Space Sciences, Peking University, Beijing, China

Abstract First-order reversal-curves (FORC) are a widely used tool to analyze magnetic mineralogy and domain states, but require extensive processing—in particular, smoothing—to be plotted as FORC diagrams. Currently, smoothing is a computationally complex task involving repeated least squares surface optimization routines, sometimes taking minutes to compute for high-resolution FORCs (leaving users with the feeling of "waiting for Godot," who never comes). Here we show how the same computation can be carried out much more efficiently in Fourier space and present a new FORC processing software, called Forcot. The new algorithm, combined with an improved user interface enables users to create print quality FORC diagrams within a few seconds. Processing times are shown to be reduced by factors from 2 to 100 depending on size and smoothing factor compared to existing FORC smoothing algorithms. Additionally, optimal smoothing factor can be determined directly from the noise spectrum in Fourier space and does not require repeated smoothing of diagrams as in previous programs. Finally, formatting of figures is done automatically by our software such that diagrams can be directly used for print.

Key Points:
- FORC diagrams are smoothed using fast Fourier transforms reducing processing times up to 100×
- Optimal smoothing is estimated automatically from power spectrum
- Print-quality FORC diagrams can be created in seconds

Supporting Information:
- Supporting Information S1

Correspondence to:
T. A. Berndt,
tberndt@pku.edu.cn

Citation:
Berndt, T. A., & Chang, L. (2019). Waiting for forcot: Accelerating FORC processing 100× using a fast-fourier-transform algorithm. *Geochemistry, Geophysics, Geosystems*, *20*. https://doi.org/10.1029/2019GC008380

Received 11 APR 2019
Accepted 14 NOV 2019
Accepted article online 25 NOV 2019

1. Introduction

First-order reversal curve (FORC) diagrams (Pike et al., 1999; Roberts et al., 2000) have become a routine tool to assess domain states, magnetostatic interactions, and mineralogy of magnetic mineral assemblages. While some recent studies have focused on quantitative analyses of FORC raw data, for example, using principal component analysis (PCA) to unmix contributions of different magnetic minerals (Lascu et al., 2015; Channell et al., 2016; Harrison et al., 2018; Roberts et al., 2018) or obtaining coercivity distributions (Egli, 2013), visual inspection of FORC diagrams remains the most commonly used way to interpret both domain states and mineralogy qualitatively and to get an intuitive understanding of the measured data. This procedure has been successful at characterizing domain states of samples, for example, distinguishing pseudo-single-domain behavior from mixtures of single-domain (SD) and multidomain behavior (Muxworthy et al., 2005), magnetic mineral identification (Roberts et al., 2006; Egli et al., 2010), and at providing insights into variations of environmental magnetic records (Yamazaki, 2009; Chang et al., 2014; Channell et al., 2016; Chang et al., 2016), among others (Roberts et al., 2014). Hence, while quantitative/mathematical methods are becoming more common place, FORC diagrams that are easy to interpret visually remain an important tool for many applications.

Therefore, processing FORC diagrams currently often resembles waiting for Godot: one starts the smoothing of a diagram but never knows how long it will take, or if the result will even be of usable quality. Here we propose a new way to view the problem of FORC diagram smoothing that leads to a significant increase in speed of FORC processing—for large diagrams by an order of magnitude. We further implemented a new processing software *Forcot* with a particular focus on convenience, speed, and ease of use to produce print-ready figures with minimum time and effort.

For the new algorithm, we view FORC smoothing as a pure image processing problem: rather than considering a collection of partial hysteresis curves, we view the mixed second derivative (the FORC distribution) as pixels of a (very) noisy image. The aim then is to filter the noise from this image to a degree that is visually optimal in the sense that real-signal features should visually appear more prominent than the noise. The difference between noise and real signal is that noise should be uncorrelated, while signal should be correlated between neighboring points in the FORC distribution—that is, noise tends to have high frequencies while signal tends to have low frequencies in the power spectrum/2-D Fourier transform of the

image. To produce smoothed FORC diagrams, one therefore simply has to filter the high frequencies from the image to a sufficient degree that the image is dominated by low frequencies. The beauty of this approach is that the whole process of calculating mixed derivatives, determining optimal smoothing factors, and computing smoothed diagrams, can be executed in Fourier space, where the whole problem reduces to a simple multiplication, as outlined below. The computation is thereby simplified to a degree that enables to create smooth FORC diagrams in near-real time in the sense that FORC raw data files can be loaded, processed and viewed in about the same time it takes to open a picture file on a computer. Therefore, unlike Samuel Beckett's infamous protagonist, one will never have to wait for *Forcot* for more than a few seconds.

Forcot is freely available on https://thomasberndt.github.io/Forcot; its MATLAB source code is available on https://github.com/thomasberndt, both are also available on 4TU.ResearchData (10.4121/uuid:3964b8d9-8e5d-4c6e-849f-b6bc10834bab). *Forcot* is optimized for usability for most common applications, that is, simple FORC smoothing, basic plotting, and export to the most common graphics formats in publication-ready quality. *Forcot* supports Windows and Mac and can process Princeton and LakeShore vibrating sample magnetometer (VSM) and alternating gradient magnetometer (AGM) files.

2. Theory

FORCs are partial hysteresis loops: first, a sample is saturated in a large positive field, then the applied field is reduced to the reversal field H_a, and finally a partial hysteresis loop is measured while slowly ramping up the field from H_a back to (positive) saturation. The field at which this hysteresis loop is measured (the "measurement field") is denoted H_b. The process is then repeated N (commonly 100–200) times for a range of reversal fields H_a. All the obtained partial hysteresis loops together can be written as $M(H_a, H_b)$. The FORC distribution $\rho(H_a, H_b)$ is defined as the mixed second derivative of M, that is,

$$\rho(H_a, H_b) = -\frac{1}{2}\frac{\partial^2 M(H_a, H_b)}{\partial H_a \partial H_b}. \tag{1}$$

Normally the FORC distribution is plotted in a rotated coordinate system $H_c = (H_b - H_a)/2, H_u = (H_b + H_a)/2$ as a contour plot, the FORC diagram. Let $\mathcal{F}[f]$ denote the two-dimensional Fourier transform (FT) of the function $f(x, y)$, then the FT of M is

$$\mathcal{F}[M(H_a, H_b)] = \int M(H_a, H_b) e^{-2\pi i(aH_a + bH_b)}\, \mathrm{d}H_a\, \mathrm{d}H_b\ , \tag{2}$$

where the FT is a function of the wave numbers a and b. Using the well-known identity for the FT of derivatives, the FT of ρ is

$$\mathcal{F}[\rho] = (2\pi i)^2 ab \mathcal{F}[M]\ . \tag{3}$$

Hence, in Fourier space, the problem of calculating the mixed derivative of the FORCs reduces to a simple multiplication. The second step of FORC processing is the smoothing algorithm: as calculating second derivatives amplifies noise, it is necessary to smooth the diagrams before plotting them. This commonly involves one or another variety of running mean algorithm: for each point (H_a, H_b) in the FORC space a window of a set number of neighboring points is chosen and a (weighted) average is calculated and used as the center point of $\rho(H_a, H_b)$. Most current algorithms are varieties of the original smoothing algorithm by Pike et al. (1999), which used a least squares polynomial surface fit in the $M(H_a, H_b)$ plane to the neighboring points within a square containing $(2SF+1)^2$ points (where SF is the smoothing factor), and calculating ρ subsequently from the fit. The popular LOESS algorithm by Harrison and Feinberg (2008) is a modification of this that uses the nearest $(2SF + 1)^2$ points contained within a circular region, and locally weighting by a tricube weight function that assigns higher weights to points closer to the center point (H_a, H_b). More recent algorithms such as VARIFORC (Egli, 2013) and irregularFORC (Zhao et al., 2015) are modifications of this approach that use different smoothing factors in different regions of the FORC diagram. Common to all these approaches is that for each point in the FORC plane an average value has to be computed by means of a least squares surface fit, which is computationally intensive: for a diagram consisting of N individual FORCs, N^2 least squares routines have to be run. An optimization of this computational problem was suggested by Heslop and Muxworthy (2005): the least squares surface fit can be reformulated as a Savitzky and Golay

(1964) convolution, greatly simplifying the least squares routine, however, still requiring N^2 convolutions to be calculated.

We will now show how this problem is greatly simplified in Fourier space. Any of the above mentioned methods can mathematically be expressed in terms of a windowing function $w(h_a - H_a, h_b - H_b)$. For each point (H_a, H_b), this windowing function selects the $(2SF + 1)^2$ neighboring points of (h_a, h_b) and suppresses all points outside the window. For example, in the original algorithm (Pike et al., 1999), $w(h_a - H_a, h_b - H_b)$ is one for $h_a - SF \leq H_a \leq h_a + SF$ and $h_b - SF \leq H_b \leq h_b + SF$, and zero elsewhere. In the LOESS algorithm (Harrison and Feinberg, 2008), the windowing function additionally serves as a weight function and is given by the tricube function in equation (5) in their paper, if one understands that w is zero outside the circular area of radius SF. Hence for each point (H_a, H_b), the smoothed FORC distribution $\hat{\rho}$ is

$$\hat{\rho}(H_a, H_b) = \int_{-\infty}^{+\infty} \mathrm{d}h_a \int_{-\infty}^{+\infty} dh_b \; \rho(h_a, h_b) w(h_a - H_a, h_b - H_b) \; , \tag{4}$$

or a similar expression with M in the integral if smoothing is performed before taking the mixed derivative. We see that smoothing is simply the convolution $(\rho \star w)(H_a, H_b)$ of the FORC distribution with an appropriate windowing function. In Fourier space, however, a convolution reduces to a simple multiplication, and we have

$$\mathcal{F}[\hat{\rho}] = \mathcal{F}[\rho \star w] = \mathcal{F}[\rho]\mathcal{F}[w] \; . \tag{5}$$

This reduces having to calculate N^2 running means involving a total of $(2SF + 1)^2 N^2$ data points (each obtained by a least-square optimization routine in practice), to a simple multiplication of N^2 numbers. Combining eq. (3) and (5), and denoting the FT of w by W, we obtain

$$\mathcal{F}[\hat{\rho}] = (2\pi i)^2 abW\mathcal{F}[M] \; . \tag{6}$$

Finally, the smoothed FORC distribution $\hat{\rho}$ is obtained by taking the inverse FT $\mathcal{F}^{-1}$. By writing W instead of $\mathcal{F}[w]$ we make clear that it is not necessary to numerically calculate the FT of w—rather, a suitable function W can be chosen directly. Hence the whole problem of FORC processing including both the calculation of mixed derivatives and the smoothing is reduced to a FT, a multiplication and an inverse FT. For simplicity, for the window w we chose a Gaussian distribution

$$w(h_a, h_b) = \frac{1}{\pi SF} \exp\left\{ -\frac{(h_a^2 + h_b^2)}{2(SF/2)^2} \right\} , \tag{7}$$

where we defined the SF as 2 standard deviations of the distribution (2 standard deviations is commonly considered the range where the distribution becomes "very small"; hence, this definition roughly corresponds to the definition used by Harrison and Feinberg (2008) for the tricube function, which is defined to be exactly zero at a distance of SF away from the point to be calculated). The FT of a Gaussian is a Gaussian itself,

$$W(a, b) = \exp\left\{ -\pi^2 SF^2 (a^2 + b^2)/2 \right\} . \tag{8}$$

The final problem to resolve is the determination of the optimal SF. A previous method (Harrison and Feinberg, 2008) consisted of calculating various smoothed FORC diagrams with different SFs, calculating residuals between smoothed and unsmoothed diagrams and selecting an SF where the marginal decrease in standard variation of residuals with a marginal increase in SF is minimal. This procedure therefore implicitly assumed that noise should follow a normal distribution (while the signal is assumed not to follow a normal distribution) such that the standard deviation of residuals is a proxy for the amount of noise filtered out. Similar to Heslop and Muxworthy (2005), we propose a measure of noise that is independent from the distribution of signal and noise—we assume that noise is mostly contained in the high frequencies, while the signal is mostly contained in the low frequencies. Note that some features, such as central ridges, are high frequency in one axis, but low frequency in the other, and are therefore not considered noise. We therefore calculate the power spectrum from the Fourier transform $\mathcal{F}[\hat{\rho}](SF)$ by integrating over a circle defined by $SF^{-2} = h_a^2 + h_b^2$; the spectrum then normally has high values at low frequencies (signal), and high values at high frequencies (noise), and a minimum in between. FORC diagrams that are intended for visual inspection/qualitative interpretation, need an SF such that the signal in the diagram visually appears much more prominent than the noise; this is achieved by choosing the smallest SF that

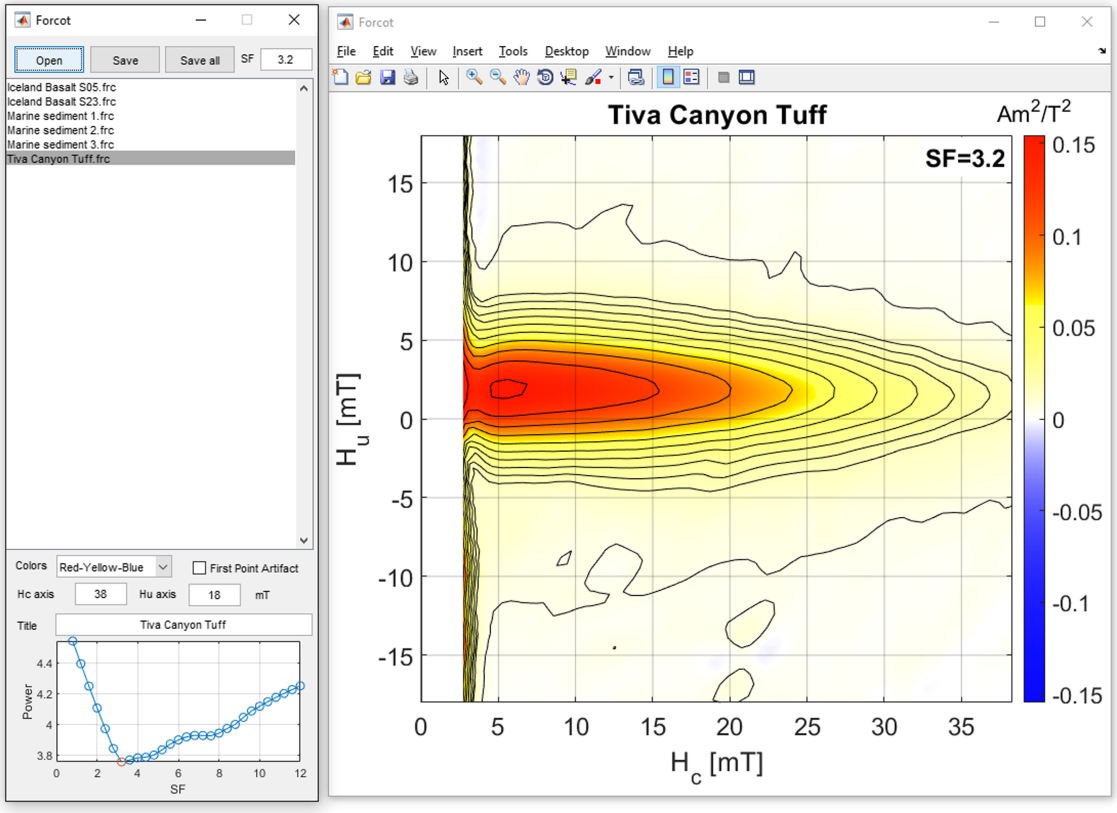

Figure 1. Screenshot of the user interface of our new FORC processing software Forcot.

completely removes the high frequencies (more specifically the lowest SF with a power in the lowest tenth quantile is selected).

3. Software: Forcot

Figure 1 shows a screenshot of our new FORC processing software *Forcot*. The user interface consists of two windows, the left window containing user controls and the right window showing the processed FORC diagram. The top part of the control window shows buttons to open FORC files, save the diagram(s) and to adjust the SF. After clicking open and selecting a FORC file on the hard drive, the selected file and all other files with the same ending (i.e. commonly '.frc') in the same folder will be displayed in the list below the open/save buttons. This allows the user to quickly browse through a large number of FORC files. When selecting one (by clicking or using the arrow keys on the keyboard) the file is automatically loaded, processed, and displayed—the high efficiency of the FFT algorithm means that all but the largest FORC files will be displayed in less than a second, making the process similar to browsing through a collection of preprocessed pictures. For each file, the default SF is the optimal SF as described in the previous section, but can be adjusted using the control in the top-right corner. Entering a new SF (which can be a decimal fraction) and hitting Enter immediately shows the reprocessed FORC.

The bottom part of the control window offers some adjustments to the diagram: The color scheme can be chosen, the first point at $H_c = 0$ can either be plotted or hidden (see Section 4.3 for a discussion of the first point artifact), and the axis limits can be adjusted, as well as the title of the diagram. Note that alternatively the default MATLAB functions for plotting are available in the figure window. Finally, the very bottom of the control window shows the power spectrum of the FORC diagram, with the optimal SF highlighted in red. When adjusting the SF, it should be chosen in the area where the value of the power spectrum is small. By default, the first point $H_c = 0$ is not plotted, that is, roughly the area up to SF times the step size of H_c

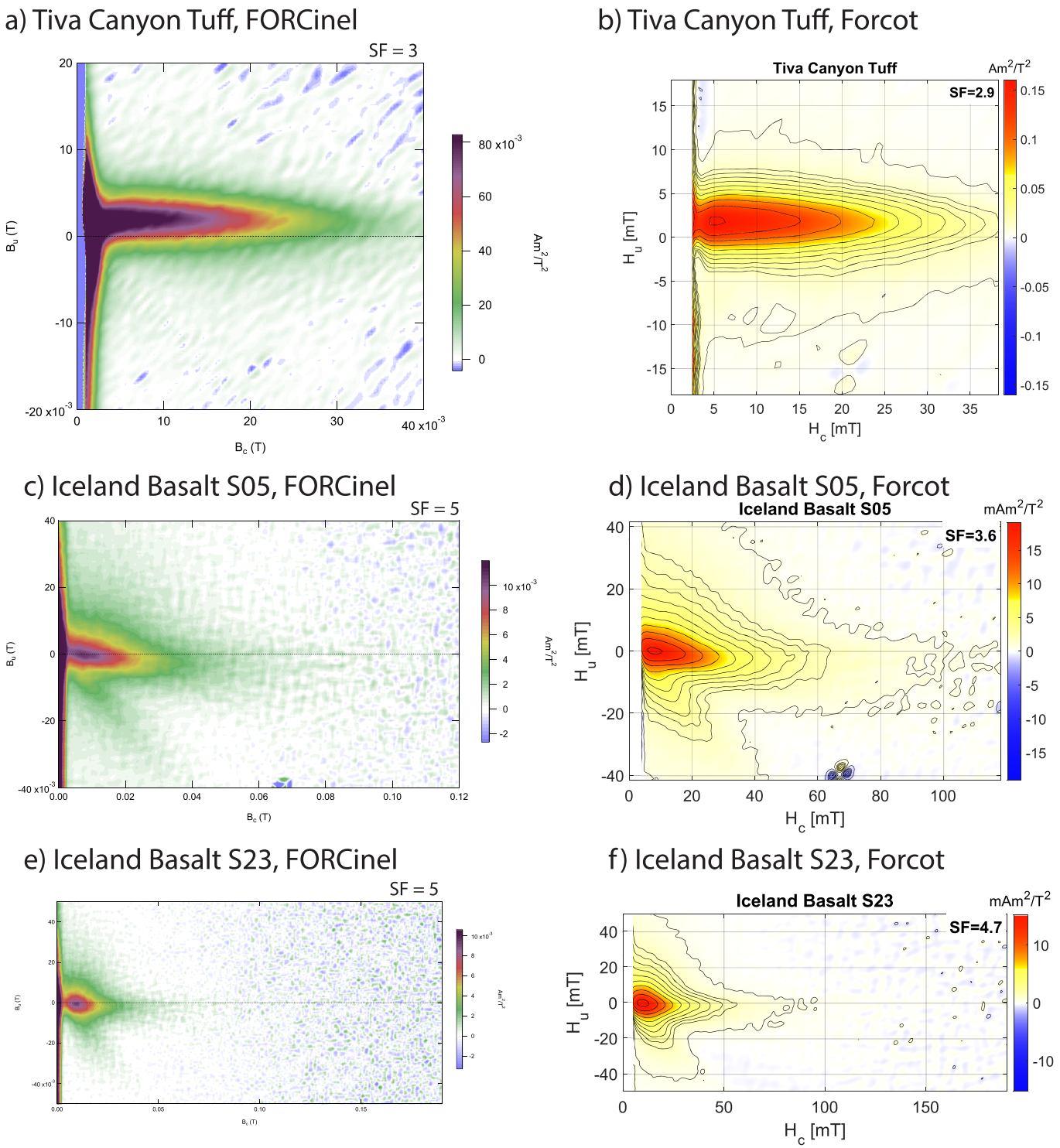

Figure 2. Comparison between FORCinel (Harrison and Feinberg, 2008) and Forcot processing for a Tiva Canyon Tuff sample and two basalt samples from Iceland. No postediting of the graphics was done other than adding the SF to the FORCinel plots. Contour lines were not added in the FORCinel plots, and font sizes were not adjusted.

is left blank—this is because this area is normally dominated by an artifact due to calculating derivatives where there are no data points at negative H_c. Finally, *Forcot* has a function for batch processing of FORCs: the button "Save all" will process all FORCs in the list with their respective default optimal SF and save the figures in the selected folder. Both vector and pixel graphics are supported (png, pdf, eps, jpg, tif, bmp, and MATLAB fig) and the figure is saved with the same dimensions as the figure window.

4. Results and Discussion

4.1. Smoothing Quality

Figure 2 shows sample FORC diagrams for three natural samples: one SD (low-Ti titano) magnetite-dominated Tiva Canyon Tuff sample (Till et al., 2011) and two magnetite-dominated basalt samples (and some TM60 titanomagnetite) from Iceland (Berndt and Muxworthy, 2017). All diagrams were processed both in Forcot and in FORCinel (Harrison and Feinberg, 2008), which is one of the most popular FORC processing programs currently in use. The primary aim of FORCinel was to achieve a constant smoothing over measurement grids with missing data points and/or irregular grids, as well as achieving some extrapolation toward the first point at $H_c = 0$ (which is actually undefined due to the lack of data at negative H_c when calculating derivatives). Forcot, by contrast, has the primary aim to provide fast processing times, ease of use and direct export of print quality figures (for this reason, the target (H_a, H_b) grids are used from the VSM settings, rather than the actual measured values). Nevertheless, FORCinel has been chosen as a standard for comparison due to its large popularity. The figures presented are all shown without any postediting using vector graphics software, other than adding the SF to FORCinel figures (these are added automatically in Forcot), and adding subfigure titles (titles are actually also added automatically by Forcot). Regarding the SF, FORCinel uses the LOESS algorithm, in which raw data points are weighted according to their distance to the processed FORC diagram data point—this is similar to the Gaussian convolution in Forcot. However, the exact weighting is slightly different (FORCinel uses a tricube weight function, Forcot a Gaussian), leading to different nominal SF yielding similar apparent smoothness. The SF presented for Forcot are those automatically selected by the software in Figure 2; the SF in FORCinel were chosen manually to achieve a similar apparent smoothness. Other than the first-point-artifact (cropped off in Forcot), the FORC distributions are very well visible in the diagrams of both programs and show the same shape. Some features, particularly at the fringes of the FORC distribution are better visible in the Forcot diagrams; however, this is probably due to the color scale and the addition of contour lines rather than the smoothing algorithm. Forcot does, however, appear to remove the noise at high field somewhat better than FORCinel, giving the diagrams a somewhat cleaner look.

Figures 3–5 show high-resolution FORC diagrams for three relatively weak sedimentary samples (Chang et al., 2016; Chang et al., 2018), one measured on a VSM and two measured on an AGM. The data files for these FORCs were very large, ~3, 6, and 12 MB, respectively. Each sample is shown with two different SF, 5 and 7 for FORCinel, and the automatically chosen SF for Forcot, as well as a manually chosen SF of 3. Again, the FORC diagrams agree well between FORCinel and Forcot. The automatically chosen SF generally shows the signal very clearly, but is on the conservative side to avoid oversmoothing. Depending on preference, users might prefer to increase the SF slightly.

4.2. Processing Times

Table 1 lists the processing times of the various FORC diagrams shown above as obtained on a modern HP Windows workstation (Processors: 2 × Intel Xeon E5-2623 v4 @ 2.60GHz, 2601 Mhz, 4 Core; 64 GB RAM; similar times were obtained on a Macbook Pro with an Intel Core i7 2.5 GHz, 16 GB RAM). Comparing the processing times, the main advantage of Forcot, the FFT smoothing algorithm, becomes evident. In Forcot, only one step is required to produce print quality FORC diagrams: loading a file immediately processes and plots the diagram. In FORCinel file handling and processing are two separate steps, whose times are reported individually in Table 1. The table shows that in all cases the total loading + processing + plotting time of Forcot is faster than only loading a file in FORCinel. For the smallest FORC diagrams, processing times are only a few seconds in both programs, with Forcot being ~2 to 10 times faster, depending on SF and file size. This means that Forcot typically produces FORC diagrams from raw data in about 2 s or less. For the larger FORC files that require larger SFs, the speed difference becomes very large: smoothing the 12 MB high-resolution FORC with SF = 7 required 238 s in FORCinel, but only 6.6 s in Forcot. Considering that the pure processing part in Forcot was only 1.2 s (excluding loading the VSM/AGM file and plotting), the processing time of the FFT algorithm is more than 100× faster than the conventional LOESS algorithm.

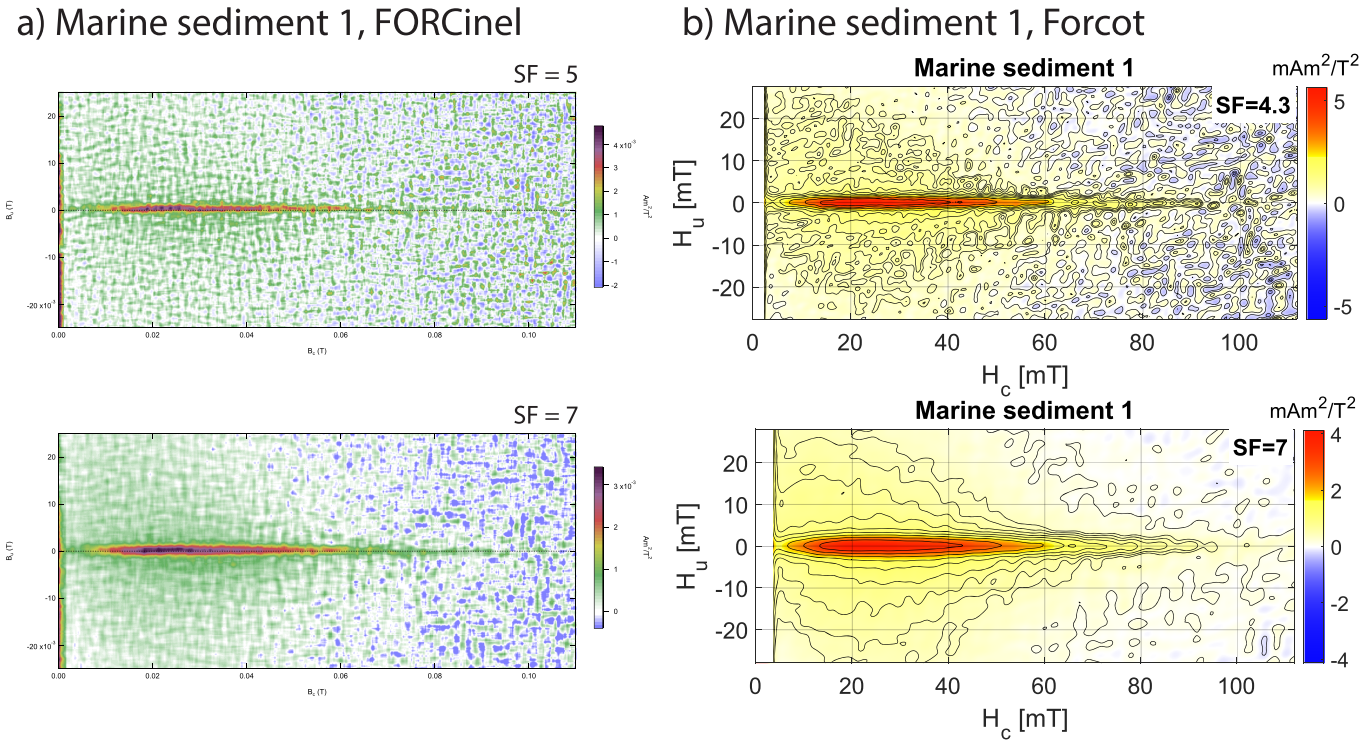

Figure 3. Comparison between FORCinel (Harrison & Feinberg, 2008) and Forcot processing for a marine sediment sample (3.5 MB FORC file, measured on a VSM). No postediting of the graphics was done other than adding the SF to the FORCinel plots. Results of two different SF are shown.

This combined with the fact that the optimal SF is selected from the FFT power spectrum rather than repeatedly reprocessing a FORC, means that optimally smoothed high-resolution diagrams are created in seconds in Forcot compared to tens of minutes in FORCinel. On older computers (as those in many laboratories), the time difference is likely significantly larger, making Forcot an excellent choice to view FORC diagrams immediately after measuring in the laboratory.

The Savitzky and Golay (1964) convolution method proposed by Heslop and Muxworthy (2005) was estimated to yield up to a 500× reduction in processing time compared to polynomial surface fitting for large high-resolution FORC—in fact a similar factor is obtained in Forcot when not using the automatic SF selection routine (e.g., setting the SF to 5 in our 12 MB FORC took 0.6 s processing time, plus plotting time). The SF selection routine is likely much faster in Fourier space than the calculation of the autocorrelation that used in the approach by Heslop and Muxworthy (2005) since it does not require repeated calculation of the complete smoothed FORC diagrams. Note, however, that the two approaches are in fact equivalent, since the Fourier transform of the autocorrelation function is simple, the Fourier power spectrum itself. This allows Forcot to select the optimal SF from a range of 50 smoothing factors (logarithmically spaced between 0.1 and 10).

4.3. Edge Artifacts

One of the main problems of FORC diagrams is that one important set of data is generally missing in FORC data: the derivative of the "first point" in each FORC, that is, the vertical left axis ($H_c = 0$) of the diagram. To calculate the derivative of the FORC data, at least two data points are needed, that is, the one corresponding to 0 coercivity (i.e., the first measured point in each curve) and the next measured data point above that (i.e., the second measured point in each curve). This means that the derivative, and hence the FORC distribution, at the $H_c = 0$ is not well-defined—the first point is "missing." Since this area is, however, often of interest to rock magnetic studies (e.g., superparamagnetic particles plot here), there are various approaches to address this problem, most involving some form of extrapolation of the derivative at $H_c = 0$ using data only from $H_c > 0$ (Roberts et al., 2014). The problem is more severe after smoothing the diagram, since in order to smooth with SF of n, it requires n data points in the raw FORC to calculate the smoothed derivative. Ultimately, the only way to obtain an accurate FORC diagram near $H_c = 0$ is to increase the number

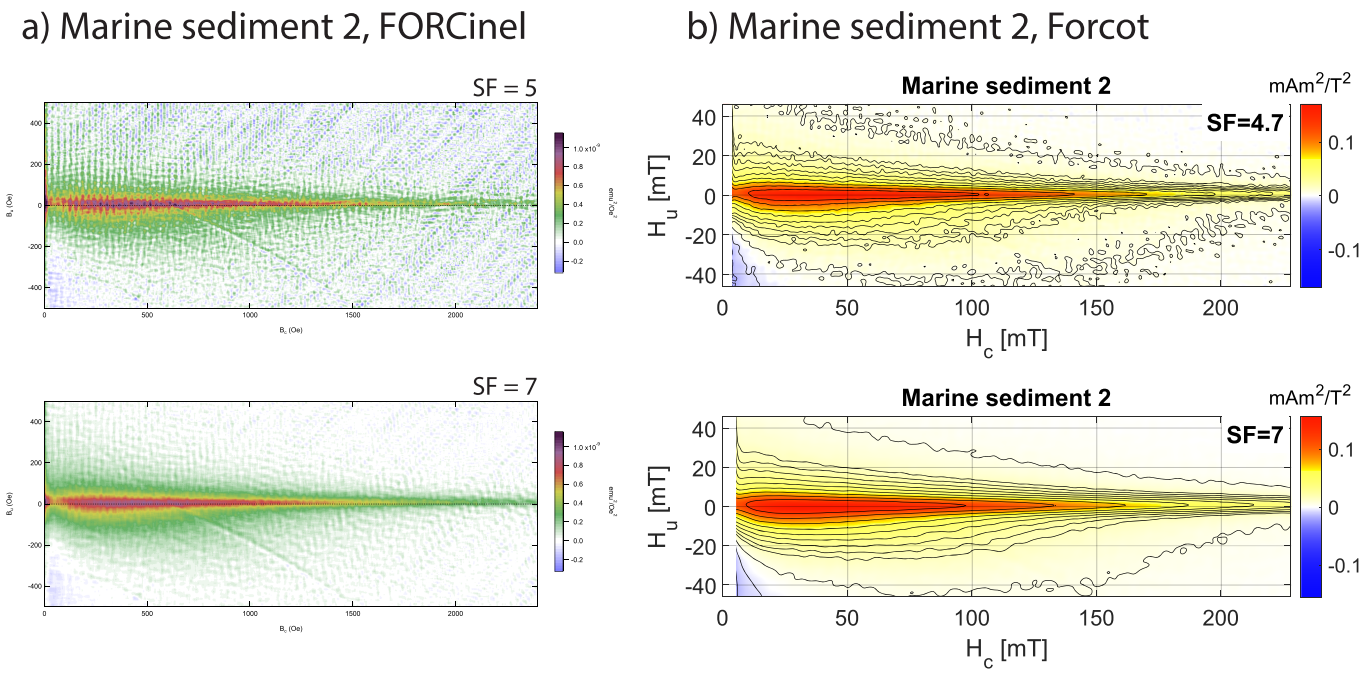

Figure 4. Comparison between FORCinel (Harrison & Feinberg, 2008) and Forcot processing for a marine sediment sample (6 MB FORC file, measured on an AGM). No postediting of the graphics was done other than adding the SF to the FORCinel plots. Results of two different SF are shown.

of measured data points close to 0; if this is not possible, the choice of how the derivative of the FORCs is extrapolated toward 0 will determine the visual appearance of the left edge of the diagram —in other words choosing the correct measurement resolution is a prerequisite, regardless of the processing strategy. For this reason, we argue that it is best not to show the area of the diagram where no data points are available (default in Forcot), we have chosen not to plot the first point by default in Forcot.

Since the computation is done in Fourier space, however, extrapolating "missing" data points has got a role beyond extrapolating toward the left axis: The FFT is only defined for complete data over a whole rectangular domain, (H_a, H_b), in this case. Hence, it is necessary to extrapolate all the points in this space (even those corresponding to negative coercivities). Since each frequency in the Fourier representation of the diagram depends on all the data points, including the missing/extrapolated ones, an 'inappropriate' choice of extrapolation values could cause high wave powers in the Fourier spectrum and hence reduce the effectiveness of the smoothing even in the diagram area where data are available. To this end, a linear extrapolation along the H_a dimension from the first measured data point to the maximum measured M was chosen (i.e., $M(H_a, H_b) = (M(H_{a,0}, H_b) - \max(M)) \times H_a/H_{a,0}$ for $H_a < H_{a,0}$, and $M(H_a, H_b) = (M(H_{a,1}, H_b) - \min(M)) \times H_a/(H_{a,1} - \max(H_a))$, for $H_a > H_{a,1}$, where the subscripts 0 and 1 indicate the first and the last "missing" data point, respectively (for any fixed H_b), and min and max refer to the global minimum and maximum values, respectively), which was found to yield lower wave powers than extrapolation with constant values. Nevertheless, since the extrapolated values do contain some wave energy in the spectrum, some degree of Gibbs oscillations occur at the edges of the FORC diagram due to the use of FFT, both at the $H_c = 0$ axis, and at the top, bottom, and right axes (where they are cropped off by Forcot; see Figure S1 for an example showing the first point), though the use of a Gaussian smoothing kernel reduces the ringing amplitude (at the cost of spreading it over larger areas). Nevertheless, the occurrence of Gibbs oscillations at the edges remains a limitation of the FFT approach.

4.4. Advantages and Disadvantages of Polynomial Surface Fitting

Traditionally, FORC diagrams have been smoothed using polynomial surface fits (Roberts et al., 2000). While this choice may initially not have been motivated by any particular reason other than providing one way to calculate smoothed mixed derivatives, polynomial fitting has a number of advantages: most notably they enable a way to estimate significance levels/confidence intervals for FORC diagrams (Heslop and Roberts, 2012), but they are also used for some implementations of FORC-PCA (Harrison et al., 2018). Since it is, however, clear that polynomial surface fits are nothing else but a convolution of the FORC distribution with

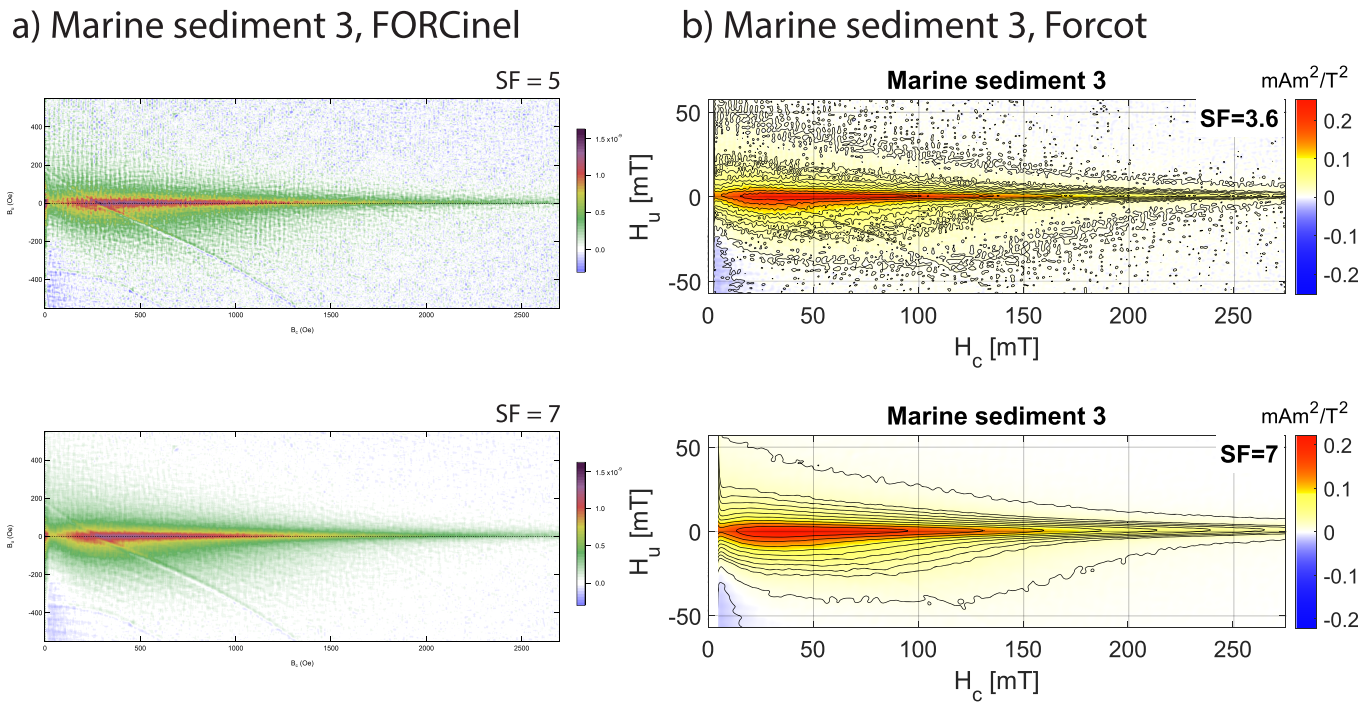

Figure 5. Comparison between FORCinel (Harrison & Feinberg, 2008) and Forcot processing for a marine sediment sample (12 MB FORC file, measured on an AGM). No postediting of the graphics was done other than adding the SF to the FORCinel plots. Results of two different SF are shown.

a very particular type of smoothing kernel Heslop and Muxworthy (2005). A straightforward generalization of the smoothing problem is therefore the use of any generic smoothing kernel that yields a good smoothed result (in our case a Gaussian kernel). Since a polynomial surface fit smoothing effectively imposes the assumption of the FORC distribution locally being a second-order polynomial—an assumption that may often not hold, in which case features are "smoothed out" not because of their noisiness but because of them not being a (2-D) parabola. In the Fourier space, however, it is straightforward to implement any smoothing kernel that is appropriate for the data—in other words, the smoothing kernel can in principle be chosen proportional to the autocovariance function of the original signal.

Table 1
Comparison of processing times between FORCinel (Harrison and Feinberg, 2008) and Forcot

Sample	Instrument	File size (kB)	FORCinel				Forcot	
			Load (s)	SF = 3 (s)	SF = 5 (s)	SF = 7 (s)	SF = auto (s)	SF = auto SF
Tiva Canyon Tuff	VSM	285	0.9	1.0	2.4		0.5	1.4
Iceland Basalt S05	VSM	1184	2.2	2.4	8.3		1.3	2
Iceland Basalt S23	VSM	2620	3.2	4.8	19.1		2.2	2.2
Marine Sediment 1	VSM	3577	4	6	24	70	2.7	2.2
Marine Sediment 2	AGM	6086	6	11	42	116	4.0	2.6
Marine Sediment 3	AGM	12458	13	22	83	238	6.6*	2.2

Note. The LOESS algorithm employed by FORCinel requires longer computation times for higher smoothing factors (SF), hence different SFs were tested. The fast Fourier transform (FFT) method used by Forcot requires the same computation time for any SF; hence, only the automatically selected SF from the noise power spectrum is reported. Additionally, the times for loading files are reported for FORCinel; these times are already included in the Forcot times, since no separate manual steps for loading and processing are required. Nevertheless, the longest time, marked with a "*" consisted of ~2.6 s for file handling + 1.2 s for processing + 2.8 s for plotting.

5. Conclusions

We have presented both a new algorithm that significantly reduces the time for processing FORC diagrams using FFT, and a new software, Forcot, that implements this algorithm, provides a significantly improved user experience, and reduces the number of manual steps needed to create print quality FORC diagrams. The program has been shown to yield diagrams of similar, if not better, quality as previous FORC processing software, while reducing the processing time by a factor between 2 and 50. The strength of the new software is clearly on basic processing of large and/or many of FORC diagrams, while not offering the same level of fine tuning offered by other FORC softwares. Some limitations of the FFT algorithm include the following: The requirement of regular FORC grids, the simplistic extrapolation toward the first point $H_c = 0$, and the use of a Gaussian weight function (which assigns a nonzero weight to all points in the diagram; although theoretically the FFT algorithm can be implemented with any smoothing kernel). We argue that the better speed and user experience may justify the relative lack of fine tuning for all but the most difficult-to-process FORC diagrams.

Looking into the future, processing and/or analyzing FORC diagrams in Fourier space, as opposed to (H_a, H_b) space may have advantages for further new applications such as FORC-FFT-PCA (Lascu et al., 2015), or quantitative analysis of time-asymmetric (TA) FORC diagrams (Berndt et al., 2018): In FFT space, the position of a FORC distribution peak is represented by phase angles, its intensity by magnitudes. For FORC-PCA, for example, end-members with variable coercivity-peaks could be detected, for example, a biogenic end-member with a characteristic central-ridge signature and a detrital end-member with a characteristic vortex/multi-domain signature, even when both components show some variations in coercivity throughout a core. Similarly, TA-FORCs could be decomposed into their constituent parts using PCA in Fourier space, since TA-FORCs show vertical shifts in their peaks in response to thermal fluctuations—these shifts would be represented by changing phase angles with unchanged frequencies in Fourier space and could therefore be decomposed using PCA. The here presented FFT representation of FORC diagrams may therefore lead to completely new ways of FORC analysis.

Acknowledgments
T. A. B. is supported by a Boya postdoctoral fellowship funded by Peking University. L. C. thanks a fellowship support from the National Natural Science Foundation of China (grant 41722402). We would like to thank Ramon Egli and Richard Harrison for a very engaged review and Joshua Feinberg for his editorial handling. *Forcot* is freely available on https://thomasberndt.github.io/Forcot; its MATLAB source code is available on https://github.com/thomasberndt; both are also available on 4TU.ResearchData (10.4121/uuid:3964b8d9-8e5d-4c6e-849f-b6bc10834bab).

References

Berndt, T. A., Chang, L., Wang, S., & Badejo, S. (2018). Time-Asymmetric FORC Diagrams: A New Protocol for Visualizing Thermal Fluctuations and Distinguishing Magnetic Mineral Mixtures. *Geochemistry, Geophysics, Geosystems*, *19*, 1–28. https://doi.org/10.1029/2018GC007669

Berndt, T. A., & Muxworthy, A. R. (2017). Dating Icelandic glacial floods using a new viscous remanent magnetization protocol. *Geology*, *45*(4), 339–342. https://doi.org/10.1130/G38600.1

Chang, L., Harrison, R. J., Zeng, F., Berndt, T. A., Roberts, A. P., Heslop, D., & Zhao, X. (2018). Coupled microbial bloom and oxygenation decline recorded by magnetofossils during the Palaeocene-Eocene Thermal Maximum. *Nature Communications*, *9*(4007), 1–9. https://doi.org/10.1038/s41467-018-06472-y

Chang, L., Roberts, A. P., Heslop, D., Hayashida, A., Li, J., Zhao, X., et al. (2016). Widespread occurrence of silicate-hosted magnetic mineral inclusions in marine sediments and their contribution to paleomagnetic recording. *Journal of Geophysical Research: Solid Earth*, *121*, 8415–8431. https://doi.org/10.1002/2016JB013109

Chang, L., Roberts, A. P., Winklhofer, M., Heslop, D., Dekkers, M. J., Krijgsman, W., et al. (2014). Magnetic detection and characterization of biogenic magnetic minerals: A comparison of ferromagnetic resonance and first-order reversal curve diagrams. *Journal of Geophysical Research: Solid Earth*, *119*, 6136–6158. https://doi.org/10.1002/2014JB011213

Channell, J. E. T., Harrison, R. J., Lascu, I., McCave, I. N., Hibbert, F. D., & Austin, W. E. N. (2016). Magnetic record of deglaciation using FORC-PCA, sortable-silt grain size, and magnetic excursion at 26 ka, from the Rockall Trough (NE Atlantic). *Geochemistry, Geophysics, Geosystems*, *17*, 1823–1841. https://doi.org/10.1002/2016GC006300

Egli, R. (2013). VARIFORC: An optimized protocol for calculating non-regular first-order reversal curve (FORC) diagrams. *Global and Planetary Change*, *110*, 302–320. https://doi.org/10.1016/j.gloplacha.2013.08.003

Egli, R., Chen, A. P., Winklhofer, M., Kodama, K. P., & Horng, C.-S. (2010). Detection of noninteracting single domain particles using first-order reversal curve diagrams. *Geochemistry, Geophysics, Geosystems*, *11*, Q01Z11. https://doi.org/10.1029/2009GC002916

Harrison, R. J., & Feinberg, J. M. (2008). FORCinel: An improved algorithm for calculating first-order reversal curve distributions using locally weighted regression smoothing. *Geochemistry, Geophysics, Geosystems*, *9*, Q05016. https://doi.org/10.1029/2008GC001987

Harrison, R. J., Muraszko, J., Heslop, D., Lascu, I., Muxworthy, A. R., & Roberts, A. P. (2018). An Improved Algorithm for Unmixing First-Order Reversal Curve Diagrams Using Principal Component Analysis. *Geochemistry, Geophysics, Geosystems*, *19*, 1595–1610. https://doi.org/10.1029/2018GC007511

Heslop, D., & Muxworthy, A. R. (2005). Aspects of calculating first-order reversal curve distributions. *Journal of Magnetism and Magnetic Materials*, *288*, 155–167. https://doi.org/10.1016/j.jmmm.2004.09.002

Heslop, D., & Roberts, A. P. (2012). Estimation of significance levels and confidence intervals for first-order reversal curve distributions. *Geochemistry, Geophysics, Geosystems*, *13*, Q12Z40. https://doi.org/10.1029/2012GC004115

Lascu, I., Harrison, R. J., Li, Y., Muraszko, J., Channell, J. E. T., Piotrowski, A. M., & Hodell, D. A. (2015). Magnetic unmixing of first-order reversal curve diagrams using principal component analysis. *Geochemistry, Geophysics, Geosystems*, *16*, 2900–2915. https://doi.org/10.1002/2015GC005909

Muxworthy, A. R., King, J. G., & Heslop, D. (2005). Assessing the ability of first-order reversal curve (FORC) diagrams to unravel complex magnetic signals. *Journal of Geophysical Research, B: Solid Earth*, *110*, B01105. https://doi.org/10.1029/2004JB003195

Pike, C. R., Roberts, A. P., & Verosub, K. L. (1999). Characterizing interactions in fine magnetic particle systems using first order reversal curves. *Journal of Applied Physics*, *85*(9), 6660–6667. https://doi.org/10.1063/1.370176

Roberts, A. P., Heslop, D., Zhao, X., & Pike, C. R. (2014). Understanding fine magnetic particle systems through use of first-order reversal curve diagrams. *Reviews of Geophysics*, *52*, 557–602. https://doi.org/10.1002/2014RG000462

Roberts, A. P., Liu, Q., Rowan, C. J., Chang, L., Carvallo, C., Torrent, J., & Horng, C.-S. (2006). Characterization of hematite (a-Fe2O3), goethite (a-FeOOH), greigite (Fe3S4), and pyrrhotite (Fe7S8) using first-order reversal curve diagrams. *Journal of Geophysical Research*, *111*, 1–16. https://doi.org/10.1029/2006JB004715

Roberts, A. P., Pike, C. R., & Verosub, K. L. (2000). First-order reversal curve diagrams: A new tool for characterizing the magnetic properties of natural samples. *Journal of Geophysical Research*, *105*, 461–428. https://doi.org/10.1029/2000JB900326

Roberts, A. P., Zhao, X., Harrison, R. J., Heslop, D., Muxworthy, A. R., Rowan, C. J., et al. (2018). Signatures of Reductive Magnetic Mineral Diagenesis From Unmixing of First-Order Reversal Curves. *Journal of Geophysical Research*, *123*, 4500–4522. https://doi.org/10.1029/2018JB015706

Savitzky, A., & Golay, M. J. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, *36*(8), 1627–1639. https://doi.org/10.1021/ac60214a047

Till, J. L., Jackson, M., Rosenbaum, J. G., & Solheid, P. (2011). Magnetic properties in an ash flow tuff with continuous grain size variation: A natural reference for magnetic particle granulometry. *Geochemistry, Geophysics, Geosystems*, *12*, Q07Z26. https://doi.org/10.1029/2011GC003648

Yamazaki, T. (2009). Environmental magnetism of Pleistocene sediments in the North Pacific and Ontong-Java Plateau: Temporal variations of detrital and biogenic components. *Geochemistry, Geophysics, Geosystems*, *10*, Q07Z04. https://doi.org/10.1029/2009GC002413

Zhao, X., Heslop, D., & Roberts, A. P. (2015). A protocol for variable-resolution first-order reversal curve measurements. *Geochemistry, Geophysics, Geosystems*, *16*, 1364–1377. https://doi.org/10.1002/2014GC005680