# Akvo: a Surface NMR Workbench

Trevor Irons, Benjamin Bloss, Thomas Kremer, Andy Kass

# Akvo: a Surface NMR Workbench

Trevor Irons
*Department of Geophysics*
*Montana Technical University*
*Butte MT, USA*
tirons@mtech.edu
Bio ▸
Benjamin R. Bloss
*United States Geological Survey, Geology, Geophysics, and Geochemistry Science Center, Denver CO, USA*
bbloss@usgs.gov
Bio ▸
Thomas Kremer
*University of Nantes, Univ Angers, CNRS, LPG, UMR 6112, F-44000 Nantes, France*
Thomas.Kremer@univ-nantes.fr
Bio ▸
M. Andy Kass
*Hydrogeophysics Group, Dept. of Geoscience, Aarhus University, Denmark*
andy.kass@geo.au.dk
Bio ▸

## Abstract

Akvo is a software project for processing, modelling, and inversion of surface nuclear magnetic resonance (sNMR) data which is being released as a resource for the community. The signal processing, inversion, front-end, and Qt5-based graphical user interface code are written in Python 3 and released under the GNU v3.0 public license. Akvo seamlessly interfaces with Merlin, an open source (Mozilla Public License) sNMR modelling API written in C++-14 for high-performance back-end functionality. Precompiled binaries and Python wrappers for Merlin are provided through pip as part of the Akvo package, allowing for simplified installation on Linux, macOS, and Windows 10 platforms. The installer also provides a standalone executable entry point, meaning that users can bypass direct interaction with the Python interpreter, which can be helpful for newcomers to Python. Akvo is entirely open source. There are no external dependencies on proprietary runtime environments. The code is managed in a public git repository (https://akvo.lemmasoftware.org). Anyone can anonymously access, read, edit, and use the code. Akvo is flexible—Akvo processing, modelling, and inversion supports multiple channel data and arbitrary transmitter configuration. All processing steps are configurable so that they can be adapted to varying survey conditions. Akvo processing is reproducible and transparent. The processing workflow is documented along with the data in self-describing human readable YAML files. This encourages reproducible publications and reports. Akvo seeks community engagement. We are looking for new users and collaborators. This runs the gamut from code development, to testing and reporting of tickets for enhancements or bugs reports, and documentation.

Keywords: hydrogeophysics, Python, open source, surface NMR

2010 MSC: 00-01, 99-00

## Introduction

Free/libre and open source software (FLOSS) projects serve a vital role in the geoscience community where they facilitate the exchange of ideas and allow for greater scrutiny when validating research findings. However, in many sub-fields non-free and proprietary software remain the only viable options for processing and inversion of data. This is fundamentally damaging as findings cannot be freely replicated and reproduced without the expenditure of great effort. Building the software infrastructure to facilitate the dissemination of research findings in a verifiable and reproducible manner is therefore of paramount concern. The Python programming language continues to be widely adopted by the scientific community, and usage statistics suggest that it is the language of choice for a large portion of open source projects (Borges et al., 2016). Additionally, software must be easy to use for a variety of users, not just experts, in order to gain a sufficiently large userbase.

Surface nuclear magnetic resonance (sNMR), also called magnetic resonance sounding (MRS), is a non-invasive geophysical method that is directly sensitive to liquid-phase water in the top≈100 m of the subsurface (e.g. Behroozmand et al., 2015). Processing and inversion of these data are typically performed using proprietary binary codes provided by instrument manufacturers and a few dominant research institutions which tend to not widely disseminate source codes. As such, newcomers to the field face a large barrier to entry. For a fledgling technique such as sNMR this limits the widespread adoption of the method and may even drive off interested parties. There are also no established standards for releasing sNMR data associated with publications which further fractionates the community.

We are aware of only three other software packages providing sNMR capabilities with freely available source code.

- **MRSmatlab**[1] is self-described as being an 'open platform' package which provides much of the same functionality as Akvo including signal processing as well as 1D modelling and inversion (Müller-Petke et al., 2016). While a useful tool, MRSmatlab has several limitations: 1) the licensing is

*Corresponding author*
*Email address: tirons@mtech.edu (Trevor Irons)*
[1]*available at https://github.com/mmpetke/MRSmatlab*

ambiguous and it is unclear what the terms of use are, 2) survey geometry is limited, 3) it is unclear how potential contributors should become involved, 4) generated files are not human-readable, and 5) it has proprietary non-free dependencies (Matlab / Matlab runtime environment). Both items 1) and 5) preclude MRSmatlab from being classified as FLOSS.

- **Comet** is a FLOSS Python project providing advanced (2D/3D) sNMR modelling and inversion capabilities with limited data processing. Comet also has fairly complex dependencies which may be difficult for some users to install and configure. Comet also lacks a graphical user interface which may be off-putting for some users.
- **Pygimli**[2] is a large FLOSS Python project providing a multi-method library for geophysical modelling and inversion (Rücker et al., 2017). At this date, Pygimli does propose sNMR inversion and modelling capabilities, but without any possibility of data processing. The data reading is also limited to the MRSmatlab format, hence constraining the use of Pygimli for sNMR inversion to the same limitations as MRSmatlab.

For these reasons, we feel that the sNMR community will benefit from a free and complete software project which is approachable for all users. Akvo[3] is a FLOSS project which aims to meet the needs of the sNMR community for a software tool which anyone can freely use and which facilitates reproducible research (section 2) and community engagement. Towards this end we are developing a tool which:

- Is freely available for anonymous download on the internet using well known software management and collaboration tools and is unambiguously released under a well-known copyleft software license (GPL V. 3).
- Has no proprietary dependencies.
- Provides a simplified and streamlined installation process and non-programmer accessible user interface which is extensible for the needs of advanced users.
- Is flexible and allows end-users to design and specify runtime parameters.
- Documents processing workflows and can generate publication quality figures, both of which are necessary for reproducible manuscripts.

Akvo is written in Python 3 but interfaces with the electromagnetic modelling library Lemma including the sNMR module Merlin[4] (Irons et al., 2012a). This interface is achieved via pybind11 python bindings (Jakob et al., 2017) and Akvo users do not need to build, manually install, or compile these dependencies. Akvo features an object oriented back end which performs the processing and logic portions of the code. This back end can be used directly using Python scripts. However, for most users, a graphical user interface (GUI) front end written in Qt (using PyQt V. 5 Python bindings) presents a user friendly means by which to interact with the software.

*[2] available at https://www.pygimli.org/*
*[3] Akvo is an Esperanto word meaning water.*
*[4] both are FLOSS written in C++14 and released under the Mozilla Public Licence and available at https://lemmasoftware.org*

Akvo currently provides data processing, 1D modeling, and 1D inversion capabilities. At the time of writing only Vista Clara GMR data are supported (Walsh, 2008). Akvo can read FID, pcPSR/T1 (Walbrecker et al., 2011) and adiabatic pulse (Grombacher, 2018) data in either the older ASCII formats or newer binary (.lvm) formats. Spin echo and Hahn echo datasets cannot be processed in Akvo at this time. Akvo only provides a 1D QT style (Müeller-Petke and Yaramanci, 2010) deterministic inversion at the time of writing although additional algorithms are planned for future releases.

## Reproducibility

One of the primary motivations for Akvo is the generation of fully-documented processing and inversion workflows which, combined with open-access data, allow for reproducible manuscripts (such as this one) and data releases. This is a pressing need of the community as an increasing number of journals are encouraging or mandating data be released upon publication.

Currently, there exists no consensus within the sNMR community regarding release of data for publication. A large percentage of publications do not include any form of public release of associated data. Most of these ‘closed’ publications generally include a methods section outlining the various preprocessing workflow steps and associated citations. Such an approach results in publications which are not verifiable or reproducible by the community. To address this shortcoming, release of data is becoming more common, but best practices for release have yet to be established. One approach has been to release fully preprocessed data (e.g. Kass et al., 2014; Keating et al., 2018). This has the benefit that readers can then work directly with the data in the same form as the authors. However, this form of data release prohibits readers from revisiting published data with improved processing algorithms. Additionally, authors tend to release these data *ad hoc* and no commonality exists in file formats across publications. As such, it may be difficult for readers to work with data released in this manner. At the opposite end of the spectrum, raw data has been released in a few instances (e.g. Davis et al., 2019). This approach has the benefit of providing data that can be manipulated at will by readers. The shortcoming exists in terms of repeating the authors processing which may not be fully documented. It may often not be possible to fully-replicate the results of the authors in such cases, especially when proprietary software is also being used. We propose an approach which combines these data release mechanisms in a standardised way which allows for full transparency, reproducibility, and flexibility.

Akvo, as well as Lemma and Merlin, relies heavily on self-describing yaml-formatted human readable file formats to achieve reproducibility. All Lemma and Merlin objects can be trivially serialized using stdout or print statements. Within Akvo, all processing workflow steps are preserved and documented throughout the processing workflow and in the final processed data. The format is flexible enough to allow for additional metadata such as coil loop configurations and geometry, UTM coordinates of the survey, weather conditions, magnetic field measurements, and arbitrary field notes to be included as well.

# Installation and use

Akvo is intended to be simple to install and use for anyone. Akvo releases are pushed to PyPi for streamlined distribution (https://pypi.org/project/Akvo/). For the majority of users, installation is trivial and the only prerequisite is a Python 3.7 installation. Developers wishing to work with the code will require a slightly more involved setup and non-standard installations will require compiling dependencies.

## Users on macOS, Window 10, and Linux

For most users, installation of Akvo and all its dependencies is trivial. If Python 3 is already installed this can be achieved through a simple pip command.

*Listing 1: Script for installing and running Akvo on an x86 64 processor running Windows 10, macOS or Linux from a Terminal (command prompt). Python 3.7 is currently a prerequisite to use pip.*

```
pip install akvo
akvo
```

Currently pip can only install Akvo on CPython 3.7. This is due to the fact that precompiled Lemma and Merlin wheels are only provided for this Python version; ostensibly this is because the developers of those projects are lazy. If the version of Python on your system is older than this, the Anaconda Python distribution (https://www.anaconda. com/) provides a convenient means by which to install multiple versions of python in independent 'environments'[5].

If you are using an Anaconda installation and are seeing ssl errors it may be helpful to set

**conda config --set ssl_verify no**

**pip install --trusted-host pypi.org **

**--trusted-host files.pythonhosted.org akvo**

## Developers

Users interested in developing or extending Akvo will need direct access to the source code which can be cloned from the git repository https://git.lemmasoftware.org/akvo.git (mirrored at https://github. com/LemmaSoftware/akvo.git). More details can be found at the project website https://akvo.lemmasoftware. org/. Installation from the source is achieved using the standard commands to build and install (Listing 2).

*Listing 2: Script for installing and running Akvo from the git repository on an x86 64 processor running Windows 10, macOS or Linux from a Terminal (command prompt). Both git and Python 3.7 are prerequisites.*

[5]*While Anaconda Python can be used to run Akvo, it is not a requirement or necessary, and even from within Anaconda, pip, not conda, is used to install Akvo.*

```
git clone \
https://git.lemmasoftware.org/akvo.git
cd akvo
python setup.py build
python setup.py install
akvo
```

## Installation on non-standard systems

Akvo is tested on x86 64 systems running Linux, macOS, and Windows10. If you need to install Akvo on a more exotic platform such as BSD, a Raspberry Pi, or your kerosene-powered cheese grater it will be necessary to compile Lemma and Merlin on that platform, details and source code are available at https://lemmasoftware.org.

## Starting the program

Upon successful installation using either of the above approaches, the Akvo GUI can be started from the command line by simply typing 'akvo'. Alternatively (and equivalently), the GUI can be launched by navigating to 'akvo/gui' in the source tree and typing python akvoGUI.py; but most users will not need to interact with the python interpreter at all.

The Akvo GUI is divided into three workflow modes: Preprocessing, Modelling, and Inversion which can be toggled in the main menubar. Preprocessing includes functionality for signal conditioning, filtering, denoising, and stacking of field data. The Modelling workflow includes kernel generation and forward modelling. Finally, inversions and model appraisal are handled in the Inversion workflow. In all three workflows the interface includes a large Matplotlib (Hunter, 2007) driven plotting window on the left side of the main window with a collapsible informational area below this reporting data attributes. Plots can be manipulated using the controls directly beneath the plotting window. The right side of the GUI contains tabs which change according to the workflow that is selected. The GUI window can be resized and design choices were made to enable use on rugged field laptops which often have low pixel counts. Most of the entry items in the GUI have explanation bubbles which become visible when you hover a mouse over them.

## Demonstration

The remainder of this paper presents a tutorial style description of program usage. Readers are encouraged to follow along and experiment with various program options. We will be reprocessing a legacy dataset collected in 2010 using an early version of the Vista Clara GMR. These data have previously been presented (Site 58A in Figs. 22, 24 in Irons et al., 2012b) and the raw dataset can be downloaded at a Zenodo archive (Irons, 2019). A 100 m square sided transmitter loop was employed on channel 1 in the survey with a 40 ms FID pulse. Channels 2 and 3 were noise reference loops, their sizes and geometry have seemingly been lost from the annals of time. The field site has been well studied making this a good example dataset to work with (Knight et al., 2012; Müller-Petke et al., 2013; Irons and Li, 2014).

## Preprocessing workflow

Preprocessing, for lack of a better term, includes all work done to a field dataset before inversion. This includes reading of raw data files, designing and applying filters, noise rejection, stacking, and statistical characterisation. Five tabs are available in this workflow: Load, Noise removal, QC, META, and Log. Initially the Preprocessing tabs are all disabled as no data have been loaded yet, which is the natural first step. Partially preprocessed datasets within Akvo can be saved from the Save processing under file in the main Akvo menubar. Similarly, these can be loaded using Open Akvo Preprocessed dataset in the file menu. Note that many of the various preprocessing steps can be performed independently, repeatedly, and in any order. In the authors experience, following the 'natural' order from the GUI structure generally gives the best results.

## Load tab

To load the raw GMR data downloaded from Zenodo, select Open GMR Header under file in the menubar. This will open a standard file dialog box. Select the (extensionless) GMR header file associated with the dataset of interest, in this case 29March100mTxCh1_40ms_FID 40ms FID (Figure 1). Upon reading the header file associated with the NMR raw data set, survey parameters such as tuning capacitance, signal sampling frequency, number of pulse moments and pulse type, frequency and length are directly extracted and displayed. Each of the boxes within this tab are discussed below. For the remainder of this paper the 29March100mTxCh1_40ms_FID dataset will be referred to as the 58A dataset.

**Input parameters.** In this box, the user specifies the number of

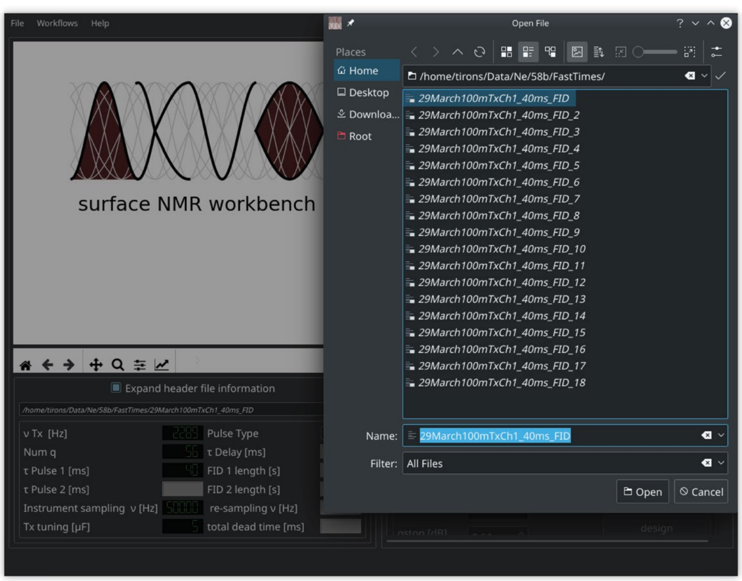



*Figure 1: Screen capture (Linux KDE/plasma) of file dialog box to open a GMR header file.*

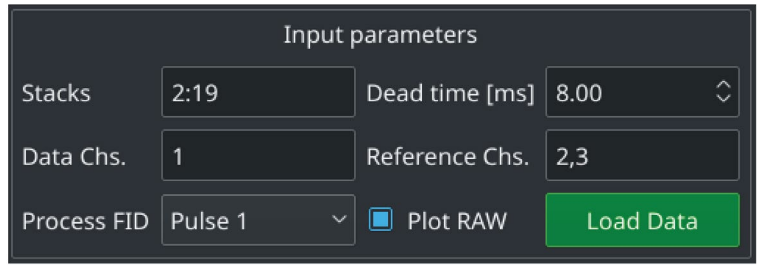



*Figure 2: The input parameters box (Linux KDE/plasma) found within the Load tab is used to import the example raw dataset. Note the increased dead time from default. Stacks 2-18 will be loaded following python array indexing convention.*

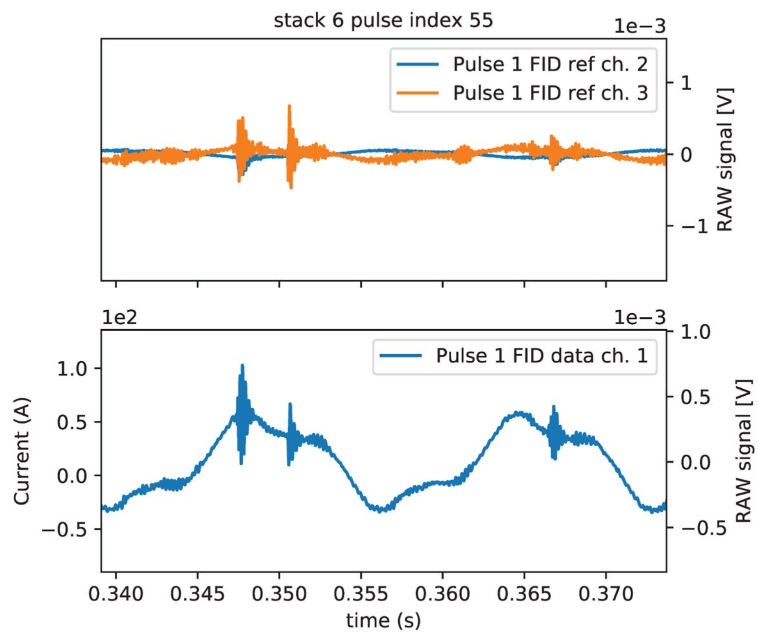



*Figure 3: Example of moderate amplitude spikes in the raw import of the Site 58A data prior to any processing in Akvo. Typical sNMR spikes manifest not as singular events, but rather packets of high intensity variation. Ordinate label on bottom left pertains to transmitter current, which is not visible at this level of zoom.*

measurement stacks to be loaded and indicates which channels are used to record the NMR signal and which are used as reference loops for subsequent reference noise cancellation. The dead-time value is set by default to 5.5 ms. Although it can be adjusted manually, for newer datasets this is generally a safe value. In the case of the dataset we are working with, it is recommended to increase the dead time to 8 ms, all of the parameters we used are shown in Figure 2. Plotting the raw data is optional, and omitting the plotting speeds up the import process.

**Despiking.** Users coming from other processing packages will notice that Akvo does not include an explicit despiking filter. This is not because we are not concerned with spikes and in fact several despike filters have been tested in Akvo in the past, but to date we have not been satisfied with their performance which tends to only obfuscate the underlying loss of information associated with the presence of a spike. The ideal time to apply a despike filter is directly after importing the data before any other filtering takes place, as spikes will induce either spectral leakage or an impulse response of filters. However, despiking appropriately early in the processing flow is difficult in sNMR, and despiking later in the processing flow has minimal benefit as the 'damage has already been done'.

To illustrate the challenge we will consider the example dataset. In our experience the majority of spikes in the raw sNMR signal tend to not be singular points, but rather packets of high intensity which may persist for several ms (Figure 3). The origin of these packets is somewhat debated and they have variously been suggested to be system responses to any combination of sources such as electric fences, changing loads on the local powergrid, or magnetotelluric plane waves. Automated identification of these events is relatively straightforward, however, the question of replacement values remains vexing due to the very low signal-to-noise-ratio of sNMR. Widely adopted despiking filters include a cascade of medians or interpolation. In either of these cases the median or interpolated values will be dominated by the slowly-varying noise and will not provide any kind of reasonable approximation of the quickly-changing NMR signal of interest. Model based subtraction of spikes by other groups has to date been largely unsuccessful (Larsen, 2016). Costabel and Müller-Petke (2014) suggest that wavelet decomposition could be a promising approach, but we have not had the opportunity to implement this in Akvo. Simply masking these points is also not a viable option as many of the processing steps rely on regularly sampled data. For this reason, we find that the application of a despike filter can actually degrade the final data as those values may be more difficult to identify as outliers while stacking. Akvo employs a statistically informed stack which rejects spike outliers as long as a statistically significant number of repeated measurements are made.

**Downsampling and Truncation.** The first step of preprocessing data with Akvo consists of down-sampling the signals and performing truncation if desired. Although this step in not mandatory in the Akvo processing workflow, it is highly recommended to avoid unnecessarily long processing times due to significant oversampling of the recorded signal by the GMR. Additionally, downsampling serves as an anti-alias filter. The GMR raw data initial sampling frequency is 50 kHz; therefore, the downsample factor of 5 in Akvo is usually a good value since the resulting Nyquist frequency is 5 kHz, which will always be above the NMR signal frequency obtained in Earth's field NMR. The downsampling includes an internal window based anti-alias filter prior to decimation. If desired, truncation can also be applied at the end of all the time series. However, care should be taken when applying truncation since it can lead to a loss of useful information, and we generally discourage its use. In the case of the example dataset we used the default values of 5 for downsampling and 0 for truncation.

**FD window Filter.** Frequency domain window filters are zero phase and simple to implement. However, they can cause large amounts of amplitude distortion. Within Akvo, various window filters can be designed with different shapes including rectangular, Hamming, and Hanning. The step response is also plotted which is used to calculated the increased dead time that will be applied by using this filter. For the example dataset we applied a 600 Hz Hamming filter. This box includes an option to 'trim dead time'. If a cascade of window and bandpass filter are used this can be unchecked. For datasets with large

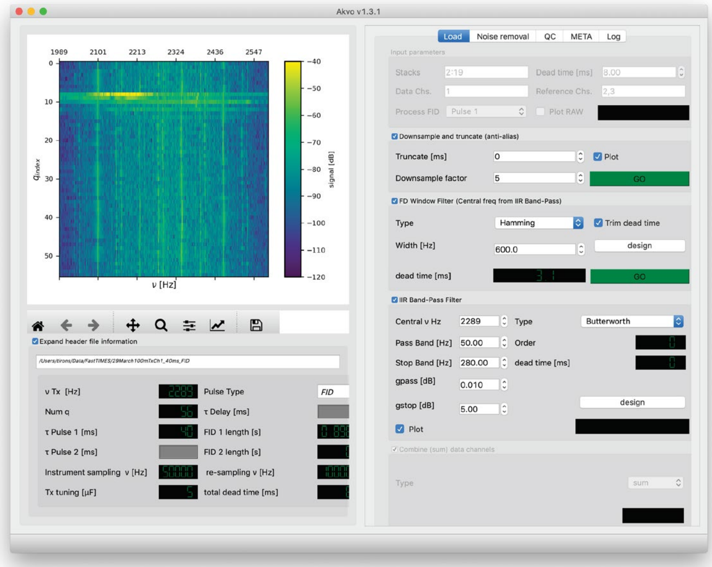



*Figure 4: Example of moderate amplitude spikes in the raw import of the Site 58A data prior to any processing in Akvo. Typical sNMR spikes manifest not as singular events, but rather packets of high intensity variation. Ordinate label on bottom left pertains to transmitter current, which is not visible at this level of zoom.*

spikes, spectral leakage can be imposed using a frequency domain window filter due to the inherent use of forward and inverse discrete Fourier transforms. This is the only frequency domain processing step in Akvo, and it can be avoided entirely if desired. A screen capture of all entered parameters in the Load tab is shown in Figure 4.

**IIR Band-Pass Filtering.** Time domain infinite impulse response (IIR) bandpass filtering can be applied either in tandem with a frequency domain window or as an alternative. In some cases a cascade of a wide frequency-domain window filter and a narrower time-domain bandpass filter has been an effective choice. When designing filters, Akvo users can choose between Butterworth (maximally flat in the passband), Chebyshev Type II, and elliptic. Since IIR filters are not zero-phase, Akvo runs these in both directions in order to achieve an effectively zero-phase filter. In the case of our demonstration, we did not apply this filter, but readers are encouraged to explore this option as roughly equivalent results can be achieved using this in lieu of a FD window filter.

## *Noise removal tab*

Surface NMR is often noise limited, and it is not uncommon for unusable data to be collected due to high levels of environmental noise. Akvo includes three (one deprecated) algorithms for noise rejection: a model based harmonic removal approach, a frequency domain transfer function formulation, and a reference loop noise rejection algorithm. The model based method has the advantage that reference loops need not be deployed. This attribute can save data that might otherwise be unusable. However, in most cases we achieve superior results using the reference based approach.

**Model-based harmonic noise modelling and removal.** Model-based harmonic noise rejection algorithms have received a lot of recent attention in the sNMR community. The formulation proposed by Larsen et al. (2014) has been shown to be effective in a wide range of sNMR applications. Akvo takes a similar approach, but rather than splitting the problem into concatenated real and imaginary parts, an Euler relation is used to pose the problem in C[1]

$$\begin{pmatrix}\alpha_0\\ \alpha_1\\ \vdots\\ \alpha_L\end{pmatrix} = \begin{pmatrix} e^{Jk_0 2\pi \frac{f_0}{f_s} n_0} & e^{Jk_1 2\pi \frac{f_0}{f_s} n_0} & \cdots & e^{Jk_N 2\pi \frac{f_0}{f_s} n_0}\\ e^{Jk_0 2\pi \frac{f_0}{f_s} n_1} & e^{Jk_1 2\pi \frac{f_0}{f_s} n_1} & \cdots & e^{Jk_N 2\pi \frac{f_0}{f_s} n_1}\\ \vdots & \vdots & \ddots & \vdots\\ e^{Jk_0 2\pi \frac{f_0}{f_s} n_L} & e^{Jk_1 2\pi \frac{f_0}{f_s} n_L} & \cdots & e^{Jk_N 2\pi \frac{f_0}{f_s} n_L}\end{pmatrix}^{-1} \begin{pmatrix}s_0\\ s_1\\ \vdots\\ s_N\end{pmatrix}$$

$$h = 2|\alpha| \cos\left(2\pi k \frac{f_0}{f_s} i_t \angle\alpha\right)$$

$$\min |s - h|,$$

where *h* denotes the harmonic model, *s* the sNMR acquisition signal, and the minimisation problem is solved to find the optimal fundamental frequency value $f_0$ that best reduces the energy of the acquisition signal.

Akvo does not solve the minimisation problem using a brute force grid search, but rather uses a non-linear search strategy which we find is able to converge quickly and reliably. Akvo permits users to search for up to two fundamental frequencies and can model subharmonics. If noise sources are found to be varying faster than a typical sNMR record, Akvo can apply the model-based subtraction to a user-specified number of subsets of the record (Kremer et al., 2019). Both the data and reference channels can be processed for harmonic noise reduction. Application of harmonic noise cancellation to the 58A dataset demonstrates effective signal reduction (Figure 5). However, we find that for many datasets, including this one, harmonic noise cancellation is outperformed by reference based approaches. For the full processing of the 58A dataset

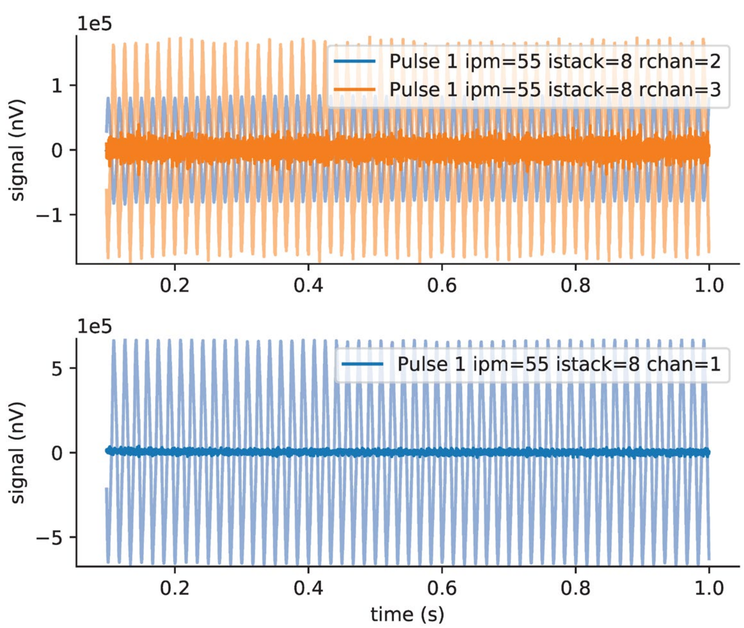



*Figure 5: Application of model-based harmonic noise subtraction on raw data in Akvo. The top panel shows the reference channels (Chs. 2 & 3) and the bottom shows the data channel (Ch. 1), light colours represent the signal before noise removal.*

we did not apply this step and Figure 5 is presented to show the utility of this tool. In circumstances where reference loops are not available or not effective, the availability of harmonic modelling can be invaluable.

**FD (static transfer function) Noise cancellation.** The development of multi-channel acquisition devices brought the possibility to deploy noise reference loops in addition to the transmitter/receiver loop. Provided that the noise in the reference loop is well correlated with the noise in the primary loop, the reference signal can be used to estimate and remove the noise in the primary loop. This general method is often referred to as reference noise cancellation (RNC), reference based filtering, adaptive noise cancellation, or active noise suppression.

There are several ways to apply this idea in the sNMR context. The Wiener filtering approach consists of computing the optimal filter (in a least-square sense) between the primary signal and the reference signal by solving the well-known Wiener-Hopf equations:

$$w = AC_{yy}^{-1}\, cc_{xy}$$

where *AC* is the auto-correlation matrix of the input signal *y* (the reference signal), and *cc* denotes the cross-correlation vector between the input signal *y* and the desired output signal *x* (the primary signal). This approach can give very satisfying results (e.g. Larsen et al., 2014) although varying degrees of efficiency can be reached depending on the noise characteristics and the optimal parameter settings associated. These aspects are well covered by (Müller-Petke and Costabel, 2014) who discuss in particular the cons and pros of computing the filter coefficients based on all the stacks (global approach) or computing them for each stack independently (local approach).

Akvo provides a frequency-domain Wiener filtering noise cancellation algorithm which computes the coefficients over all of the pulse moments within a single stack. The approach implemented in Akvo is admittedly rarely well-performing, as we do not allow for records with broadband spikes to be removed during the transfer function calculation. In almost all cases superior performance can be achieved in the time-domain (next paragraph) and as such use of this algorithm within Akvo is generally discouraged.

**Time-domain RLS Active Noise Suppression.** Another approach relates to the adaptive noise cancellation, initially proposed by Widrow et al. (1975). The adaptive aspect denotes to the fact that the filter coefficients are updated in time, depending on the coefficients values from previous records and on the evolution of the noise characteristics. The adaptive filtering domain is vast and numerous techniques can be adopted (Haykin, 2008), many of which have yet to be applied in the sNMR context. However, Dalgaard et al. (2012) conducted a comparative study between Wiener filtering and two types of adaptive filters and concluded that the three methods showed equivalent efficiency. The active noise suppression in Akvo is based on the recursive least-square (RLS) method, which was actually tested by Dalgaard et al. (2012). Akvo's implementation handles an arbitrary number of reference channels and proceeds as follows

$$y(k) = \sum_{j=0}^{M} \sum_{i=0}^{N} w_{i,j} \; x_{i,j}(k)$$
$$y(k) = \mathbf{x}^T(k) \; \mathbf{w}(k),$$

where the coefficients at time index $k$ are computed based on the coefficients at time $k - 1$

$$w\,(k) + \mathbf{R}(k)\,\mathbf{x}(k)\;(d(k) - y(k))$$

Where $d$ is the measured signal and $R$ is the inverse of the autocorrelation matrix which is computed

$$\mathbf{K}(0) = \frac{1}{\mu}\mathbf{I}$$
$$\mathbf{R}(k) = \frac{1}{\lambda}\left(\mathbf{R}(k-1) - \frac{\mathbf{R}(k-1)\mathbf{x}(k)\mathbf{x}(k)^T\mathbf{R}(k-1)}{\lambda + \mathbf{x}(k)^T\mathbf{R}(k-1)\mathbf{x}(k)}\right).$$

The parameter $\lambda$ is the forgetting factor ($0.95 \leq \lambda \leq 1$), which controls how quickly the filter changes, and $\mu$ is an initialisation factor providing regularisation as the filter is trained. One consequence of the filter is that it takes some time to 'wind up' and as a result the output is more favourable after several cycles of training. For this reason in Akvo the filter is run backwards from the end of the signal towards the beginning which maximises effectiveness at the beginning of the record. While filter coefficients are retained across records, it is still usually necessary to truncate some of the late time signal for which the filter has not yet been sufficiently trained. We processed the example dataset with 200 taps, no PCA[6], $\mu = 0.01$, $\lambda = 0.99$, and a truncation length of 800 ms (Figure 6).

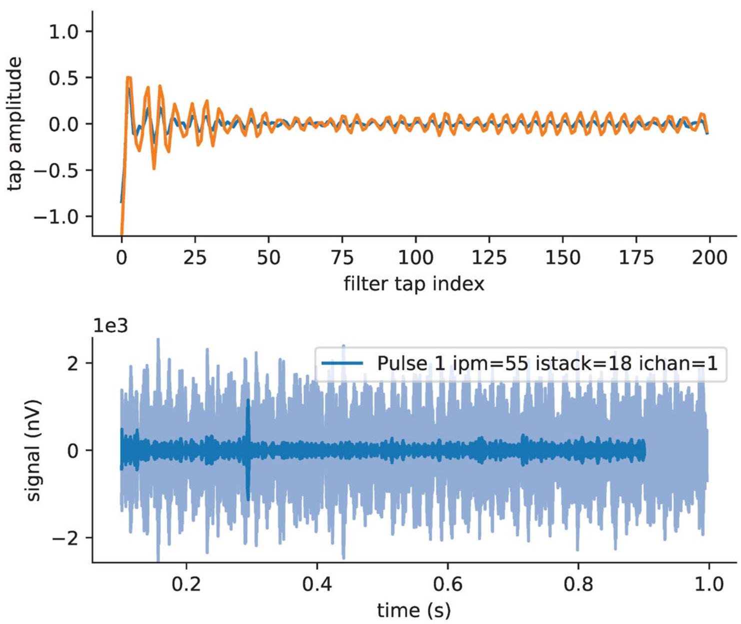



*Figure 6: Application of the RLS algorithm on the example dataset, In the bottom plot the light colour represents the noisy sample, and the darker colour is the filter output. The top plot contains the filter taps at the end of the process.*

[6]*Principle component analysis can be applied to the reference channels prior to noise removal*

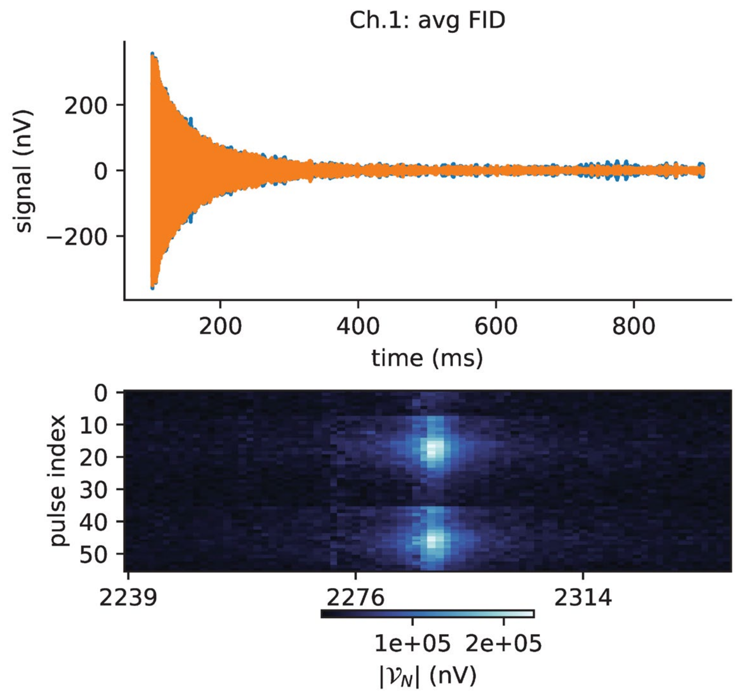



*Figure 7: Stack of the example dataset in Akvo. The top plot shows the average FID signal across all pulse moment, and the bottom plot shows the modulus of the 2D FFT of the entire dataset.*

## QC tab

The QC (quality control) tab contains methods for data aggregation and is the final digital signal processing step. The output of this step is a complete stacked and noise-characterised dataset that is ready for inversion.

**Pulse Moment Calculation.** Most sNMR instruments utilise capacitor banks to achieve high amplitude pulses. One consequence of this is that current amplitude tends to decrease during the pulse. Akvo calculates the effective pulse moment, for every pulse, using a Fourier based method to determine the total energy of the pulse at the Larmor frequency. This is a somewhat simplified approach, and some researchers in the community have been looking into a full Bloch equation tipping model. As such, this portion of the Akvo code may benefit from enhancement in the future. The Pulse Moment Calculation does not have any variables which need to be specified by users.

**TD SmartStack.** Averaging, or stacking, of sNMR data is simply the act of reducing the dimensionality of a sNMR dataset by combining repeat measurements. Akvo takes a group of similar amplitude pulse moments and stacks them in the time domain, identifying outliers based on the statistical distribution of all records at each time index. The median absolute deviation (MAD) of the population is calculated, and if a particular point lies above the user-specified threshold value, it is rejected. Akvo defaults to a 1.480 threshold value, which corresponds to a Gaussian noise assumption. While the example dataset is relatively clean, the benefits of this stacking approach clearly improve the sNMR signal in contrast to a traditional average (Figure 7).

**Gate integrate.** After demodulation through quadrature detection the data are oversampled. This oversampling will unnecessarily

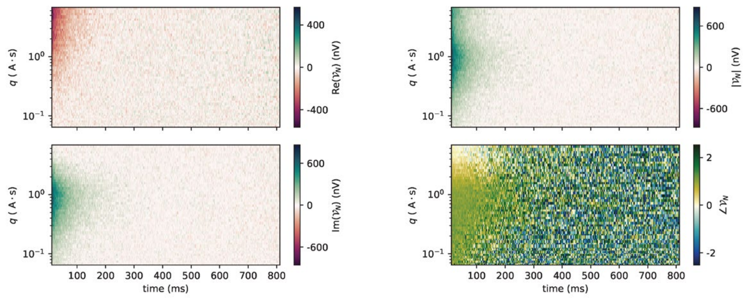



Figure 8: Output of the quadrature detection scheme are shown for the example dataset. In the left column, real (top) and imaginary (bottom) representations are shown. In the right column, the phased amplitude (top) and phase (bottom) are shown.

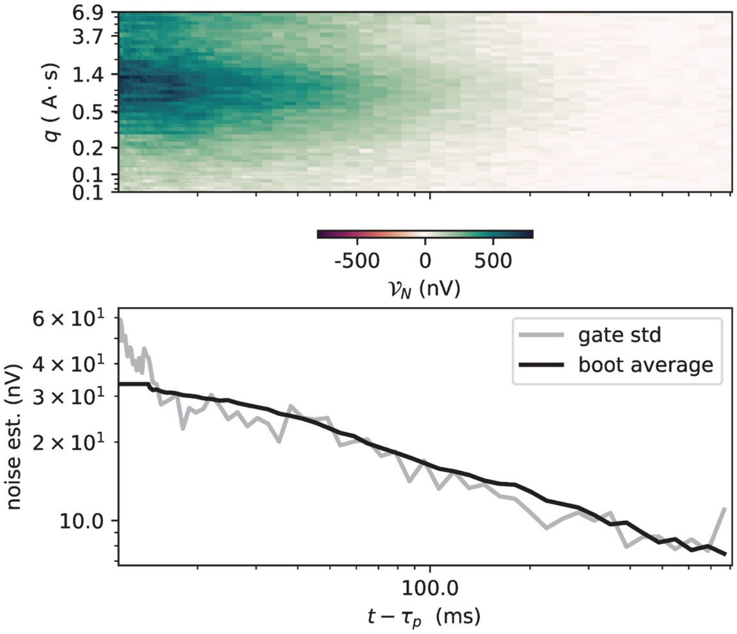



Figure 9: Output of the quadrature detection scheme are shown for the example dataset. In the left column, real (top) and imaginary (bottom) representations are shown. In the right column, the phased amplitude (top) and phase (bottom) are shown.

hinder the performance of inversion algorithms due to the very large dataspace. Options for reducing this include resampling, gate integration, or further dataspace transformation. Gate integration has the desirable characteristics that it operates as a sliding moving average low-pass filter with increasing order at later times when signal is lower but changes are slower. A consequence of gate integration is that noise levels are a function of gate width. Assumptions of ideal stacking rarely hold for sNMR, and as such Akvo uses a bootstrap block resampling approach to achieve a reliable noise estimate (Irons et al., 2018). Users may specify the number of gates per decade to calculate, however in our experience the default value of 20 gates is sufficient. In the Site 58A data we used 20 gates (Figure 9).

## *META tab*

The last tab in the preprocessing workflow allows users to enter pertinent survey information (Figure 10). This information includes, survey configuration and loop locations (Figure 11), as well as local measurements of the earth's magnetic field ($B_0$, inclination, and declination). This auxiliary information is written into the processed data file. By including this information in the dataset itself we have created a means of archival of all the information needed for kernel generation and inversion (Figure 10).

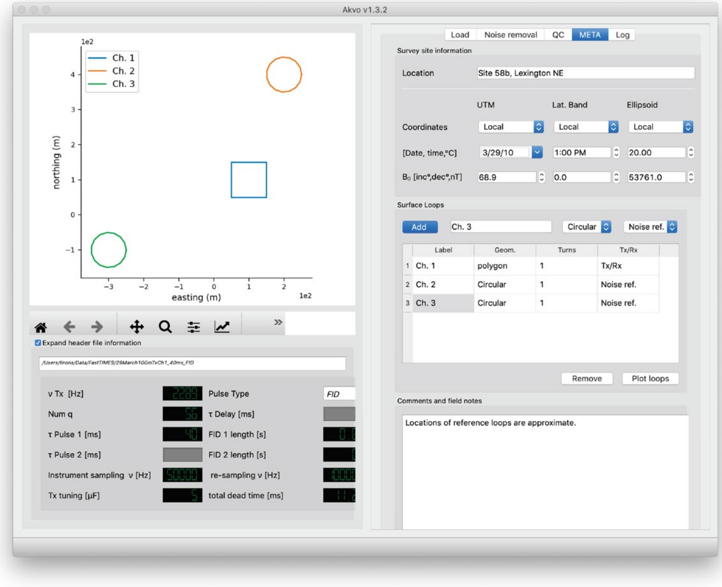



Figure 10: Meta tab shown with plot of loop locations on macOS.

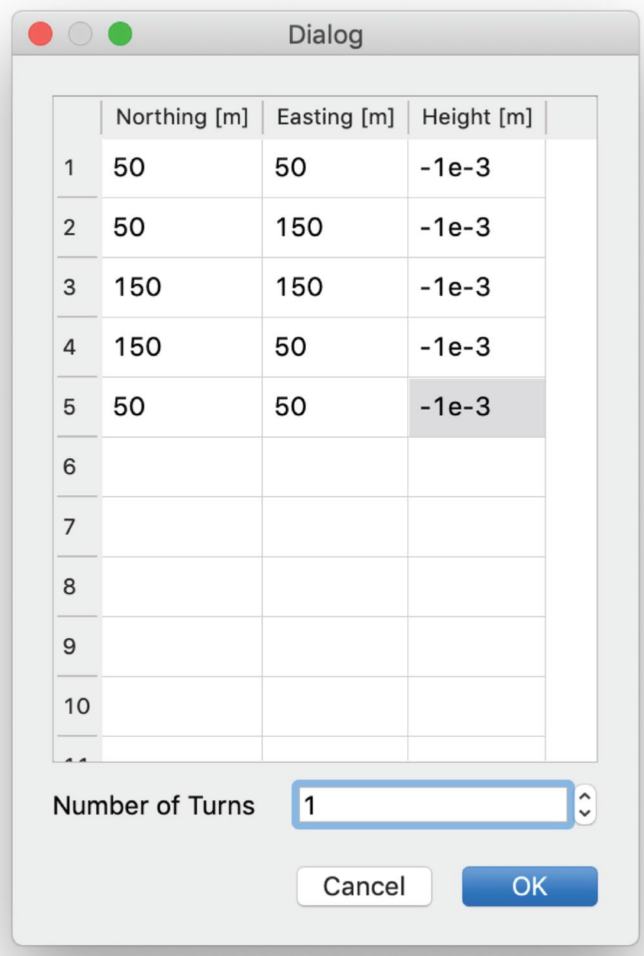



Figure 11: Dialog box for entering arbitrary polygonal loops. Note that the loop must be closed so that the first and last points are coincident. Additionally, the loop is given a slight vertical height above the ground which improves numerical stability.

*Listing 3: Output YAML datafile produced by Akvo. The actual gated data are omitted from the listing here, but the full file is available on Zenodo.*


```
!<AkvoData>
Akvo_VERSION: 1.4.3
TuneCapacitance: 5.0
headerstr: /home/tirons/Data/Ne/58b/FastTimes/29March100mTxCh1_40ms_FID
nPulseMoments: 56
processed: Akvo v1.4.3, on 29/10/2019
pulseLength: [0.04]
pulseType: FID
transFreq: 2289.0
Import:
  GMR Header: /home/tirons/Data/Ne/58b/FastTimes/29March100mTxCh1_40ms_FID
  opened: '2019-10-28T22:25:36.182739'
  pulse Type: FID
  stacks: [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
  data channels: [1]
  reference channels: [2, 3]
  pulse records: Pulse 1
  instrument dead time: 0.008
META:
  Location: Lexington, NE, Site 58A
  Coordinates: {UTM: Local, LatBand: Local, ellipsoid: Local}
  DateTime: 2010-03-29T12:00
  Temp: 20.0
  B_0: {inc: 68.0, dec: 0.0, intensity: 53761.0}
  Field Notes: 'Reference␣loop␣locations␣are␣approximate.␣'
  Loops: {Ch.1: Ch.1.yml, Ch.2: Ch.2.yml, Ch.3: Ch.3.yml}
Processing:
- {STEP 0: Resample, downsample factor: '5', truncate length: '0'}
- {STEP 1: Window filter, centre: '2289.0', trim: 'True', type: Hamming, width: '600.0'}
- {PCA: No, STEP 2: TD noise cancellation, lambda: 0.99, mu: 0.01, n_Taps: 200, truncate: 800.0}
Stacking:
  Calc Q: true
  TD stack: {cutoff: '1.48', outlier: MAD}
  Quadrature detection: {loss: linear, method: trf, trim: '2'}
  Gate integrate: {gpd: '20'}
```


## Data export

Once users have processed data to their satisfaction and entered all necessary META data, the data can be exported. This is achieved using the Export to Lemma item under File in main GUI menubar. This exports the data as a YAML file which can be inverted by Merlin or Akvo. A listing of the Site 58A processed file is included in Listing 3. This file is self-descriptive and readers will be able to reproduce data at this processing level.

## Modelling workflow

The modelling workflow includes aspects involved in solving the forward problem of simulating data. In surface NMR this problem is normally posed using an adjoint macroscopic formulation (Weichman et al., 2000; Hertrich et al., 2009)

$$\overline{\mathcal{V}}_N^{(i)}(q,t) = \int_z \overline{\mathcal{K}}_0^{(i)}(\mathbf{B}_0, z, q, \mathcal{B}_T, \mathcal{B}_R^{(i)}) \int f_p(z, T_2^*) \times e^{-\frac{t+\tau_p/2}{T_2^*}} \, dT_2^* \, dz \quad (1)$$

In Equation 1 $\overline{\mathcal{V}}_N^{(i)}(q,t)$ represents the quadrature data for receiver loop *i* at time *t* and pulse moment *q*, which is the pulse current times the duration. The initial amplitude imaging kernel $\overline{\mathcal{K}}_0^{(i)}(\mathbf{B}_0, z, q, \mathcal{B}_T, \mathcal{B}_R^{(i)})$ is a complex-valued function of the static magnetic field $\mathbf{B}_0$, position (*r*), *q*, and transmitter and receiver loop magnetic fields $\mathcal{B}_{T,R}$. The response amplitude and decay rates are sensitive to the fractionated water model $f_p$ which corresponds to the amount of water at a position with a specific $T^*_2$ decay time. Additional details are available in the aforementioned literature.

Direct calculation of the imaging kernel $\overline{\mathcal{K}}_0^{(i)}$ is generally the most convenient means by which to perform inversion when the geoelectrical model is not changing. The kernel calculation in Akvo is provided by Merlin and is computed using an adaptive octree mesh which automatically refines itself in areas where the kernel is changing rapidly (Figure 12, Irons et al. (2010)). The $\mathcal{B}_{T,R}$ fields are computed in Lemma using adaptive discretization of arbitrary line source currents between points. Together, this has proven to be an extremely flexible approach which lends itself to arbitrary survey configuration (Irons et al., 2014). Akvo currently only supports 1D kernel function generation, but the underlying formulation is general and can be extended.

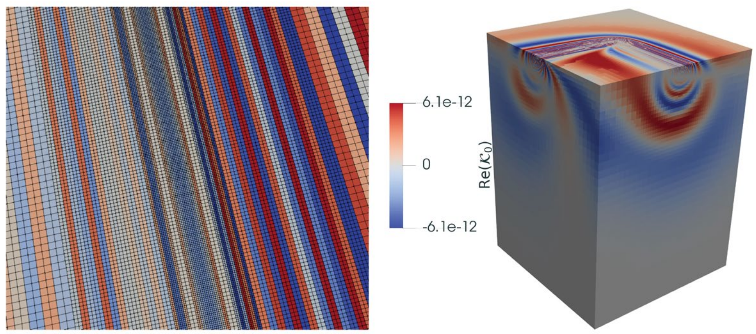



*Figure 12: The kernel calculation in Akvo is performed on an adaptive octree grid. In (a) a zoom of the top of the grid at 2.5 metres in depth is shown which illustrates the quickly changing behaviour of the kernel near the transmitter lines. In (b) 1/4 of a calculation volume is shown computed on an octree grid, gridlines are not shown. The grid is allowed to coarsen when the kernel is changing less rapidly and users can specify the desired precision of the kernel.*

## 1D Kernel tab

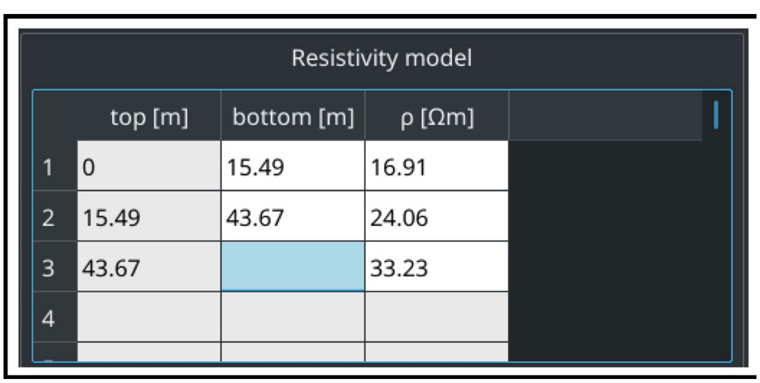
In the case that a single transmitter and receiver loop are used minimal lateral spatial information is present, and the 3D problem in Equation 1 is converted into a 1D sounding problem.

$$\overline{\mathcal{V}}_N^{(i)}(q,t) = \int_z \overline{\mathcal{K}}_0^{(i)}(\mathbf{B}_0, z, q, \mathcal{B}_T, \mathcal{B}_R^{(i)}) \int f_P(z,T_2^*) \times e^{-\frac{t+\tau_P/2}{T_2^*}} \, dT_2^* \, dz \qquad (2)$$

If the geoelectrical model is known or can be determined using other means (i.e. electrical conductivity, $\sigma$, from a transient electromagnetic (TEM) survey) then the forward sNMR problem can be posed linearly for a discrete set of depth layers (0, $N_z$) and pulse moments (0, $N_q$). Ignoring relaxation during pulse and dead time at the initial time ($t$ = 0) the exponential term in Equation 2 vanishes leaving

$$\overline{\mathcal{V}}_0^{(i)}[q] = \sum_{q=0}^{N_q} \sum_{z=0}^{N_z} \overline{\mathcal{K}}_0^{(i)} f. \qquad (3)$$

The matrix $\overline{\mathcal{K}}_0^{(i)} \in \mathbb{C}^1$ is the 1D discrete 1D kernel and has dimensions $N_z \times N_q$, the real vector $f$ is the total water at each layer has a length of $N_z$. This is a common and flexible formulation of the sNMR problem and exponential decay can easily be added to Equation 3 allowing for $\overline{\mathcal{K}}_0^{(i)}$ to be used for modelling various $f_p$ models.




| | top [m] | bottom [m] | ρ [Ωm] |
|---|---|---|---|
| 1 | 0 | 15.49 | 16.91 |
| 2 | 15.49 | 43.67 | 24.06 |
| 3 | 43.67 | | 33.23 |
| 4 | | | |

*Figure 13: Input of the resistivity model to use for the Site 58A dataset (Linux KDE/plasma).*

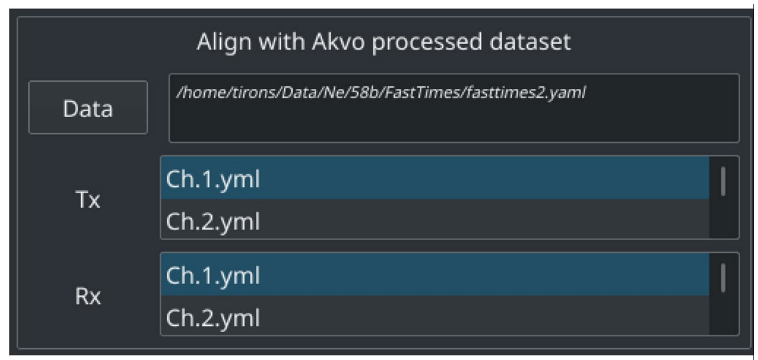



*Figure 14: Specification of the dataset and loops to use in kernel calculation (Linux KDE/plasma).*

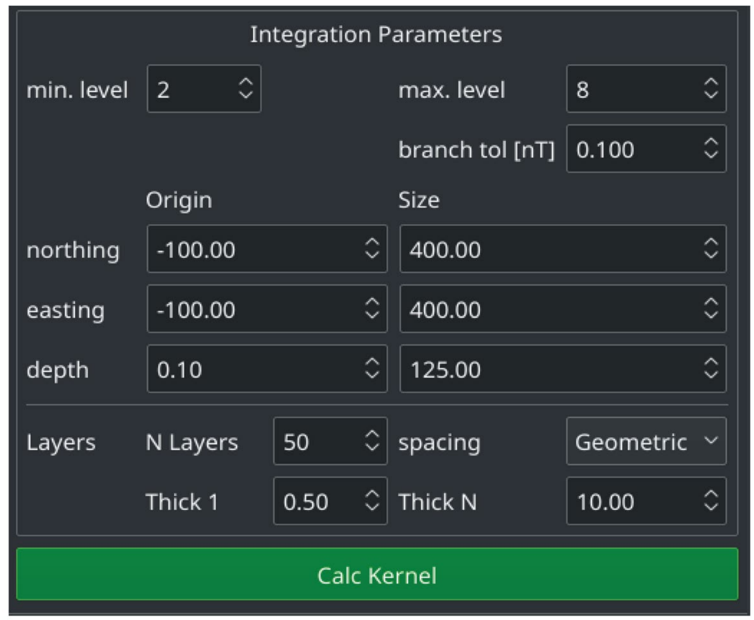



*Figure 15: Integration parameters used when calculating a 1D kernel for the 58A dataset (Linux KDE/plasma).*

The 1D Kernel tab allows Akvo users to calculate sNMR kernels over a layered earth. There are three parts to this process: 1) specification of the subsurface electrical conductivity model, 2) selection of the dataset which pulse parameters will be aligned with, and 3) integration parameters. After this setup the kernel calculation then proceeds by calling Lemma/Merlin routines. The user of Akvo has control over the precision of the Kernel calculation which affects the computational speed. Furthermore, the kernel calculation is multithreaded using OpenMP directives and benefits from multiprocessor shared-memory machines.

**Conductivity model.** While Lemma permits users to vary electrical conductivity, permittivity, and susceptibility (and can also include complex resistivity), Akvo simply allows users to specify a real 1D resistivity ($\rho$=1/$\sigma$) model which is entered in a 'from-to' style table. For the 58A dataset, a three-layer TEM-derived model is specified (Figure 13).

**Align with Akvo processed dataset.** In this portion of the GUI users select an Akvo processed yaml file which they would like the kernel to be based on. The users may then select the transmitter and receiver loops to be used for a specific kernel generation. For the 58A dataset, fasttimes.yaml (Listing 3) is selected along with Ch. 1 for the transmitter and receiver.

**Integration Parameters.** Each layer forms a rectangular volume and the kernel is first evaluated at the centroid. The rectangular

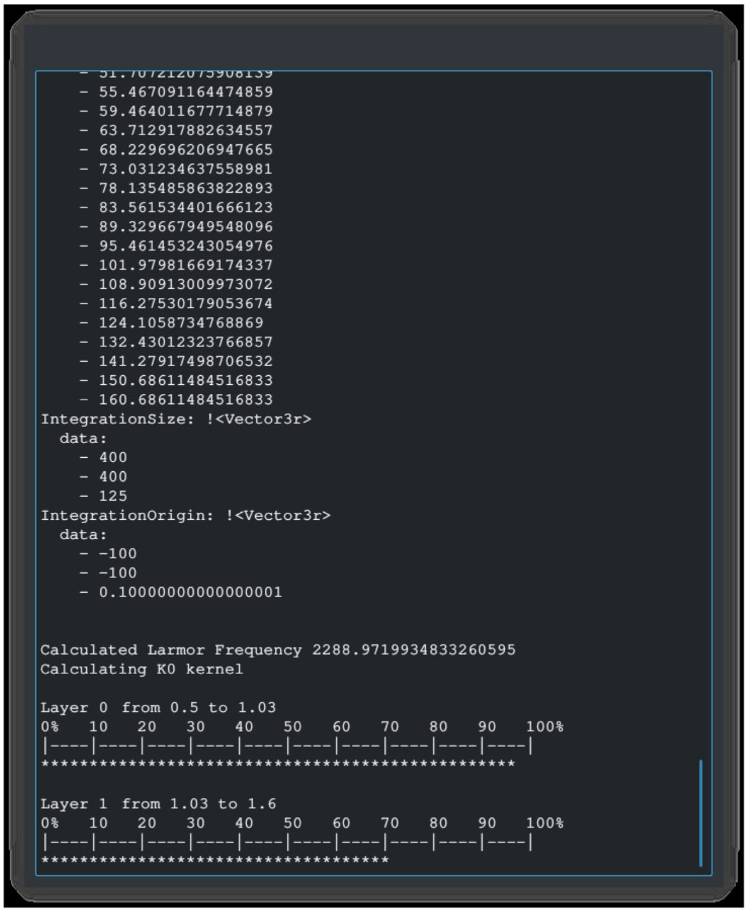



*Figure 16: Popup window displaying the progress of kernel calculation using Akvo on Linux (KDE/plasma).*

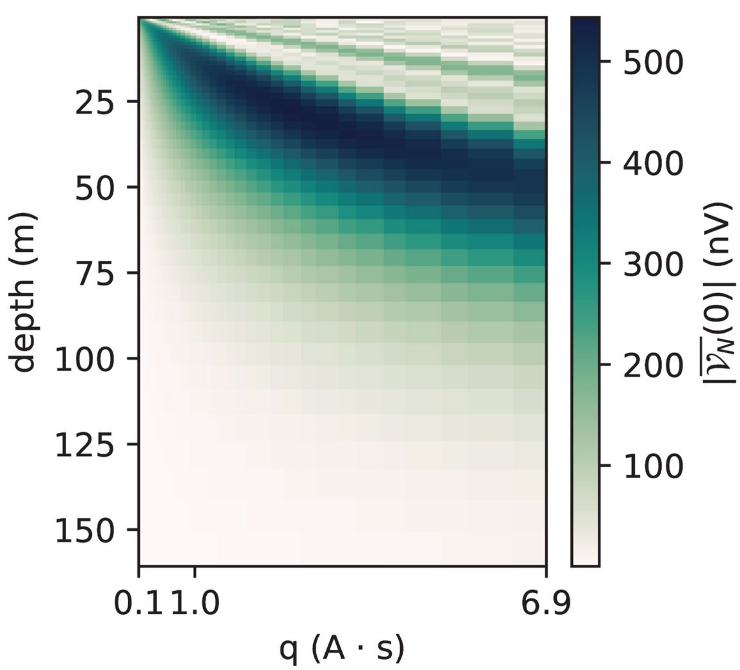



*Figure 17: Initial amplitude kernel calculation result for site 58A.*

volume is then split into 8 child cells of equal volume forming the first level of the octree. The kernel is evaluated at the centroid of each child assuming constant properties throughout the cell and the result is compared to the initial calculation. If the difference is greater than the tolerance, the splitting process proceeds and thereby refines the grid.

The octree used to calculate the kernel is configurable by users in Akvo. First the minimum (min. level) and maximum levels (max. level) can be specified, this determines how many times the grid will be split. The tolerance (branch tol) for the splitting is also user specified. The dimensions of the whole integration volume are defined by an origin point in the top southwest corner of the volume, and the size of the grid in northing, easting, and depth directions can then be specified. Finally, the number of layers (N Layers) needs to be set along with the thickness of the top (Thick 1) and bottom (Thick N) layers and the spacing between these, options include Geometric, Log, and Linear. The interface for these parameters is shown in Figure 15 for the Site 58A dataset. Pressing the Calc Kernel button will launch the kernel generation and result in a popup window indicating the calculation progress (Figure 16). The calculated kernel for the 58A dataset is shown in Figure 17.

## Inversion workflow

The sNMR inverse problem is concerned with mapping the recorded voltages into a water content model of the subsurface. As in many geophysical applications, the problem is ill-posed and care must be taken in order to achieve plausible solutions. The most widely-adopted sNMR inversion is the QT inversion which solves for a distribution of decay times at a discrete number of depth layers using the full time record in a comprehensive (all-at-once) approach (Müeller-Petke and Yaramanci, 2010). Many other inversion algorithms exist. Non-exhaustively these include QT inversion with monoexponential fits at each depth layer, parametric minimum layer inversion, frequency-domain $Q\omega$ inversion, and a large family of stochastic approaches.

The benefits of using complex data in inversion have been demonstrated numerous times in the literature. Braun et al. (2005) demonstrated the benefits of complex inversion on initial amplitude data. Iris instruments provided complex time step inversion in their proprietary inversion algorithms for many years as well. Irons and Li (2014) demonstrated the first comprehensive complex inversion ($Q\omega$) using actual field data which included dephasing effects which manifest as non-exponential $T_2^*$ decay. Complex QT inversion remains largely elusive due to disparities between $T_2^*$ and $T_2$. Grombacher (2018) proposed a novel Bloch equation approach to account for this in the quadrature data, but a robust time domain complex inversion remains a pressing need of the community. Phased (amplitude only) data is far less sensitive to these effects and inversion schemes operating on the real data are robust, the trade-off is somewhat lower resolution.

## QT Inversion tab

Akvo currently provides a QT inversion of the phased data. This is achieved by defining a set of $T^*_2$ bins and adding this decay to the initial amplitude Kernel $\forall\, T_2^*, K[t] = \left|\overline{\mathcal{K}_0}^{(i)}\right| e^{-t/T_2^*}$. The time gates and receiver channels are then concatenated to form a single kernel matrix $K$ with dimensions $M\left(=N_{T_2^*}\times N_{lay}\right)\times N\left(=N_t\times N_q\right)$. The data $v_{obs}$ are similarly concatenated from all available receiver

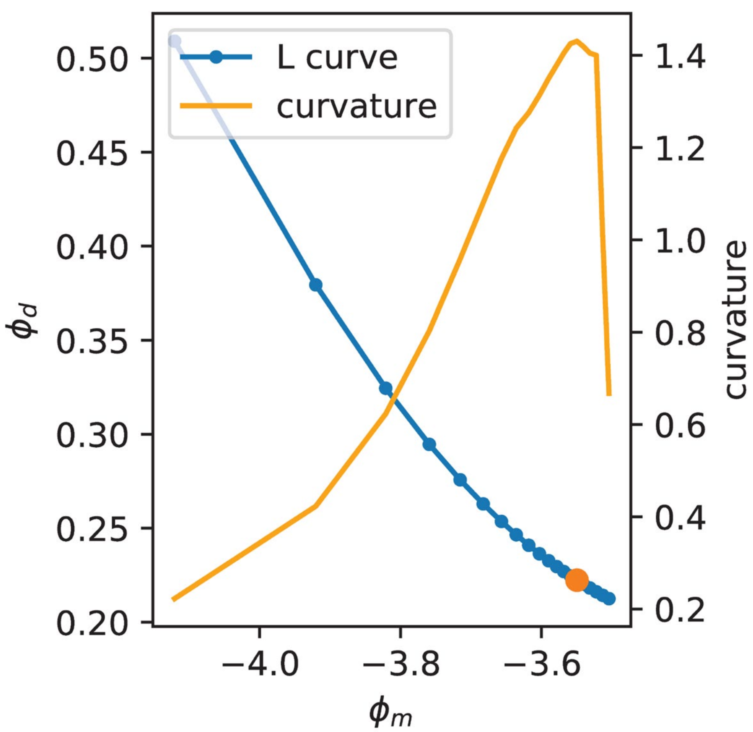



*Figure 18: L-curve from the inversion of the Site 58A data, the orange circle denote λ*.*

loop channels and the desired model fp is the distribution of partial water content for each $T^*_2$ bin at each depth layer. Each index of $f_p$ must be non-negative and a logarithmic barrier approach is taken. A Tikhonov regularisation strategy is employed using a trade-off parameter $\lambda$. Following Calvetti et al. (2004) a single step in the inversion is obtained by solving

$$\left(\mathbf{K}^T\mathbf{K} + \mu_x\mathbf{X}^{-2} + \lambda\mathbf{W}_m^T\mathbf{W}_m\right)\mathbf{f}_p = \mathbf{K}^T\mathbf{W}_d^T\mathbf{W}_d\mathbf{v}_{\mathrm{obs}} + 2\mu_x\mathbf{X}^{-1}\mathbf{c}. \qquad 12$$

In Equation 12 $\mathbf{c} = [1, . . . , 1]^T$ and $\mathbf{X}$ = diag $\{f_1, . . . , f_M\}$. The matrix $\mathbf{W}_m$ provides smoothness constraints. For this term Akvo uses either $\mathbf{I}$ which corresponds to a direct $L^2$ measure of model complexity or a finite difference first derivative term which penalizes non-smooth solutions in the depth and $T^*_2$ dimensions. The vector $\boldsymbol{v}_{obs}$ represents the observed field data ($\mathbf{V}_N$). The matrix $\mathbf{W}_d$ is a data weighting matrix whose diagonal entries are the bootstrapped error estimates for each time gate of $\boldsymbol{v}_{obs}$ (i.e. std ($\tilde{\in}$)). The $\mu_x$, and $\mathbf{X}$ terms impose the non-negativity constraint and are relaxed within a non-linear optimisation strategy. The ideal trade-off parameter $\lambda^*$ is determined using the point of maximum curvature in the parametric L-curve graph of data misfit and model complexity, both of which are functions of $\lambda$ (Figure 18). As such, selection of $\lambda^*$ is not a user-selected input which reduces the chance of operator-induced bias.

For the Site 58A data, the inversion result is shown in Figure 19. This result is in good agreement with previously published results. This inversion used 35 $T^*_2$ bins spaced logarithmically between 10 and 1000 ms, a smallest model objective function in both $T^*_2$ and depth, and an initial $\lambda$ value of 1 × $10^7$. The inversion algorithm is repeatable by running Listing 4 on the command line.

*Listing 4: Script for repeating the inversion result presented in this paper, invert.yml is available on Zenodo.*

```
akvoQT invert.yml
```

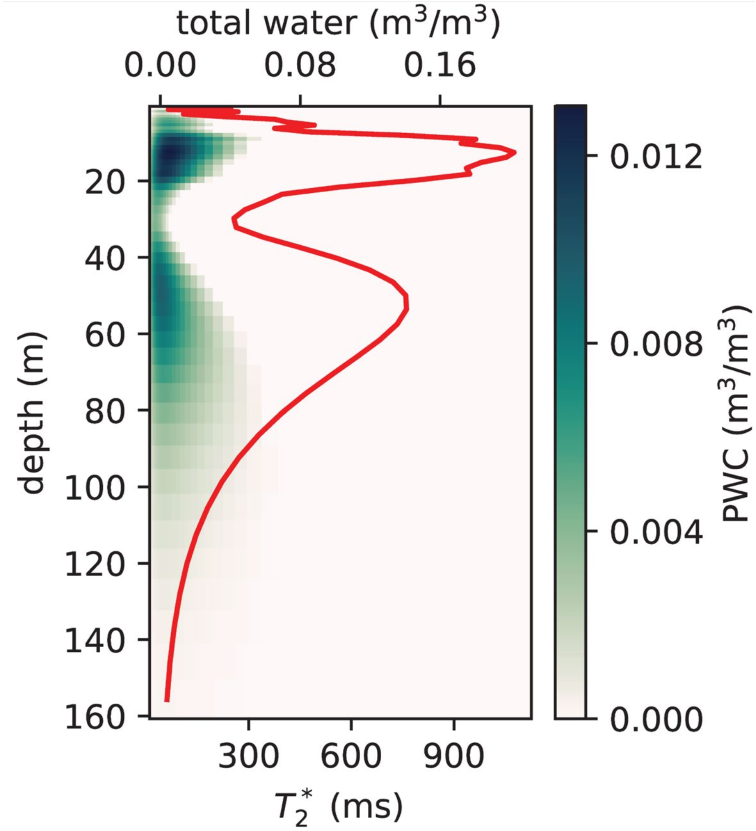



*Figure 19: Inversion result for the 58A dataset, the total water contents (red line) are in agreement with previously published results. Depth of investigation limit not shown, but sensitivity tapers off past 60 m.*

## Engagement

Open source software projects such as Akvo thrive when there is a community of engaged users, testers, developers working together. As such, we are keen to increase our userbase and to attract new contributors. Additionally, we are hopeful that this tool will be useful to others and encourage the release of reproducible publications and data. Readers interested in getting involved, at any level, are encouraged to contact the core developers through links on the project website [https://akvo.lemmasoftware.org](https://akvo.lemmasoftware.org). Pull and feature requests can be submitted either by email or through the GitHub mirror.

A paper such as this one necessarily reports a snapshot of a changing and dynamic software infrastructure. We strive to improve functionality, user interfaces, and also to remove bugs as they are identified. Towards this end, Akvo has several planned enhancements including

- FIR filters design – The current choices of frequency domain windowing and IIR time domain filtering would be complemented by an option for finite impulse response (FIR) filter design and implementation.
- Non-FID pulses – Akvo can read and process adiabatic and T1 pulses, but no inversion currently exists for these data.
- Frequency-domain inversion – This alternative inversion scheme has some advantages over the standard approach, and it will be nice for users to compare the results.

- Inversion model appraisal – Assessment of depth of investigation and parameter uncertainty need to be added to the inversions.

## Conclusions

Akvo is a free and open source surface NMR workbench which has been released to the community in the hopes that it will be useful. Additionally, Akvo provides a means by which to publish reproducible surface NMR results which improves the transparency of findings. Interested readers are encouraged to get involved in the project.

## References


Behroozmand, A.A., Keating, K., Auken, E., 2015. A review of the principles and applications of the nmr technique for near-surface characteriza- tion. Surveys in Geophysics 36, 27–85. URL: https://doi.org/10.1007/s10712-014-9304-0, doi:10.1007/s10712-014-9304-0.

Borges, H., Hora, A., Valente, M.T., 2016. Understanding the factors that impact the popularity of github repositories, in: 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 334–344. doi:10.1109/ICSME.2016.31.

Braun, M., Hertrich, M., Yaramanci, U., 2005. Study on complex inversion of magnetic resonance sounding signals. Near surface geophysics 3, 155–163.

Calvetti, D., Lewis, B., L.Reichel, Sgallari, F., 2004. Tikhonov regularization with nonnegatvity contraint. electronic transactions on numerical analysis 18, 153–173.

Costabel, S., Müller-Petke, M., 2014. Despiking of magnetic resonance signals in time and wavelet domains. Near Surface Geophysics 12, 185–197.

Dalgaard, E., Auken, E., Larsen, J.J., 2012. Adaptive noise cancelling of multichannel magnetic resonance sounding signals. Geophysical Journal International 191, 88–100. URL: +http://dx.doi.org/10.1111/j.1365-246X.2012.05618.x, doi:10.1111/j.1365-246X. 2012.05618.x.

Davis, A.C., Skibbe, N., Müller-Petke, M., 2019. First measurements of surface nuclear magnetic resonance signals in a grounded bipole. Geophysical Research Letters 46. doi:10.1029/2019GL084342.

Grombacher, D., 2018. Numerically optimized modulations for adiabatic pulses in surface nuclear magnetic resonance. GEOPHYSICS 83, JM1–JM14. URL: https://doi.org/10.1190/geo2016-0574.1, doi:10.1190/geo2016-0574.1.

Haykin, S.S., 2008. Adaptive filter theory. Pearson Education India.

Hertrich, M., Green, A.G., Braun, M., Yaramanci, U., 2009. High-resolution surface NMR tomography of shallow aquifers based on multio-set measurements. Geophysics 74, G47–G59. URL: http://link.aip.org/link/?GPY/74/G47/1, doi:10.1190/1.3258342.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. Computing in Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Irons, T., 2019. Surface nuclear magnetic resonance survey in Lexington, NE 29 March 2010. URL: https://doi.org/10.5281/zenodo. 3477548, doi:10.5281/zenodo.3477548.

Irons, T., Kass, M.A., et al., 2012a. Lemma v1. https://lemmasoftware.org.

Irons, T., Li, Y., McKenna, J.R., 2010. Frequency domain surface nuclear magnetic resonance forward modeling on an adaptive octree mesh, in: SEG Technical Program Expanded Abstracts, pp. 3935–3939. URL: http://library.seg.org/doi/abs/10.1190/1.3513671, doi:10. 1190/1.3513671, arXiv: http://library.seg.org/doi/pdf/10.1190/1.3513671.

Irons, T.P., Hobza, C.M., Steele, G.V., Abraham, J.D., Woodward, D.C., 2012b. Quantification of aquifer properties with surface nuclear magnetic resonance in the Platte River valley, central Nebraska, using a novel inversion method. Scientific Investigations Report 2012-5189. U.S. Geological Survey. Available at http://pubs.usgs.gov/sir/2012/5189/.

Irons, T.P., Li, Y., 2014. Pulse and Fourier transform surface nuclear magnetic resonance: comprehensive modelling and inversion incorporating complex data and static dephasing dynamics. Geophysical Journal International 199, 1372–1394. doi:10.1093/gji/ggu323.

Irons, T.P., McPherson, B.J., Kass, M.A., 2018. Reliable noise measure in time-gated NMR data. Geophysical Journal International 215, 959–964. URL: https://doi.org/10.1093/gji/ggy318, doi:10.1093/gji/ggy318.

Irons, T.P., Quinn, M.C., Li, Y., McKenna, J.R., 2014. A numerical assessment of the use of surface nuclear magnetic resonance to monitor internal erosion and piping in earthen embankments. Near Surface Geophysics 12, 325–334. doi:10.3997/1873-0604.2014004.

Jakob, W., Rhinelander, J., Moldovan, D., 2017. pybind11 – seamless operability between c++11 and python. Https://github.com/pybind/pybind11.

Kass, M., Bloss, B., Irons, T., Cannia, J., Abraham, J., 2014. Magnetic resonance sounding data collected in the north platte, twin platte, and south platte natural resource districts western nebraska, fall 2012: U.s. geological survey open-file report 2014-1138. http://dx.doi.org/10.3133/ofr20141138.

Keating, K., Binley, A., Bense, V., Van Dam, R.L., Christiansen, H.H., 2018. Combined geophysical measurements provide evidence for unfrozen water in permafrost in the Adventdalen valley in Svalbard. Geophysical Research Letters 45, 7606–7614. URL: https://agupubs. onlinelibrary.wiley.com/doi/abs/10.1029/2017GL076508, doi:10.1029/2017GL076508.

Knight, R., Grunewald, E., Irons, T., Dlubac, K., Song, Y., Bachman, H.N., Grau, B., Walsh, D., Abraham, J.D., Cannia, J., 2012. Field experiment provides ground truth for surface nuclear magnetic resonance measurement. Geophysical Research

Letters 39, 7p. URL: http://dx.doi. org/10.1029/2011GL050167, doi:10.1029/2011GL050167.

Kremer, T., Juul Larsen, J., Nguyen, F., 2019. Processing harmonic EM noise with multiple or unstable frequency content in surface nmr surveys. Geophysical Journal International 219, 753–775.

Larsen, J.J., 2016. Model-based subtraction of spikes from surface nuclear magnetic resonance data. Geophysics 81, WB1–WB8.

Larsen, J.J., Dalgaard, E., Auken, E., 2014. Noise cancelling of MRS signals combining model-based removal of powerline harmonics and multi- channel wiener filtering. Geophysical Journal International 196, 828–836.

Müeller-Petke, M., Yaramanci, U., 2010. Qt inversion — comprehensive use of the complete surface NMR data set. Geophysics 75, WA199– WA209. URL: http://link.aip.org/link/?GPY/75/WA199/1, doi:10.1190/1.3471523.

Müller-Petke, M., Braun, M., Hertrich, M., Costabel, S., Walbrecker, J., 2016. Mrsmatlab — a software tool for processing, modeling, and inversion of magnetic resonance sounding data. Geophysics 81, WB9–WB21. URL: http://geophysics.geoscienceworld.org/content/81/4/ WB9, doi:10.1190/geo2015-0461.1.

Müller-Petke, M., Costabel, S., 2014. Comparison and optimal parameter settings of referencebased harmonic noise cancellation in time and frequency domains for surface-nmr. Near Surface Geophysics 12, 199–210.

Müller-Petke, M., Walbrecker, J.O., Knight, R., 2013. The inversion of surface-nmr T1 data for improved aquifer characterization. GEOPHYSICS 78, EN83–EN94. URL: https://doi.org/10.1190/geo2013-0035.1, doi:10.1190/geo2013-0035.1.

Rücker, C., Günther, T., Wagner, F.M., 2017. pyGIMLi: An open-source library for modelling and inversion in geophysics. Computers and Geosciences 109, 106–123. URL: http://www.sciencedirect.com/science/article/pii/S0098300417300584, doi:10.1016/j. cageo.2017.07.011.

Walbrecker, J.O., Hertrich, M., Lehmann-Horn, J.A., Green, A.G., 2011. Estimating the longitudinal relaxation time T1 in surface NMR. Geophysics 76, F111–122. URL: http://geophysics.geoscienceworld.org/cgi/content/abstract/76/2/F111, doi:10.1190/1.3549642.

Walsh, D.O., 2008. Multi-channel surface NMR instrumentation and software for 1d/2d groundwater investigations. Journal of Applied Geophysics 66, 140 – 150. URL: http://www.sciencedirect.com/science/article/pii/S0926985108000347, doi:DOI:10.1016/j. jappgeo. 2008.03.006. resonance Sounding – a Reality in Applied Hydrogeophysics.

Weichman, P.B., Lavely, E.M., Ritzwoller, M.H., 2000. Theory of surface nuclear magnetic resonance with applications to geophysical imaging problems. Phys. Rev. E 62, 1290–1312. doi:10.1103/PhysRevE.62.1290.

Widrow, B., Glover, J.R., McCool, J.M., Kaunitz, J., Williams, C.S., Hearn, R.H., Zeidler, J.R., Dong, J.E., Goodlin, R.C., 1975. Adaptive noise cancelling: Principles and applications. Proceedings of the IEEE 63, 1692–1716.

# Author Bios

Trevor Irons
***Department of Geophysics***
***Montana Technical University***
***Butte MT, USA***
tirons@mtech.edu

Trevor Irons is a Research Assistant Professor in the Department of Geophysics at Montana Technical University, in Butte Montana, where his interests include near-surface hydrogeophysics, electromagnetics, carbon sequestration, and groundwater modelling. He enjoys working with students almost as much as studying rocks. He graduated from the Colorado School of Mines in 2013 with a PhD in Geophysics and was previously employed at the US Geological Survey in Denver, CO and the University of Utah, in Salt Lake City. When not advocating for free and open source software he enjoys hiking, cycling, and telemark skiing in Montana with two misbehaved dogs, his beautiful wife, and their two wonderful children.

Thomas Kremer
***University of Nantes, Univ Angers, CNRS, LPG, UMR 6112, F-44000 Nantes, France***
Thomas.Kremer@univ-nantes.fr

Thomas Kremer is a near-surface geophysicist and is presently a post-doc at the University of Nantes in France. He holds a PhD in applied geophysics from the University of Paris-Sorbonne. His main research focuses on the understanding of hydrogeological systems using electromagnetic methods including induced polarization, electrical resistivity tomography, and surface nuclear magnetic resonance. Applications of interest include the characterization of carbo-gaseous water springs, karstic fractured aquifers, and agro-geophysical issues. Beyond research, Thomas also enjoys teaching Earth science courses at the University of Nantes, from first year to master classes. On the weekends, sailing boats, walking the earth, and riding bikes are a perfect way to spend some time.

Benjamin R. Bloss
***United States Geological Survey, Geology, Geophysics, and Geochemistry Science Center,***
***Denver CO, USA***
bbloss@usgs.gov

Ben Bloss is a geophysicist at the USGS and holds a master's degree from the Colorado School of Mines. Ben uses electromagnetic (EM) methods to help map physical property distributions from scales ranging from small hydrogeologic projects to large regional mineral resource assessments. A couple of Ben's current research interests include using airborne EM data to help map modern distribution of mineralized zones and help refine the understanding of the tectonic history of the Stillwater Complex in Montana, understanding the effects of cultural contamination in time-domain EM data, and developing novel CSEM methods to better understand the effects of carbon dioxide injections into geologic formations. When not conducting research, Ben enjoys spending time in nature, both on foot and ski, with his friends and family.

M. Andy Kass
***Hydrogeophysics Group, Dept. of Geoscience,***
***Aarhus University, Denmark***
andy.kass@geo.au.dk

Andy is a geophysicist with the HydroGeophysics Group at Aarhus University, Denmark. He completed his PhD in 2012 at the Colorado School of Mines, and spent the next 5 years as a research geophysicist with the US Geological Survey. His current research interests are in ground-based tensor magnetic gradiometry, surface NMR, and transient electromagnetics. When not constructing algorithms or surveying across the world, he runs an apple plantation outside of Aarhus.